ILWIS 2.2 for Windows

The Integrated Land and Water Information System

Installation and New Functionality

ILWIS Development,
International Institute for
Aerospace Survey and Earth Sciences
Enschede, The Netherlands

ITC
The International Institute for Aerospace Survey and Earth Sciences, Enschede, is the largest institute for international higher education in the Netherlands. Its main objective is to assist developing countries in human resources development in aerospace surveys, Remote Sensing applications, and the establishment of geoinformation systems and the management of geoinformation. To this end, ITC concentrates on three activities: education/training, research and advisory services. In-house expertise covers an extensive range of disciplines in the above fields.

PCI-ILWIS Nederland B.V.
ITC & the PCI Geomatics Group started a joint venture for the distribution of geomatics software solutions and training to developing nations. This partnership resulted in the official establishment of "PCI-ILWIS Nederland B.V."
As of now you are welcome to contact the PCI-ILWIS Nederland B.V. for your ILWIS orders.

Contact addresses
For general information about ILWIS,
please contact:
PCI-ILWIS Nederland B.V.
P.O. Box 6,
7500 AA Enschede
The Netherlands
Tel. : +31-53-4 874 337
Fax : +31-53-4 874 484
E-mail: ilwis@itc.nl
Web-Site: http://www.itc.nl/ilwis

Mail wishes and suggestions to: wishes.ilwis@itc.nl
Mail bug reports to: bugs.ilwis@itc.nl
Mail support questions to: support.ilwis@itc.nl
Tel. : +31-53-4 874 217
Fax : +31-53-4 874 484

Furthermore, you can subscribe yourself to the ILWIS discussion list. For more information, refer to our web pages.

Authors and Acknowledgments

The first chapter in this ILWIS 2.2 Guide briefly lists all new functionality in ILWIS 2.2 compared to ILWIS 2.1 This chapter is written by Wim Koolhoven.

Chapter 2 in this ILWIS 2.2 Guide completely replaces the ILWIS 2.1 Installation Guide; the new Installation chapter is written by Willem Nieuwenhuis.

Chapters 3, 4, 5, 6, 7, and 8 in this ILWIS 2.2 Guide should be seen as a supplement to the ILWIS 2.1 Reference Guide. Only new or drastically changed Help texts as written by Petra Budde are included in this ILWIS 2.2 Guide. For other descriptions, please refer to the ILWIS 2.1 Reference Guide.

Chapter 9 in this ILWIS 2.2 Guide contains two exercises, which can be seen as an extension of the ILWIS 2.1 User's Guide. The exercises were written by Remco Dost, Hans de Brouwer and Jan Hendrikse. The methods and the data used in the exercises are the sole responsibility of the authors of the exercise. For questions and remarks concerning a case study, please contact the author.

Technical proofreading is done by Wim Koolhoven and Petra Budde. This guide is compiled and edited by Willem Nieuwenhuis.

The ILWIS software is designed by Wim Koolhoven and Jelle Wind. The entire ILWIS Development group responsible for the release of ILWIS 2.2 consists of:

- Software Development: Jan Hendrikse, Wim Koolhoven, Willem Nieuwenhuis, Martin Schouwenburg, Dick Visser and Jelle Wind.
- Testing and Support: Boudewijn van Leeuwen, Jelger Kooistra, and Guillermo Calderón,
- Help and Documentation: Petra Budde and Willem Nieuwenhuis.

Project Leader: Wim Koolhoven.

We like to thank the alpha and beta testers both within and outside ITC who gave useful feedback to improve the software.

Contents

Preface 9

1. ILWIS 2.2 new features 13

1.1 Visualization and Annotation 13
1.2 Referencing and Transformation 13
1.3 Statistics 14
1.4 Other operations 15
1.5 Miscellaneous 15

2. Installing ILWIS 2.2 17

2.1 The ILWIS 2.2 package and related services 17
2.1.1 The ILWIS 2.2 package 17
2.1.2 Warranty 17
2.1.3 ILWIS hardware key 17
2.1.4 Technical support and software maintenance 18
2.2 Hardware and software requirements 18
2.2.1 Minimum hardware and software requirements 18
2.2.2 Recommended hardware and software requirements 18
2.2.3 Digitizers, printers and plotters 18
2.2.4 Display drivers 19
2.2.5 Required hard disk space 19
2.3 Installing ILWIS 2.2 19
2.3.1 Installation from CD-ROM 19
2.3.2 Add components to the current installation 21
2.3.3 Uninstalling the ILWIS 2.2 software 22
2.3.4 After installation 22
2.3.5 Viewing ILWIS documentation PDF files 24
2.3.6 Installing the Adobe Acrobat PDF reader 24
2.4 Installation and configuration of digitizers 24
2.4.1 Set up your digitizer using a WinTab driver 25
2.4.2 Set up your digitizer manually 25
2.4.3 Settings for some widely available digitizers 27
2.5 Printer and plotter setup 31

3. ILWIS window types 33

3.1 Main window 33
3.1.1 Main window Menu commands 33

4. ILWIS objects 37

4.1 Data objects 38
4.1.1 Raster maps 38
4.1.2 Polygon maps 41
4.1.3 Segment maps 44
4.1.4 Point maps 47
4.1.5 Map lists 51
4.2 Service objects 53
4.2.1 Representations 53
4.2.2 Georeferences 57
4.2.3 Coordinate system 62
4.3 Annotation 70
4.3.1 Annotation Text 70
4.3.2 Simple Annotation 71

5. Editors 77

5.1 Representation Class editor 77
5.2 Georeference Tiepoints editor 80
5.3 Coordinate system Tiepoints editor 84
5.4 Annotation Text editor 88

6. Map & Table Calculation 93

6.1 Map calculation 93
6.1.1 Logical operators 93
6.2 Table calculation 94
6.2.1 Aggregating values 94
6.2.2 Joining columns of other tables 98
6.2.3 Creating and running scripts (example) 104

7. Operations 107

7.1 Visualization 107
7.1.1 Show map list as Color Composite 107
7.2 Raster Operations 108
7.2.1 Cross 108
7.2.2 Glue raster maps 111
7.3 Image Processing 119
7.3.1 Filter 119
7.3.2 Filter types 119
7.3.3 Color composite 136
7.3.4 Resample 145
7.4 Statistics 150
7.4.1 Optimum Index Factor 150
7.4.2 Point statistics 152
7.4.3 Spatial correlation 153

7.5 Interpolation 163
7.5.1 Point interpolation 163
7.5.2 Nearest point 165
7.5.3 Moving average 167
7.5.4 Trend surface 172
7.5.5 Moving surface 175
7.5.6 Kriging 181
7.6 Vector operations 194
7.6.1 Transform vector map 194
7.6.2 Transform coordinates 198
7.7 Rasterize 200
7.7.1 Polygons to raster 200
7.7.2 Segments to raster 202
7.8 Vectorize 205
7.8.1 Raster to polygons 205
7.8.2 Polygons to points 208
7.9 Table operations 211
7.9.1 Change domain of table 211
7.9.2 Table to point map 214
7.9.3 Glue tables 218

8. How to… 225

8.1 How to add… 225
8.1.1 How to add annotation to a map window 225
8.1.2 How to add coordinates to an image etc. 227
8.2 How to calculate… 231
8.2.1 How to calculate a confusion matrix 231
8.2.2 How to calculate a Digital Elevation Model 232
8.2.3 How to calculate the Optimum Index Factor (OIF) 235
8.3 How to calculate (Advanced)… 236
8.3.1 How to calculate a cross-section through the terrain 236
8.4 How to change… 239
8.4.1 How to change the domain of a map 239
8.5 How to create… 242
8.5.1 How to create a color composite 242
8.5.2 How to create a georeference 244
8.5.3 How to create an orthophoto 246
8.5.4 How to create a representation 247
8.5.5 How to create a sample set for image classification 249
8.6 How to display… 250
8.6.1 How to display a map list as a slide show 250
8.7 How to edit… 252
8.7.1 How to edit a georeference 252
8.7.2 How to edit a representation 253
8.8 How to export data 256
8.9 How to import… 257
8.9.1 How to import ASCII point data 257

8.9.2 How to import data 264
8.10 How to monoplot 266
8.11 How to save… 269
8.11.1 How to save annotation 269
8.11.2 How to save a map view 270
8.12 How to screen digitize 270
8.13 How to use… 273
8.13.1 How to use the Georeference Tiepoints editor 273
8.13.2 How to use parameters in scripts 275

9. Exercises 277

9.1 Georeference Orthophoto 277
9.1.1 Materials 277
9.1.2 An affine transformation 278
9.1.3 A bilinear transformation 278
9.1.4 A projective transformation 279
9.1.5 A projective transformation and relief displacement correction 279
9.2 Geostatistics in ILWIS 2.2 282
9.2.1 Point maps with a value domain 282
9.2.2 Spatial correlation and empirical semi-variogram 283
9.2.3 Modeling the semi-variogram 284
9.2.4 Point map interpolation 284
9.2.5 Kriging estimation and error map 285

Appendices 287

A. ILWIS objects 287
B. ILWIS operations 295
C. ILWIS expressions 311
D. ILWIS expressions (alphabetic) 327
E. ILWIS script language (syntax) 331
E.1 Additional script commands 333
F. Errata ILWIS 2.1 Reference Guide 349
G. Software License Agreement 353
H. Customers Services Report Form 355
I. Index 357

Preface

Thank you for purchasing ILWIS 2.2!

ILWIS 2.2 is an upgrade of ILWIS 2.1; compared to ILWIS 2.1, the following functionality has been added:

- Extended visualization and annotation capabilities;
- New types of georeferences to handle aerial photographs of non-flat terrain;
- New types of coordinate systems to handle artificial and local coordinates;
- Extended capabilities for GeoStatistics, most notably the Kriging operation;
- Extended import and export possibilities.

Chapter 1 of this ILWIS 2.2 Guide lists the new functionality of ILWIS 2.2 in more detail.

This ILWIS 2.2 Guide which accompanies the ILWIS 2.2 software:

- replaces the ILWIS 2.1 Installation Guide, and
- should be regarded as a supplement to the ILWIS 2.1 Reference Guide and the ILWIS 2.1 User's Guide.

Below, the contents of this guide are listed in more detail.

What is ILWIS?

ILWIS is an acronym for the Integrated Land and Water Information System. It is a Geographic Information System (GIS) with Image Processing capabilities.

ILWIS allows you to input, manage, analyze and present geographical data. From the data you can generate information on the spatial and temporal patterns and processes of the earth surface.

ILWIS is developed by the International Institute for Aerospace Survey and Earth Sciences (ITC), Enschede, The Netherlands. For more than a decade, since 1985, the software has undergone major improvements. By December 1996, with the release of ILWIS 2.0, the software environment had been changed from DOS to Windows. Major upgrade ILWIS 2.1 was released in October 1997. In the near future, we plan to port ILWIS to Win32 and incorporate new features of Windows 95/98/NT.

At ITC, the ILWIS Development group continues to improve the ILWIS software. Any suggestions, wishes or requests for new functionality are most welcome at wishes.ilwis@itc.nl. The ILWIS user group will have a large influence on the priority settings of any new developments.

The ILWIS media

The ILWIS 2.2 Guide is part of the documentation for ILWIS 2.2. The full documentation includes:

- ILWIS Beginner's Guide. A first look at ILWIS, introducing the basic concepts, some essential ILWIS techniques and the main operations included in ILWIS;
- ILWIS User's Guide. Training in the skills you need to work with ILWIS. It provides numerous exercises to practice GIS techniques and Image Processing operations;
- ILWIS Applications Guide. Advanced procedures to work with ILWIS, providing 25 case studies for various research disciplines;
- ILWIS 2.1 Reference Guide. Detailed description of the functionality of ILWIS including its window types, the objects, the operations, etc., illustrated with tips and examples.

Besides these Guides, extensive on-line and context-sensitive Help is available in the program.

Structure of this ILWIS 2.2 Guide

This ILWIS 2.2 Guide should be used in combination with the existing ILWIS 2.1 documentation as described above; the ILWIS 2.2 Guide does *not* replace the existing ILWIS 2.1 documentation.

This ILWIS 2.2 Guide consists of 9 chapters:

- Chapter one, New Features, gives a brief overview of new and updated functionality in ILWIS 2.2.
- Chapter two, Installing ILWIS 2.2, provides instructions to install ILWIS, as well as information about connecting a digitizer and/or printers and plotters. This chapter replaces the ILWIS 2.1 Installation Guide.
- Chapter three, ILWIS window types, contains the updated menu of the Main window.
- Chapter four, ILWIS objects, contains updated documentation for ILWIS objects.
- Chapter five, Editors, describes the functionality of new or changed editors: the representation class editor, the georeference tiepoint editor, the coordinate system tiepoint editor and the annotation text editor.
- Chapter six, Map and Table calculation, lists new operators and functions to calculate with maps and tables and updated documentation.
- Chapters seven, Operations, describes new operations in ILWIS 2.2 and contains updated documentation for several other operations.

- Chapter eight, How to…, provides you with new and updated information on performing a certain task and choosing the correct menu commands.
- Chapter nine, Exercises, consists of two exercises illustrating the new functionality: Georeference Orthophoto and Geostatistics.
- Finally, the Appendices provide overviews of all objects, operations, commands, expressions that can be used on the command lines, syntax for scripts, file extensions, etc.

Conventions used in the ILWIS 2.2 Guide

This guide is formatted in such a way that the specific actions dealing with the software are separated from the accompanying text.

- Formulas that you should type exactly as described are shown in `Courier New 10`.
  For example: `Mapc = Mapa + Mapb`

☞ This is a *tip box*. It is used to give tips.

Chapter 1

ILWIS 2.2 new features

This chapter gives a brief overview of new and improved functionality of ILWIS 2.2. For detailed information, see chapters 3, 4, 5, 6, 7, and 8, or the on-line Help.

1.1 Visualization and Annotation

Interactive Color Composite
Maps in a map list can be shown directly as a color composite in a map window. The Display Options of the map list allows you to interactively specify which bands of the map list should be displayed and how they should be mapped on colors.

Improved representation class
A *representation class* now stores more information. For points, any symbol font with a certain color and size can be used. Segments can be displayed by complex line types (single, double, triple, dot, dash, dash-dot, dash-dot-dot, blocked, symbol); polygons can be displayed with hatchings or patterns. The annotation legends for points, segments and polygons have been adapted to visualize these new representation methods.

Special symbols for ID point data with attribute data
A point map with an identifier domain and a linked attribute table can now be displayed using one column to define colors and another to define symbols. Furthermore, the attributes can be used to display ID point maps as arrows (e.g. for magnetic fields or wind speed and direction), piecharts, bar graphs, line graphs, composite bars and volume cubes.

Store multiple texts in an Annotation Text object
An *Annotation Text* object is a new object designed to store multiple texts with a fixed location on a map. Font properties such as font type, size, color, justification, rotation etc. can be set for individual text elements. An *Annotation Text* object can be created from a point map, segment map or polygon map (i.e. as automatic labels), or from scratch. You can visually edit an *Annotation Text* object in the Annotation Text editor in a map window, or as a table in a table window.

1.2 Referencing and Transformation

New types of georeferences
The new georeference types *GeoRefOrthoPhoto* and *GeoRefDirectLinear* handle aerial photographs of non-flat terrain. The georeferences can be completely defined

with the help of a DTM, specified tiepoints and, for the GeoRefOrthoPhoto, the camera parameters. This enables monoplotting by mouse on a scanned aerial photograph as well as the creation of an orthophoto by using the Resample operation.

New projections
The following projections have been included: *Bonne*, *Cassini*, *UPS*, *Oblique Mercator*, *General Perspective*.
Furthermore, some national topographic systems and their projection parameters were included*: Gauss-Krueger* (D), *Gauss-Boaga* (I), *Lambert Conical Conformal* (F).

New types of coordinate systems
The following coordinate systems types, which are not based on a projection, have been defined:

- a *CoordSystemFormula* relates to another coordinate system with a user-defined shift, scaling, rotation or by a user-specified formula.
- a *CoordSystemTiePoints* relates to another coordinate system with a user-specified transformation method and a series of tiepoints. This enables you to work with local coordinate systems and to handle imported vector data that is not georeferenced.
- a *CoordSystemLatLon* enables datum transformations on geographic coordinates.

1.3 Statistics

Improved Spatial correlation operation
Semi-variances can now be calculated with a user-specified lag. You can choose between omni-directional semi-variances or bi-directional semi-variances with a user-specified direction, tolerance and bandwidth. In a graph window, you can display the semi-variogram and draw semi-variogram models through the output semi-variance values.

More possibilities to create graphs
Besides creating graphs from data in columns, you can now also define a graph by an expression as `y = f(x)`, use a least squares fit, and draw semi-variogram models through semi-variance values obtained from the Spatial correlation operation.

New operation Kriging
From a point map and a user-specified semi-variogram model, the Kriging operation calculates kriging estimates. Both Simple Kriging and Ordinary Kriging are available. Optionally, an error map can be obtained.

Enhanced column statistics in a table window
Statistics on columns is now available from the menu of a table window and is enhanced. You can calculate: minimum, maximum, sum, average, variance, standard

deviation and standard error of one column, and the correlation and covariance of two columns. Furthermore, a Students *t-test* and a χ^2-*test* can be performed.

1.4 Other operations

New operation: Change domain of table
After calculations or import, it may be useful to change the domain of a table. One of the options is to use a column from the table to define the new domain; in that case aggregations are possible.

New operation: Table to point map
From a table containing X and Y coordinate columns, a point map can be created.

New operation: Glue tables
The operation glues multiple tables together. The operation is automatically called when multiple identifier maps with attribute tables are glued together.

Glue raster maps
The operation now accepts more than two input raster maps; this allows for easy mosaicking. Furthermore, when combining class and value maps, the result will become a color map that can be used for presentation purposes.

Cross (TableCross)
An option has been added to ignore or include undefined values in the input maps in the output table.

Polygons to points
An option has been added to ignore or include undefined polygons in the output point map.

1.5 Miscellaneous

Table window
You can now edit multiple fields at the same time. Furthermore, pasting data from the clipboard into a table with domain `None` without active selections will add the pasted data to the table as records.

Import/Export
- All improvements in the three patches of ILWIS 2.1 are incorporated.
- Arc/Info .E00 import has been improved.
  - Split files (E00, E01, E02 etc) can now be read sequentially
  - Attributes are now imported into one or more attribute tables
  - Raster grid (GRD) can now be imported.
- Arc/View Shape import has been added.
- DXF import and export have been improved.
  - Import of DXF polygon maps now gives proper ILWIS polygons.
- TIFF import and export will now use GeoTIFF extensions.
- Import and Export are now available in the script and on the command line.

Dialog boxes
All dialog boxes now have COPY and PRINT commands in the system menu; under Windows 95 this is also reachable by using the right mouse button on the title bar.

DDE
DDE improvements have been added to let the calling application have a better control over the communication of ILWIS commands.

- Parentheses ( ) are no longer stripped from the command, so also definition statements can be issued from the DDE-client.
- Square brackets [ ] around commands are now optional.
- Ending a command with a semicolon will send the command directly to the script processor and only return after the command is finished. The semicolon can also be used to delimit commands on a line. The commands will be executed in sequence.

New script commands

- `additemtodomaingroup` *domainname upperlimit classname [classcode]*
- `begincomment`
  This is comment
  `endcomment`
- `message` *any text to display*
- `pause` *seconds*
- `show -noask` *mapname.ext*
- `crtbl` *tblname domain | nrrecs*
- `crmap` *mapname georef domain*
- `crpntmap` *mapname coordsys domain*
- `crsegmap` *mapname coordsys domain*
- Numerous commands to import files into ILWIS from another format or to export ILWIS objects to another format.

Chapter 2

Installing ILWIS 2.2

2.1 The ILWIS 2.2 package and related services

2.1.1 The ILWIS 2.2 package

The ILWIS 2.2 package contains the following items:

- sealed CD-ROM, containing the system and accessory data;
- ILWIS Hardware Key (not included when upgrades are involved);
- ILWIS 2.2 Guide (this manual);
- Beginner's Guide;
- User's Guide;
- Applications Guide;
- Reference Guide.

Upgrades only include the CD-ROM and the ILWIS 2.2 Guide.

If any of these items are missing or damaged, please contact your local distributor or the PCI-ILWIS Nederland B.V.
Before you break the seal of the CD-ROM, carefully read the terms of the ILWIS 2.2 software license agreement (Appendix G). By breaking the seal you signify that you have read this agreement and accept its terms.

2.1.2 Warranty

There is a 60-days warranty on the entire package, starting on the date the package leaves our premises. In the event of notification and return of the defective part(s) within the warranty period, your local distributor or PCI-ILWIS Nederland B.V. will replace the defective part(s).

2.1.3 ILWIS hardware key

The ILWIS 2.2 software package is protected against unauthorized use by a unique hardware key. ILWIS 2.2 software can be executed only when this hardware key is plugged in properly and you have entered the registration code (see 2.3: Installing ILWIS 2.2). Do not lose or damage the hardware key, without this key ILWIS 2.2 software can not be activated! Lost or damaged keys will not be replaced. Store the hardware key in a secure place when not in use. We advise you to insure the ILWIS hardware key for the price of the package. Furthermore, it is recommended to keep a copy of your registration code.

2.1.4 Technical support and software maintenance

PCI-ILWIS Nederland B.V. provides a Help Desk service to assist users in solving ILWIS related problems. However, before requiring this assistance, we advise users to consult the on-line help or documentation. Questions that require in-depth analysis should be reported through the Customers Services Report Form (Appendix H).

For up-to-date information concerning ILWIS you can browse to the web address: http://www.itc.nl/ilwis

The following E-mail addresses are available:

support.ilwis@itc.nl	When you want to know how to achieve a certain task or when you encounter problems, you can ask support.
wishes.ilwis@itc.nl	For suggestions and requests for new functionality; not for support.
bugs.ilwis@itc.nl	To report bugs; not for support.
ilwis@itc.nl	PCI-ILWIS Nederland B.V. E-mail address; for requests concerning the ILWIS package as a whole, e.g. pricing, etc.

ILWIS Users may obtain a Software Maintenance Contract (SMC). The SMC covers all package improvements released by PCI-ILWIS Nederland B.V. during the contract period. For more information on the SMC, contact your local distributor or PCI-ILWIS Nederland B.V.

2.2 Hardware and software requirements

2.2.1 Minimum hardware and software requirements

To ensure that ILWIS 2.2 will run under Windows you need at least a CPU with 80386 processor, 8 MB RAM, 256 colors graphics board with a resolution of 640×480, a CD-ROM player and a mouse. The Operating System should be Microsoft Windows 3.1 or later.

2.2.2 Recommended hardware and software requirements

For best performance of ILWIS 2.2 under Windows, we recommend using a Pentium class processor (CPU), 16 MB RAM or more and a 24 bit graphics Windows accelerator board, with a resolution of minimum 1024×768, a CD-ROM player and a mouse. The recommended Operating System is Microsoft Windows 95 or later.

2.2.3 Digitizers, printers and plotters

The most common method to enter spatial data in ILWIS is by digitizing a map or any other paper document (e.g. photo-interpretation). To be able to digitize in ILWIS, a digitizer tablet with a digitizer cursor with a minimum of 4 cursor buttons is needed. For digitizer installation and configuration, see section 2.4.

To be able to print or plot maps, graphs and tables, a printer or plotter is needed. For printer and plotter installation, see section 2.5.

2.2.4 Display drivers

ILWIS makes all graphic calls to Windows. Windows and the associated graphic display drivers then show the graphic output on the screen. When display problems are encountered, this may be related to your graphic drivers.
You can check display driver related issues in Win 3.1 by double-clicking the Windows Setup icon in the Windows Main program group (then under Options, choose Change System Settings); or in Windows 95, by choosing Start, Settings, Control Panel (then double-click Display icon and click the Settings tab).

- Make sure that you have installed the correct display driver. If you do not know which one to use, the option Standard display types should always work. If you experience any display driver errors, please contact your manufacturer.
- To work with ILWIS for Windows you should have a resolution of at least 256 colors (8 bits). Maps are then generally displayed in satisfactorily quality. When you work with 16 colors only, maps cannot be properly displayed. Optimal results are achieved when you use display drivers with a pixel depth of 16 or 24 bits (High Color or True Color).

2.2.5 Required hard disk space

For the installation of the ILWIS software, you will need approximately 20 MB of hard disk space.
For the *additional installation* of the Beginner's Guide data, you will need 2.7 MB, for the User's Guide data 65 MB, and for the Applications Guide data 100 MB. For the Georeference Orthophoto exercise you need 17 MB, and for the Geostatistics exercise 250 KB.
A full installation from CD-ROM thus requires approximately 205 MB.

2.3 Installing ILWIS 2.2

The set up program can be used to:
- Install ILWIS from diskette or CD-ROM.
- To add components that were not previously installed.

Before running the ILWIS set up program, it is strongly recommended that you exit all running programs.
When you have started the set up program, you can step through the different dialog boxes by clicking the Next and Back buttons. At any time you can to exit the setup program just by clicking the Cancel button.

2.3.1 Installation from CD-ROM

1. Place the ILWIS Hardware Key in one of the parallel ports. Printer devices can still use the selected port simultaneously.
2. Start up the computer.

3. Insert the CD-ROM into the CD-ROM drive and select Run from the Program Manager File menu (Windows 3.1/NT 3.51) or click the Start button and select Run… (Windows 95/98/NT4).
4. Type:
   `D:\setup.exe` (where D is the disk letter of your CD-ROM drive)
5. The installation starts, and the Welcome dialog box will open.
6. Read the text in this dialog box and click Next to move to the next dialog box.
7. Read the text in the Information dialog box and click Next to continue.
8. The Select Components dialog box opens. Select the components that you want to install by clicking the check box in front of the components. A checkmark appears in the check box when a component is selected. To deselect a component, click a check box again. Some components have sub-components that can be selected for installation individually. When a component consists of sub-components, the Change button will become active. Click this button to view the sub-components and select or deselect them.
   When you select a component and mark its check box, all sub-components are selected automatically. Likewise, removing the mark will deselect all sub-components. The Change button can then be used to customize the main setting. Click Next to continue.
9. The Select Directories dialog box is displayed. The installation offers default directories in which you can install ILWIS 2.2. Change the defaults or select other directories by using the Browse buttons to install all components (ILWIS program files, Beginner's Guide data, User's Guide data and Applications Guide data). Click the Next button.
10. The Select Program Folder dialog box opens. You can change the default name proposed to create a new program group (Windows 3.1/NT 3.51) or program folder (Windows 95/98/NT4). You can also choose to use an already existing program group or folder. Click the Next button.
    ILWIS 2.2 will now be installed. A progress bar appears indicating the status of installation. Pressing the Cancel button will interrupt the installation procedure.
    During installation ILWIS will try to add `ansi.sys` in the `config.sys` file if not yet included. The setup will search for the `ansi.sys` file on harddisk, and if found will ask the user to confirm addition to the `config.sys` file. A backup copy of the `config.sys` file will be made in case you confirm the addition. The line in `config.sys` will resemble the line:
    `device=c:\dos\ansi.sys` The `ansi.sys` driver is used by the Convert14 program. Although not required, the display of this DOS program will look much better with the driver loaded.
11. When the installation is successful the Setup Complete dialog opens. Select Yes, I want to view the Release Notes and Yes, I want to install the security key driver and Yes, I want to register the security key. Then click the Finish button.
    Note: Checking the option Yes, I want to register the security key will automatically select Yes, I want to install the security key driver.
12. The `readme` file now opens. Read this file and close it when you are done.

13. The security key installation will start a separate installation program that will install the correct driver for the hardware key. It will first ask to select a language to view the installation in. Then, you are asked to select a directory for the hardware key utilities. You can choose any directory you like.
14. The Register ILWIS Key program now starts. Carefully enter your registration code and click OK. The registration code is printed on the inside front cover of the User's Guide. Note that The code is case sensitive and must therefore be entered **exactly** as written, i.e. with small characters and capitals!
15. When you have installed ILWIS 2.2 correctly and entered your registration code correctly, you can now run ILWIS 2.2 by double-clicking the ILWIS 2.2 icon in the appropriate folder.

2.3.2 Add components to the current installation

If you did not install all ILWIS 2.2 components during the first installation, this can be done afterwards.
To add components:

1. Insert the CD-ROM and select Run from the Program Manager File menu (Windows 3.1/NT 3.51) or click the Start button and select Run… (Windows 95/98/NT4).
2. Type:
   `D:\setup.exe` (where `D` is the disk letter of your CD-ROM drive)
3. The Select Components dialog box opens. The set up program automatically recognizes previously installed components. Select the components that you want to add and click the Next button to continue.
4. The Select Directories dialog box opens. Modify the default directories to those into which you want to install the new components. Click the Next button to continue.
   If the setup detects that you *deselected* previously installed components, it will display the Confirm Delete Directory dialog. Accept removal of the previously installed components and install the *selected* components, or reject to continue without changing the contents of this directory.
5. The Select Program Folder dialog box opens. In this dialog box select the desired folder to add the ILWIS 2.2 program icons to, or select another already existing program folder to which the new program icons must be added. Click the Next button.
   The new components will now be installed. A progress bar appears indicating the status of installation. Pressing the Cancel button at any time will interrupt the installation procedure.
6. When the installation is successful, the Setup Complete dialog box opens. Select Yes, I want to view the Release Notes and click the Finish button to complete the setup procedure.
7. You can now run ILWIS 2.2 by double-clicking the ILWIS 2.2 icon in the appropriate folder.

2.3.3 Uninstalling the ILWIS 2.2 software

To uninstall ILWIS 2.2, double-click the uninstall icon in the ILWIS 2.2 program folder. All ILWIS 2.2 program files and data that has been added to your hard disk by the installation program will now be deleted. Data that has not been added by the installation program (e.g. data that you created) will not be deleted.
When you delete installed files by other methods, a message will appear when using uninstall afterwards, however this does not effect the uninstall procedure.

Note:
When you created data yourself based on the ILWIS data, make sure that dependency links of the new data are broken. Otherwise, this data may become corrupt after the uninstallation.

Before using the uninstall option, copy all installed data files that you do not want to remove to another directory.

☞ Note: To uninstall ILWIS 2.2, you need to use the same Windows version used to install ILWIS 2.2.

2.3.4 After installation

ILWIS icons

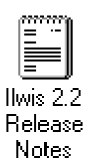
When a full installation is performed, i.e. when all ILWIS 2.2 components are installed, the new folder ILWIS 2.2 or the selected folder will contain the following icons:

Double-click this icon to start ILWIS 2.2, with the Beginner's Guide data directory as the Start up directory.

Double-click this icon to open the general ILWIS Help.

Double-click this icon to open the ILWIS Help file on operations.

Double-click this icon to open the ILWIS Help file on menu commands.



Double-click this icon to display the release notes.

Double-click this icon to start the ILWIS Hardware Key registration program.

Double-click this icon to start the uninstall program.

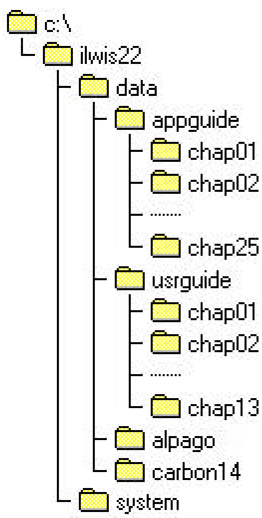
File structure after installation

The following files are added or changed (assuming all selections were the defaults):

- An ILWIS 2.2 directory is added to the root;
- In your Windows directory, an ILWIS.INI file is added;
- In the SYSTEM.INI file, a HARDLOCK.VXD virtual device driver is added. In addition, in the `[386ENH]` section a `device=` line is added (Windows 3.1 only).

ILWIS 2.2 directory structure

The ILWIS 2.2 directory tree has the following structure:



Where `c:\` is the drive and `ILWIS22` the directory in which you installed ILWIS 2.2. The files in each `chap` subdirectory correspond to the files needed for the chapters in the Applications Guide and the Users Guide.
If you did not perform a full installation, you will only find directories for those chapters you did install in the above structure.

2.3.5 Viewing ILWIS documentation PDF files

Most of the printed ILWIS documentation is also available on the CD-ROM in `.pdf format.` When you install Adobe Acrobat Reader, you can read all these files on your screen, exactly as they appear in the printer ILWIS Guides.

The `\doc` directory on the CD-ROM contains the text files of the ILWIS documentation, stored in `.pdf` format, as well as the Adobe Acrobat Reader 3.0. The Adobe Acrobat Reader 3.0 is used with permission of Adobe Systems Incorporated. The directory has the following structure:

`readpdf`	This directory contains the Adobe Acrobat reader for the `.pdf` files and consists of two subdirectories: `16bit`: for installing the Adobe Acrobat reader in Windows 3.1 `32bit`: for installing the Adobe Acrobat reader in all other Windows platforms.
`appguide`	The 25 chapters of the ILWIS Applications guide, stored as `appch00.pdf` (contents and preface), `appch01.pdf` (chapter 1) to `appch25.pdf` (chapter 25).
`bgnguide`	The file `bgnguide.pdf` contains the text of the ILWIS Beginner's guide.
`refguide`	This file contains the 8 chapters and some appendices of the ILWIS Reference Guide, stored as `refch01.pdf` (chapter 1) to `refch08.pdf` (chapter 8) and `refappen.pdf` (appendices).
`usrguide`	The 13 chapters of the ILWIS User's Guide, stored as `usrch00.pdf` (contents and preface), `usrch01.pdf` (chapter 1) to `usrch13.pdf` (chapter 13) and `usrindex.pdf` (index).

2.3.6 Installing the Adobe Acrobat PDF reader

1. Insert the CD-ROM into the CD-ROM drive and select Run from the File menu of the Program Manager (Windows 3.1/NT 3.51) or click the Start button and select Run… (Windows 95/98/NT4).
2. Type (for D substitute the disk letter of your CD-ROM drive):
   `D:\doc\16bit\setup` (Windows 3.1)
   `D:\doc\32bit\setup` (Windows 95/98/NT3.51/NT4)
3. Follow the instructions of the set up program.

The set up program will create a program group Adobe Acrobat on your hard disk, and it will associate `.pdf` files with the Adobe Acrobat Reader. After that you can open the `.pdf` files and look at their contents. There is no need to copy the `.pdf` files to the hard disk: they can be viewed from the CD-ROM. You can also print and copy the files or parts from it.

2.4 Installation and configuration of digitizers

The most common method to enter spatial data in ILWIS 2.2 is by digitizing a map or any other paper document (e.g. photo-interpretation). To use a digitizer, it must be properly installed and configured first. Depending on the type of digitizer used, the digitizer can be configured manually or with the use of a WinTab driver. The digitizer manufacturer usually supplies a WinTab driver. It is advised to use a WinTab driver, because it makes it easier to set up your digitizer and the digitizer can be used for other applications too.

2.4.1 Set up your digitizer using a WinTab driver

1. Connect all cables of the digitizer to the proper components.
2. Install the WinTab driver (see your digitizer manual or WinTab documentation).
3. Start ILWIS 2.2.
4. In the Main window, open the Options menu, choose Digitizer, Setup Digitizer. The Setup Digitizer dialog box appears.
5. Select the WinTab option to configure your digitizer.
6. You can now reference your map (see How to reference a map in the Reference Guide) and start digitizing (see How to digitize in Reference Guide).

☞ When you start Windows after you have installed the WinTab driver and the digitizer is disconnected or switched off, an error message is displayed because Windows can not detect the tablet. If you continue, the digitizer cannot be used, unless you connect and/or switch on the tablet and reboot your system.

2.4.2 Set up your digitizer manually

1. Connect all cables of the digitizer to the proper components.
2. Configure your digitizer, using the configuration software and/or the digitizer dip switches. Make sure that the digitizer:
   - is set in stream (run) mode (sometimes called continuous mode),
   - sends data in ASCII format,
   - gives Carriage Return (cr) and/or Line Feed (lf),
   - sends data using serial communication (rs-232-c),
   - supports Xon/Xoff signals.
3. Write down the configuration for the Baud Rate, Parity, Data Bits and Stop Bits and the Resolution (lines per inch or per mm). For more information, refer to your digitizer manual or section 2.4.3, where the settings for some widely available digitizers are described.
4. Up until this stage, you do not need ILWIS yet!
5. Start ILWIS 2.2.
6. In the Main window, open the Options menu, choose Digitizer, Setup Digitizer. The Setup Digitizer dialog appears.
7. Select the serial port (COM1 to COM4) to which your digitizer is connected. A message box now appears which lists the communication settings of the selected port (baud rate, parity, data bits, stop bits, flow control).
8. Make sure that these port settings are **exactly** the same as the communication settings of your digitizer you wrote down at step 3. If these settings are exactly the same press yes to open the Setup digitizer dialog box and continue with step 9. When the settings do not match press No to open the Control Panel (Windows 3.1) or the System Properties (Windows 95), and change the Port settings until they match the digitizer configuration. Go back to step 5.

9. When communication is established between the digitizer and the port, the coordinates are displayed as readable numbers that change when the digitizer cursor is moved. Continue with step 10. If this is not the case, check the set up of the digitizer tablet itself, and change the port settings accordingly to the digitizer settings until you have communication. When you do have communication, but not in readable numbers, it is most likely that you did not configure the digitizer in ASCII mode or the wrong port settings are selected. If so, reconfigure the digitizer in ASCII mode or change the port settings.
10. Follow the instructions for automatic set up; press the digitizer cursor buttons which you want to use as buttons 0, 1, 2 and 3 (Press the button you want to use as 0 button, when asked Press Button 0, etc.). Remember the sequence you have entered, because these are the buttons you have to press to execute the command following a 0, 1, 2 or 3 in the different Digitizer Menus of the editors.

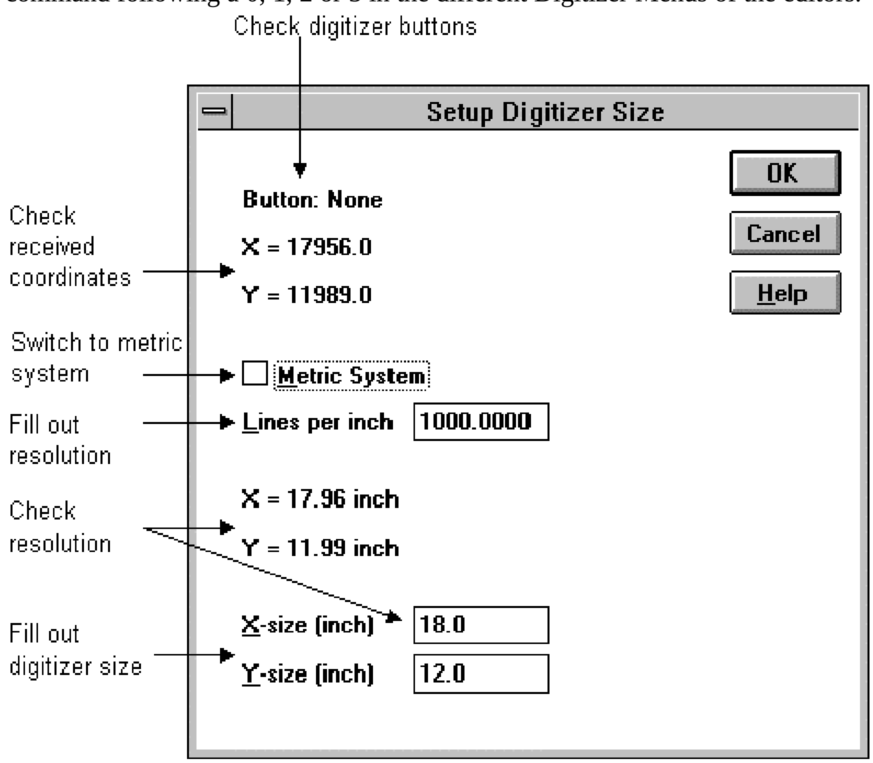


Figure 1: Setup digitizer size dialog

11. In the upper part of the Setup Digitizer Size dialog (see Figure 1): check if all buttons work, and check whether the X-values increase by moving the digitizer cursor to the right, and Y-values increase by moving the cursor up. If the communication is not correct, press Cancel to return to the previous dialog box and change until the communication is correct. When you have good communication, fill out the resolution (lines per inch or per mm) that the digitizer is using and the size of the digitizer. Remember to enter the same resolution, as you have configured for your digitizer (in step 3). You can check

the resolution by locating the digitizer cursor in the upper right corner of the tablet. The received X and Y coordinates (in mm or inch) mentioned under the Lines per mm (or inch) option should now be equal to the filled out digitizer size (at the bottom of the dialog). If the resolution is not correctly specified, an incorrect scale will be displayed during Map Referencing. Check the feedback you obtain in the lower part of the dialog and make changes until satisfied.

12. Reference your map (see How to reference a map in the Reference Guide) and start digitizing (see How to digitize in the Reference Guide).

Troubleshooting

If for some reason the automatic set up does not work, it is also possible to do the set up manually. Click the manual button in the Setup Digitizer dialog. You can now enter the format and the flag codes in the dialog. These can be obtained by analyzing the received coordinates in the second dialog. These contain the amount of characters, the position and amount of the X and Y characters and the flag codes. Move the cursor horizontally and vertically over the tablet to obtain X and Y position. The flag codes can be obtained by pressing the cursor buttons sequentially. The corresponding characters will show on the flag codes position.

Received Coordinates:
`03791,07676,0`

Means:

Format:	XXXX,YYYY,c (Format: X, Y, flag codes)
Flag Codes:	1248 (depending on the tablet you use)

And thus:

Nr chars:	the amount of characters in the format is 11
X Pos:	the start position of the first X character is 1
X Length:	the amount of X characters of the format is 4
Y Pos:	the start position of the first Y character is 6
Y Length:	the amount of Y characters of the format is 4
Flag Pos:	the position of the flag codes is 11
Flag Codes:	the flag codes are 1248

Note that all characters are counted as positions. You can also find the format and flag codes in your digitizer manual. You can now enter the format and flag codes in the dialog and click OK to continue with step 11.

2.4.3 Settings for some widely available digitizers

The next pages list possible configurations of some well known digitizers. Note that these are just suggestions; digitizers can often be configured for several data formats. This appendix only presents one of the possibilities; other configurations may well be possible. Note that the set up is not always automatically saved, this depends on the type of digitizer used. In some cases, it may well be possible that you have to reconfigure your digitizer every time you use ILWIS! Read your digitizer manual for more information.

☞ If your digitizer does not store the setup, put the setup command in the autoexec.bat.

CalComp 2000

Dip switch settings:

	1	2	3	4	5	6	7	8	9
S1	open	open	close	open	close	close	open	open	open
S2	open	close	open	open	close	close	open	open	close
S3	open	open	open	open	open	open	open	open	close

This configures the digitizer for the following set up:
Baud Rate 9600, Even Parity, 7 Data Bits, 1 Stop Bit and Resolution 1000 lines per inch.
Format: XXXX,YYYY,c
Flag Codes: 1248

CalComp Drawing Board 2

All sizes, using the set up menu.

Menu settings:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
bank A	1	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0
bank B	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

This configures the digitizer for the following set up:
Baud Rate 9600, Even Parity, 7 Data Bits, 2 Stop Bits and Resolution 1000 lines per inch.
Format: XXXXX,YYYYY,c
Flag Codes: 1248

CalComp Drawing Board 3

All sizes, using the set up menu.

Menu settings:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
bank A	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0
bank B	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
bank C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This configures the digitizer for the following set up:
Baud Rate 9600, Even Parity, 7 Data Bits, 2 Stop Bits and Resolution 1000 lines per inch.
Format: XXXXX,YYYYY,c
Flag Codes: 1248

Numonics 2205 (12”×12”) / 2206 (12”×17”)

Dip switch settings:

1	2	3	4	5	6	7	8
off	off	off	on	on	on	off	on

This configures the digitizer for the following set up:
Baud Rate 9600, Even Parity, 7 Data Bits, 2 Stop Bits and Resolution 200 lines per inch.
Format: XXXX,YYYY,c
Flag Codes: 1234

Numonics 2207

The program SETTAB.COM, which is a content of your digitizer package, is needed to configure your digitizer.
To configure:
Go in DOS to the directory where SETTAB.COM is located.
Type:
`SETTAB ASCII 7E COM1`
(enter)

The digitizer is now configured. This will set your digitizer on ASCII format, with 7 data bits and even parity on comport 1. Make sure that the tablet is attached to the specified Comport (here: COM 1). If the tablet is attached to another Comport, change the 1 in the formula in the Comport no. used (e.g. 2).
Then RESTART Windows (NOT your computer!) and start ILWIS.
This configures the digitizer for the following set up:
Comport 1, Baud Rate 9600, Even Parity, 7 Data Bits, 2 Stop Bits and Resolution 200 lines per inch.
Format: XXXX,YYYY,c
Flag Codes: 1248

Numonics GraphicMaster

To configure the digitizer for ILWIS:
This digitizer can be configured with Softkey Selection (See the digitizer manual). First align the Numonics set up Sheet with the bottom left side of the digitizer. Then remove the digitizer cursor. Install the cursor again, but when you do this, keep one cursor button pressed down. The set up Menu is now active. Select Baud Rate 9600, 8 Data Bits, None Parity, 1 Stop Bit, Enable XON/XOFF, Stream Mode, Stream Rate 20, Metric Resolution 10 (not higher, or the format changes) and Standard Formats S-BP1 ASCII.
Exit with Save Configuration.
This configures the digitizer for the following set up:
Baud Rate 9600, None Parity, 8 Data Bits, 1 Stop Bit and Metric Resolution 10 lines per mm.
Format: XXXX,YYYY,c
Flag Codes: 1248

Numonics GraphicMaster II

To configure the digitizer for ILWIS:
This digitizer can be configured with Softkey Selection (See the digitizer manual).

First align the Numonics set up Sheet with the bottom left side of the digitizer. Then place the cursor over the Softkey labelled set up. The beeper will emit a continuous series of beeps. Press and hold any cursor button, until the beeper stops. The set up Menu is now active. Select Standard Formats S-BPII ASCII, Baud Rate 9600, 7 Data Bits, Even Parity, 2 Stop Bit, Enable XON/XOFF, Stream Mode, Stream Rate 20, Metric Resolution 10 (not higher, or the format changes).
Exit with Confirm Change or Save to.
This configures the digitizer for the following set up:
Baud Rate 9600, Even Parity, 7 Data Bits, 2 Stop Bits and Resolution 200 lines per inch.
Format: XXXX,YYYY,c
Flag Codes: 1248

SummaSketch II

The SummaSketch does not have the possibility to store a configuration. The programs SEND.COM and MMRST.COM, which are a content of your digitizer package, are needed to configure your digitizer.
To configure:
Go in DOS to the directory where SEND.COM and MMRST.COM are located.
Type:

```
mmrst /1
send /1 /Cza@
```

(enter)

The digitizer is now configured. Make sure that the tablet is attached to the specified Comport (here: COM 1). If the tablet is attached to another Comport, change the 1 in the formula in the Comport no. used (e.g. 2).
This configures the digitizer for the following set up:
Comport 1, Baud Rate 9600, Odd Parity, 8 Data Bits, 1 Stop Bit and Resolution 20 lines per mm.
Format: XXXX,YYYY,c

2.5 Printer and plotter setup

In ILWIS a printer and/or plotter is necessary to generate output of maps and images. This chapter describes the steps to set up a printer or a plotter. Note that using a plotter under Windows may cause problems, we therefore recommend to create a plot file instead and send this to the plotter.
We will only describe the setup procedure for a printer, because the setup procedure for printer and plotter is identical. Simply read *plotter* instead of *printer* for a plotter setup.

Set up your printer:

1. Connect all cables of the printer to the proper components. Make sure that you connect a parallel printer to a parallel port and a serial printer to a serial port!
2. Install the printer driver (see your printer manual or Windows documentation).
3. Configure your printer using the printer configuration software (and/or dip switches). For more information about the configuration, see your printer manual.
4. Start ILWIS.
5. Open the Printer Setup dialog box: open the File menu of a graph window, table window or a map window, and select the option Printer Setup. Fill out the following options:
   - Default printer: Select this option when the printer that you want to use is set as the default, or
   - Specific printer: Select the printer you want to use; ILWIS simply lists installed printer drivers.
   - Orientation: Select the orientation of the paper in relation to the map(s) you want to print:
   - Portrait: Vertical paper orientation.
   - Landscape: Horizontal paper orientation.
   - Paper: Select the paper size and the source of the paper:
   - Paper size: Select the paper size for the selected printer, e.g. A4 or A3, or another paper size appearing in the drop-down list box. The paper sizes that appear depend on the printer driver.
   - Source: Select the paper source for the selected printer. The paper sources that appear in the drop-down list box depend on the printer driver.
   - Options button: Controls printer settings for the printer you select in the Specific Printers list. The available options vary depending on the printer driver. To get Help for the selected printer, choose the Options button, and then choose the Help button.
   - Network button: Connects your computer to a network printer. The Network button is available only if your computer is connected to a network. To get Help on network printers, choose the Network button, and then choose the Help button.
6. Click the ok button to finish the set up, or the cancel button to return without changes.
7. You can now print, see the Reference Guide : How to print maps, tables and annotation

Chapter 3

ILWIS window types

Note: This chapter is written as an enhancement to Chapter 3: ILWIS Window types in the ILWIS 2.1 Reference Guide. It does *not* completely replace the mentioned chapter: instead you are invited to use both as one.

3.1 Main window

3.1.1 Main window Menu commands

File	**Create**	Create Point Map ...
		Create Segment Map ...
		Create Raster Map ...
		Create Map List ...
		Create Table ...
		Create 2 Dimensional Table
		Create Domain ...
		Create Representation ...
		Create Georeference ...
		Create Coordinate System
		Create Sample Set ...
		Create Filter ...
		Create Function ...
		Create Script ...
	Open ...	
	Open as Table …	
	Open Pixel Information	
	Switch to	*select window*
	Close All	
	Minimize All	
	Restore All	
	Import ...	
	Export ...	
	Batch Import from 1.4 ...	
	Exit	
Edit	Edit Object ...	
	Properties …	
	Copy Object ...	
	Delete Object ...	

Operations	**Visualization**	Show Map ...	
		Show Table ...	
		Show Map List	as Color Composite ...
			as Slide Show …
		Display 3D ...	
		Apply 3D ...	
	Raster Operations	Map Calculate ...	
		Attribute Map ...	
		Cross ...	
		Aggregate Map ...	
		Distance ...	
		Iteration ...	
		Area Numbering ...	
		Sub Map ...	
		Glue Maps ...	
		Mirror Rotate ...	
	Image Processing	Filter ...	
		Stretch ...	
		Slicing ...	
		Color Separation ...	
		Color Composite …	
		Cluster ...	
		Sample ...	
		Classify ...	
		Resample ...	
	Statistics	Histogram ...	
		Raster	Autocorrelation ...
		Map List	Principal Components ...
			Factor Analysis ...
			Variance-Covariance ...
			Correlation Matrix ...
		Polygons	Neighbour Polygons ...
		Segments	Direction Histogram ...
		Points	Spatial Correlation ...
			Pattern Analysis ...
	Interpolation	Densify Map ...	
		Contour Interpolation ...	
		Point Interpolation	Nearest Point ...
			Moving Average ...
			Trend Surface ...
			Moving Surface ...
			Kriging …
	Vector Operations	Unique ID	
		Polygons	Attribute Map ...
			Mask Polygons ...
			Assign Labels ...
			Transform Polygons ...

		Segments	Attribute Map ... Mask Segments ... Assign Labels ... Sub Map ... Glue Maps ... Densify Coordinates ... Transform Segments ... Tunneling ...
		Points	Attribute Map ... Mask Points ... Sub Map ... Glue Maps ... Transform Points ... Transform Coordinates
	Rasterize	Polygon to Raster ... Segment to Raster ... Segment Density ... Point to Raster ... Point Density ...	
	Vectorize	Raster to Polygon ... Raster to Segment ... Raster to Point ... Polygon to Segment ... Polygon to Point ... Segment to Polygon ... Segment to Point ...	
	Table Operations	Transpose Table ... Change Domain … Table to Point Map … Glue Tables …	
Options	Catalog ... √ Active		
	Digitizer	Map Reference Set-up Digitizer	
	√ Command Line √ Operation-List √ Button Bar √ Status Line		
Help	Help on this Window Contents Search Map & Table Calculation ILWIS Operations ILWIS Expressions Menu Commands How to ... Index Glossary How to use Help About ILWIS		

Chapter 4

ILWIS objects

Note: This chapter is written as an enhancement to Chapter 4: ILWIS Objects in the ILWIS 2.1 Reference Guide. It does *not* completely replace the mentioned chapter: instead you are invited to use both as one.

Introduction

This chapter only contains those ILWIS objects for which the documentation has changed. Descriptions are included for the following objects:

- Data objects:
  - Raster maps
  - Polygon maps
  - Segment maps
  - Point maps
  - Map lists
- Service objects:
  - Representation
  - Georeferences
  - Coordinate systems
- Annotation:
  - Annotation Text
  - Simple annotation

For ILWIS objects **not** described here, refer to the ILWIS 2.1 Reference Guide. These ILWIS objects are:

- Data objects:
  - Tables
  - Columns
- Service objects:
  - Domains
- Special objects:
  - Map views
  - Histograms
  - Sample sets
  - Two-dimensional tables
  - Matrices
  - Filters
  - Functions
  - Scripts

4.1 Data objects

4.1.1 Raster maps

A raster map is a data object used to store spatial geographic information and remote sensing data as pixels (picture elements) of a certain size, e.g. 20 × 20 m. These pixels are either codified by IDs, class names, values or colors; this is determined by the domain of the map. The relation between pixels in a raster map and the position on earth is defined by the georeference that the raster map is using.

Raster maps can be displayed in a map window, and can be edited with the pixel editor. You can calculate with raster maps (MapCalc) and you can perform many other raster and image processing operations on them such as: Filter, Cross, Distance calculation, etc. In ILWIS, most spatial operations are performed on raster maps.

A raster map can be obtained:

- by importing an existing raster map from ILWIS version 1.41,
- by rasterizing an existing point, segment, or polygon map,
- by creating a new raster map and editing its pixels with the pixel editor,
- by using a satellite image which is already a raster map,
- by scanning a map or photograph and importing it into ILWIS.

As mentioned above, the output of spatial operations is usually another raster map.

When working with the multi-spectral bands of a satellite image or with multiple raster map of a time series, you can combine these bands or maps into a map list.

Contents of a raster map and location of pixels
The contents of a raster map are defined by the domain that the raster map is using. A raster map may store for example:

- rasterized individual point, segment or polygon features codified by IDs for instance cadastral plot IDs; the map uses an ID domain,
- rasterized classes of point, segment, or polygon features codified by class names, for example land use classes; the map uses a class domain,
- rasterized point, segment or polygon values for example height values; the map uses a value domain,
- reflection values in case of satellite images; the maps use the Image domain, the NOAA domain, or the Radar domain,
- true or false pixels; the map uses the Bit domain,
- true, false or undefined pixels; the map uses a Bool domain,
- colors in case of a scanned picture that was imported into ILWIS; the map uses a Picture domain, or the Color domain,
- output values after a map calculation (any domain),
- output values (any domain) after performing an operation, e.g. classes after the Classify operation (image classification), or values after a Distance calculation, or colors after creating a Color Composite,
- attribute values (any domain) when an attribute map was created from an attribute table,

- classified values, when a value map was classified (a group domain) with the Slicing operation.

For more information on domains, refer to Basic concepts : working with domains.

The spatial location of the pixels in a raster map is defined by the georeference that the raster map is using.

For maps using a class or ID domain, you can create an attribute table which stores additional information on the classes or IDs in the map. Use the same class or ID domain for the attribute table as you used for the map. Then, add the attribute columns to the table. To create an attribute table, refer to How to create an attribute table.

The manner in which raster maps are displayed is specified in a Display Options Raster Map dialog box.

- When a raster map uses a *class domain*, you can use or create a representation class. In a representation class, every class has its own color. You can edit the colors of a representation class in the Representation Class editor, or directly in a map window by using the Double-click action.
- When a raster map uses a *value domain*, you can use or create a representation value or a representation gradual.
- Raster maps that are satellite images and use the *Image domain*, are usually displayed with system representation Gray; the default stretch range is the 1% interval.
- When a raster map uses an *ID domain*, you can display the map in one single color or in multiple system colors. However, an ID raster map will usually have an attribute table and you can then choose to display the map by one of its attribute columns. For class columns, you can select or create a domain class; for value columns, you can select or create a representation value or a representation gradual, etc.
- When a raster map uses a *Bool domain*, you can specify colors for True and False pixels.

☞ To store the display setting of one or more maps displayed in a map window, save the map window as a map view; open the File menu in a map window and choose the Save View or the Save View As command.

Names of raster maps

Any raster map name must start with a character from A to Z. For the other characters in an object name, you are allowed to use: characters A through Z, numbers 0 through 9, and underscores. Object names cannot exceed 8 characters.

To create a raster map

Usually, you have to rasterize a point, segment or polygon map with the Points to Raster, Segments to Raster or Polygons to Raster operation. In the Rasterize process, you have to specify the dimensions of the raster map's pixels in the georeference of the raster map.

To display a raster map
The easiest way to display a raster map is to double-click the map in the Catalog. The map will be displayed in a new map window.

In the Display Options Raster Map dialog box, you can specify how the map should be displayed.

To edit a raster map
You can edit a raster map by clicking it with the right mouse button in the Catalog and subsequently choose Edit from the context-sensitive menu. When a raster map is already displayed in a map window, you can also choose the Edit Layer command from the Edit menu in the map window. For other methods, see How to edit point, segment, polygon and raster maps.

Raster maps are edited with the Pixel editor; you can edit the class names, IDs, or values of the pixels in the raster map. Dependent raster maps and read-only raster maps cannot be edited.

Operations on raster maps
You can calculate with raster maps by typing a Map Calculation formula on the command line of the Main window or in the Map Calculation dialog box. When the definition symbol = is used in a calculation, the output map is dependent.

You can also perform other operations on a raster map by selecting an operation from the Operation-list or from the Operations menu (for other methods, see How to start operations). Subsequently, fill out the appearing dialog box of that operation and click the OK button; this generates an ILWIS expression on the command line. Output objects that are obtained through an operation's dialog box are always dependent.
Advanced users can type the complete ILWIS expressions on the command line of the Main window or create a script to execute a series of expressions. For more information, see Appendices: ILWIS expressions.

When the output map of a calculation or another operation uses a value domain, you can usually define a value range and precision for the output map, see below. Operations on raster maps are for instance Filter, Cross and Distance calculation. For more information on operations, refer to ILWIS operations.

Technical information
A raster map consists of an ASCII object definition file (.MPR) and a binary data file (.MP#). The object definition file contains further references to the domain and the georeference that the raster map is using; these are properties of a raster map.

By viewing the properties of a raster map, you can see whether the map is dependent or not, which other objects the map is using, etc. In the properties dialog box of a value raster map, you can select the Interpolation option to obtain interpolated values on sub-pixel level. For dependent maps, you can manage dependencies: break dependencies, make dependent maps up-to-date, delete the dependent map's data file, etc.

The IDs, class names or values of pixels are stored line by line, starting from the first pixel (left) on the first line in the map (up), to the last pixel in the last line. Raster maps are stored as bit, byte, 2-byte, 4-byte or 8-byte maps. The store type of a raster map is primarily defined by the domain that the map is using. For value maps this is determined by the value range and precision that you specify for the output map during a calculation or operation. Usually, ILWIS will come up with defaults, but you can adjust these values. Choosing a wide value range and a very small precision results in a large data file on disk as more bytes are needed to store each pixel value in the map, also the speed of display and the speed of calculations may decrease.

For value maps that use 1 or 2 bytes per pixel, the map's histogram is automatically calculated when you open the map and a histogram is not available yet. For value maps that are stored using 4 or 8 bytes per pixel, you need to use the Histogram operation to obtain the map's histogram.

Limitations
The maximum number of lines in a raster map is 2 billion. The maximum number of columns is 32000 for bit, byte and 2-byte maps, 16000 for 4-byte maps and 8000 for 8-byte maps.

4.1.2 Polygon maps

A polygon map is a data object used to store spatial geographic information that consists of polygons, i.e. closed areas including the boundaries making up the areas. The areas are either codified by IDs, class names or values; this is determined by the domain of the map. The relation between polygons in a polygon map and the position on earth is defined by the coordinate system that the map is using.

Maps containing uniquely codified areas such as cadastral plots, or mapping units such as geological formations, land use classes, or soil units, can all be stored as polygon maps. Polygon maps are usually used as a stepping stone to raster maps. Polygon maps can be displayed in map windows, and edited with the polygon editor.

A polygon map can be obtained:
- by importing an existing polygon map from ILWIS version 1.41,
- by creating a segment map and editing it with the segment editor (with or without digitizer); then polygonize the segment map either in the segment editor or with the Segments to Polygons operation,
- by performing ILWIS operations that return a polygon map as output, for instance the Attribute map of polygon map operation.

Like in ILWIS 1.4, you first have to digitize segments and then polygonize these segments to obtain a polygon map.

Contents of a polygon map and location of polygons
The contents of a polygon map are defined by the domain that the polygon map is using. A polygon map may store for example:

- individual area features with or without attributes; each polygon has its own mostly unique ID, for instance cadastral plot IDs; the map uses an ID domain,
- classes of area features with or without attributes; polygons may occur in several places in the map, for instance land use classes or soil classes; the map uses a class domain,
- area features representing measurable values; polygon values representing for instance population densities or concentration levels of air pollution; the map uses a value domain,
- true, false or undefined polygons; the map uses a Bool domain,
- output values (any domain) after performing an operation, e.g. IDs after the Unique ID operation,
- attribute values (any domain) when an attribute map was created from an attribute table with the Attribute map of polygon map operation.

For more information on domains, refer to Basic concepts : working with domains.

The spatial location of polygons in a polygon map is defined by the coordinate system that the polygon map is using.

For maps using a class or ID domain, you can create an attribute table which stores additional information on the classes or IDs in the map. Use the same class or ID domain for the attribute table as you used for the map. Then, add the attribute columns to the table. When you have for example a map with building blocks (coded by IDs), you can add an attribute table with the predominant landuse per building block and the number of residents per building block. Or, when you have a soil map (coded by classes), you can add an attribute table with pH, soil texture, etc. for each soil unit.
To create an attribute table, refer to How to create an attribute table.

The manner in which polygon maps are displayed is specified in the Display Options Polygon Map dialog box.

- When a polygon map uses a *class domain*, you can display polygons in a single color, or by using or creating a representation class. In a representation class, every class can have its own color, and optionally you can use hatchings or patterns.
- When a polygon map uses a *value domain*, you can display polygons in a single color or you can use or create a representation value or a representation gradual.
- When a polygon map uses an *ID domain*, you can display polygons in a single color or in multiple system colors. However, an ID polygon map will usually have an attribute table and you can then choose to display the map by one of its attribute columns. For class columns, you can select or create a domain class; for value columns, you can select or create a representation value or a representation gradual, etc. For more information, see How to display a map by one of its attributes.

- When a polygon map uses a *Bool domain*, you have to specify colors each time you display the map.

☞ To store the display setting of one or more maps displayed in a map window, save the map window as a map view; open the File menu in a map window and choose the Save View or the Save View As command.

Names of polygon maps
Any polygon map name must start with a character from A to Z. For the other characters in an object name, you are allowed to use: characters A through Z, numbers 0 through 9, and underscores. Object names cannot exceed 8 characters.

To create a polygon map
To create a polygon map, you first have to create a segment map in which you can digitize polygon boundaries. Furthermore, you can create a point map in which you can digitize the polygon names or IDs as label points. When you display both the segment and point map in one map window, you can easily see whether the areas enclosed by segments have a label point. Then, polygonize the segments from within the segment editor. For more information, refer to How to create polygon maps or to Segment editor. To check after polygonization whether all polygons obtained a class name, ID or value, you can use the Polygons to Points operation.

To display a polygon map
The easiest way to display a polygon map is to double-click the map in the Catalog. The polygon map will be displayed in a new map window. You can also drag a polygon map from the Catalog to an existing map window, in order to show that polygon map on top of other information shown in the map window.

In the Display Options Polygon Map dialog box, you can specify how the map should be displayed.

To edit a polygon map
Class names, IDs, or values of selected polygons can be edited in the polygon editor. To change the shape of polygons, or to delete or undelete polygon boundaries you need to use the segment editor; then re-polygonize the segment map.

To open the polygon editor, you can click a polygon map with the right mouse button in the Catalog and subsequently choose Edit from the context-sensitive menu. When a polygon map is already displayed in a map window, you can also choose the Edit Layer command from the Edit menu in the map window. For other methods, see How to edit point, segment, polygon and raster maps. In the polygon editor, you can use a digitizer but you can also use the mouse pointer. Dependent polygon maps and read-only polygon maps cannot be edited.

Operations on polygon maps
The most frequently used operation on polygon maps is probably the Polygons to Raster operation, which rasterizes a polygon map. During this operation, you have to select or create a georeference for the raster map to define the size of the pixels in

the raster map. Another operation on polygon maps is for instance Attribute map of polygon map.

You can perform operations on a polygon map by selecting an operation from the Operation-list or from the Operations menu (for other methods, see How to start operations). Subsequently, fill out the appearing dialog box of that operation and click the OK button; this generates an ILWIS expression on the command line. Output objects that are obtained through an operation's dialog box are always dependent. Advanced users can type the complete ILWIS expressions on the command line of the Main window or create a script to execute a series of expressions.

For more information on operations, refer to Chapter 7: Operations.

Technical information
A polygon map consists of an ASCII object definition file (.MPA) and a number of binary data files (.PC#, .PD#, .PL#, .PS#, .TP#). The object definition file contains further references to the domain and the coordinate system that the polygon map is using; these are properties of a polygon map.

By viewing the properties of a polygon map, you can see whether the map is dependent or not, which other objects the map is using, etc. For dependent maps, you can also manage dependencies: break dependencies, make dependent maps up-to-date, delete the dependent maps's data files, etc.

A polygon map stores one or more polygons. A polygon is nothing more than a list of XY-coordinate pairs (the polygon boundaries) which form the boundaries of a polygon. The first and last coordinate pair of a boundary is called a node, so all individual polygon boundaries are connected to each other by nodes. Each polygon boundary is used both for the polygon on the left and right of this boundary; this is called a topological structure. Polygons are identified by IDs, class names or values.

Coordinate pairs are stored as pairs of 2-byte fixed point reals. This means that a paper map with a size of 3 by 3 meter can be stored with a resolution of 0.1 millimeter on paper. For all practical purposes, this should be enough. Paper maps smaller than 3 by 3 meters can thus be stored with an even finer resolution. When you create a map and specify the coordinate boundaries of the map, you should take care that the boundaries do not exceed this size of 3 by 3 meter on paper.

Limitations
The maximum number of polygons in one polygon map is 32000. The maximum number of polygon boundaries is 32000. The maximum number of coordinate pairs in one polygon boundary is 1000 (taken care of by the system).

4.1.3 Segment maps

A segment map is a data object used to store spatial geographic information that consists of lines, for example roads, rivers or contour lines. Segments are either codified by IDs, class names or values (height map); this is determined by the domain of the map. The relation between segments in a segment map and the position on earth is defined by the coordinate system that the map is using. Segment maps can be displayed in map windows, and edited with the segment editor.

A segment map can be obtained:
- by importing an existing segment map from ILWIS version 1.41,
- by creating a new segment map and editing it with the segment editor with or without using a digitizer, or
- by performing ILWIS operations that return a segment map as output, for instance the Mask Segments operation.

Contents of a segment map and location of segments
The contents of a segment map are defined by the domain that the segment map is using. A segment map may store for example:
- individual line features: each segment is codified by its own ID for instance when each pipe line or river section has its own unique ID; the map uses an ID domain,
- classes of line features: segments may occur in several places in the map for instance primary roads, secondary roads, etc.; the map uses a class domain,
- line features representing measurable values, for instance a contour map with height information; the map uses a value domain,
- true, false or undefined segments; the map uses a Bool domain,
- output values (any domain) after performing an operation, e.g. IDs after the Unique ID operation,
- attribute values (any domain) when an attribute map was created from an attribute table with the Attribute map of segment map operation.

For more information on domains, refer to Basic concepts: working with domains.

The spatial location of segments in a segment map is defined by the coordinate system that the segment map is using.

For maps using a class or ID domain, you can create an attribute table which stores additional information on the classes or IDs in the map. Use the same class or ID domain for the attribute table as you used for the map. Then, add the attribute columns to the table. To create an attribute table, refer to How to create an attribute table.

The manner in which segments are displayed is specified in the Display Options Segment Map dialog box.
- When a segment map uses a *class domain*, you can display segments in a single color, or by using or creating a representation class. In a representation class, every class can have its own line type, line width, color, etc. You can also represent a line by equally spaced symbols. For more information, see Representation Class editor and the Edit Repr Class item (segment) dialog box.
- When a segment map uses a *value domain*, you can display segments in a single color, in multiple system colors or by using or creating a representation value or a representation gradual.

- When a segment map uses an *ID domain* or a *Bool domain*, you have to specify colors each time you display the map.

☞ To store the display setting of one or more maps displayed in a map window, save the map window as a map view; open the File menu in a map window and choose the Save View or the Save View As command.

Names of segment maps
Any segment map name must start with a character from A to Z. For the other characters in an object name, you are allowed to use: characters A through Z, numbers 0 through 9, and underscores. Object names cannot exceed 8 characters.

To create a segment map
Open the File menu of the Main window and select the Create Segment Map command, or double-click the New SegMap item in the Operation-list. In the appearing Create Segment Map dialog box, enter a name and description for the segment map, select or create a coordinate system, enter boundary values for the map and select or create a domain. For other create methods, see How to create a map.

To display a segment map
The easiest way to display a segment map is to double-click the map in the Catalog. The map will be displayed in a new map window. You can also drag a segment map from the Catalog to an existing map window, in order to show that segment map on top of other information shown in the map window. For other methods, see How to display maps and tables.

In the Display Options Segment Map dialog box, you can specify how the map should be displayed.

To edit a segment map
You can edit a segment map by clicking it with the right mouse button in the Catalog and subsequently choose Edit from the context-sensitive menu. When a segment map is already displayed in a map window, you can also choose the Edit Layer command from the Edit menu in the map window. For other methods, see How to edit point, segment, polygon and raster maps.

Segment maps are edited with the Segment editor: you can insert new segments and delete and undelete existing ones, edit the class names, IDs, or values of segments, change the shape of segments, split and merge segments, and pack, check and polygonize segments. You can use a digitizer but you can also use the mouse pointer. Dependent segment maps and read-only segment maps cannot be edited.

Operations on segment maps
Frequently used operations on segment maps are for instance the Segments to Polygons operation to polygonize a segment map and the Segments to Raster operation, which rasterizes a segment map. During the rasterize operation, you have to select or create a georeference for the raster map to define the size of the pixels in

the raster map. Another operation on a segment map is for instance Contour interpolation.

You can perform operations on a segment map by selecting an operation from the Operation-list or from the Operations menu (for other methods, see How to start operations). Subsequently, fill out the appearing dialog box of that operation and click the OK button; this generates an ILWIS expression on the command line. Output objects that are obtained through an operation's dialog box are always dependent. Advanced users can type the complete ILWIS expressions on the command line of the Main window or create a script to execute a series of expressions.

For more information on operations, refer to Chapter 7: Operations.

Technical information
A segment map consists of an ASCII object definition file (.MPS) and three binary data files (.CD#, .SC#, .SG#). The object definition file contains further references to the domain and the coordinate system that the segment map is using; these are properties of a segment map.

By viewing the properties of a segment map, you can see whether the map is dependent or not, which other objects the map is using, etc. For dependent maps, you can also manage dependencies: break dependencies, make dependent maps up-to-date, delete the dependent maps's data files, etc.

A segment map stores one or more segments. A segment is nothing more than a list of XY-coordinate pairs. Each list of XY-coordinates represents a (part of a) linear feature. Segments are identified by IDs, class names or values. Places where two or more segments are connected are called **nodes**: a node is thus a common XY-coordinate for two or more segments. Further, the beginning and end of a segment are always nodes.

Coordinate pairs are stored as pairs of 2-byte fixed-point reals. This means that a paper map with a size of 3 by 3 meter can be stored with a resolution of 0.1 millimeter on paper. For all practical purposes, this should be enough. Paper maps smaller than 3 by 3 meters can thus be stored with an even finer resolution. When you create a map and specify the coordinate boundaries of the map, you should take care that the boundaries do not exceed this size of 3 by 3 meter on paper.

Limitations
The maximum number of segments in one segment map is 32000. The maximum number of coordinate pairs per segment is 1000 (taken care of by the system).

4.1.4 Point maps

A point map is a data object used to store spatial geographic information which consists of points, for example water wells or sample points. Points are either identified by IDs, class names or values; this is determined by the domain of the map. The relation between points in a point map and the position on earth is defined by the coordinate system that the map is using. Point maps can be displayed in map windows, and edited with the point editor.

A point map is obtained:

- by importing an existing point map from ILWIS version 1.41,
- by creating a new point map and editing it with the point editor with or without using a digitizer,
- by converting a table which contains coordinate columns to a point map with the Table to Point map operation, or
- by performing ILWIS operations that return a point map as output, for instance the Attribute map of point map operation, or the Segments to Points operation.

Contents of a point map and location of points

The contents of a point map are defined by the domain that the point map is using. A point map may store for example:

- individual point features: each point is codified by its own ID, for instance individual sample points, individual bore holes, individual rainfall stations, etc.; the map uses an ID domain,
- classes of point features: points may occur in several places in the map for instance all water wells of a certain type; the map uses a class domain,
- point features representing measurable values, for instance height information; the map uses a value domain,
- true, false or undefined points; the map uses a Bool domain.
- output values (any domain) after performing an operation, e.g. IDs after the Unique ID operation,
- attribute values (any domain) when an attribute map was created from an attribute table with the Attribute map of point map operation.

For more information on domains, refer to Basic concepts : working with domains.

The spatial location of points in a point map is defined by the coordinate system that the point map is using.

For maps using a class or ID domain, you can create an attribute table which stores additional information on the classes or IDs in the map. Use the same class or ID domain for the attribute table as you used for the map. Then, add the attribute columns to the table. When you have for a example a point map with rainfall stations, coded with IDs, you can add an attribute table with the rainfall figures per month for each station. To create an attribute table, refer to How to create an attribute table.

Point symbols

The manner in which points are displayed is specified in the Display Options Point Map dialog box.

- When a point map uses a *class* domain, you can display point symbols as single symbols or by using or creating a representation class. In a representation class, every class can have its own symbol, size, color, etc. For more information, see Representation Class editor and the Edit Repr Class item (point) dialog box.

- When a point map uses an *ID domain* and when the point map has an *attribute table*, you can display point symbols as single symbols, as arrows, as graphs or by attributes. In this way, you can for instance obtain pie charts, bar graphs, volume cubes, etc. for every point in the map. Sizes, colors, etc. can be specified by using columns in the attribute table.
- When a point map uses a *value domain*, you can choose a symbol set and a symbol for all the points while the size of the selected symbol may depend on the value of the points.
- When a point map uses a *Bool domain*, you can specify colors for True and False points.

☞ To store the display setting of one or more maps displayed in a map window, save the map window as a map view; open the File menu in a map window and choose the Save View or the Save View As command.

Names of point maps
Any point map name must start with a character from A to Z. For the other characters in an object name, you are allowed to use: characters A through Z, numbers 0 through 9, and underscores. Object names cannot exceed 8 characters.

To create a point map
Open the File menu of the Main window and select the Create Point Map command, or double-click the New PntMap item in the Operation-list. In the appearing Create Point Map dialog box, enter a name and description for the point map, select or create a coordinate system, enter boundary values for the map and select or create a domain. For other create methods, see How to create a map.

When you have a table, which contains columns with X- and Y-coordinates, you can use the Table to Point map operation to obtain a point map from this table.

To display a point map
The easiest way to display a point map is to double-click the map in the Catalog. The map will be displayed in a new map window. You can also drag a point map from the Catalog to an existing map window, in order to show that point map on top of other information shown in the map window. For other methods, see How to display maps and tables.

In the Display Options Point Map dialog box, you can specify how the map should be displayed.

Point maps can also be opened as tables: click the point map with the right mouse button in the Catalog, and subsequently choose Open as Table from the context-sensitive menu. When you display a point map in a table window, you can edit the XY-coordinates of the points and perform calculations with the XY-coordinates of the points.

To edit a point map
You can edit a point map by clicking it with the right mouse button in the Catalog and subsequently choose Edit from the context-sensitive menu. When a point map is already displayed in a map window, you can also choose the Edit Layer command from the Edit menu in the map window. For other methods, see How to edit point, segment, polygon and raster maps.

Point maps are edited with the Point editor: you can insert new points and delete existing ones, edit the class names, IDs, or values of points, and move points to a new position. You can use a digitizer but you can also use the mouse pointer. Dependent point maps and read-only point maps cannot be edited.

Operations on point maps
Examples of operations on point maps are for instance point interpolations: moving average, trend surface, moving surface, or kriging. The point map itself may use a value domain but you can also use point maps with a class or ID domain and that have an attribute table, which contains the values, which you want to interpolate.

You can perform operations on a point map by selecting an operation from the Operation-list or from the Operations menu (for other methods, see How to start operations). Subsequently, fill out the appearing dialog box of that operation and click the OK button; this generates an ILWIS expression on the command line. Output objects that are obtained through an operation's dialog box are always dependent. Advanced users can type the complete ILWIS expressions on the command line of the Main window or create a script to execute a series of expressions.

For more information on operations, refer to Chapter 7: Operations.

Technical information
A point map consists of an ASCII object definition file (.MPP) and a binary data file (.PN#). The object definition file contains further references to the domain and the coordinate system that the point map is using; these are properties of a point map.

By viewing the properties of a point map, you can see whether the map is dependent or not, which other objects the map is using, etc. For dependent maps, you can also manage dependencies: break dependencies, make dependent maps up-to-date, delete the dependent maps's data files, etc.

Each point is stored as a pair of XY-coordinates. The coordinates pairs are stored as pairs of 8-byte floating point reals. Points are identified by a class name, ID or a value.

Limitations
The maximum number of points in a point map is 2 billion.

4.1.5 Map lists

A map list is a data object and stores the names of a set of raster maps, for example of the multi-spectral bands of a satellite image that you want to classify. A map list may also store the names of several raster maps of a time series which you can display as a slide show. All raster maps in a map list must have the same the same domain and georeference.

A map list is used:

- for sampling and image classification,
- for the creation of interactive color composites,
- for a principal components analysis or factor analysis,
- for the calculation of multiband statistics (variance-covariance matrix or correlation matrix),
- to present temporal changes in maps; make a time-series by combining several maps of the same theme but of for instance different years into a map list and show the map list as a slide show.

Names of map lists

Any map list name must start with a character between A and Z. For the other characters in an object name, you are allowed to use: characters A through Z, numbers 0 through 9, and underscores. Object names cannot exceed 8 characters.

To create a map list

Open the File menu in the Main window and select the Create Map List command, or double-click the NewMapList item in the Operation-list, or start an operation which requires a map list as input and click the create button in the operation's dialog box. In the appearing Create Map List dialog box, enter a name and description for the map list, and select the raster maps that together should form the map list.

To display a map list

A map list is simply a list of raster map names. You can display the individual raster maps as usual, for instance by double-clicking a raster map in the Catalog; the map is displayed in a new map window. To open and edit the contents of a map list, see paragraph To edit a map list.

When your map lists contains raster maps, which have the Image domain or another value domain, and your graphics board is configured to use more than 256 colors, e.g. High Color 16-bit or True Color 24-bit, you can display three maps of a map list as an interactive color composite. Choose Visualization, Show Map List as Color Composite from the Operations menu and select a map list, or use the right mouse button on a map list in the Catalog and choose Visualization, as Color Composite from the context-sensitive menu. You can select the maps of the map lists, which should be used in the color composite and you can specify the stretch ranges for each of the selected maps. By changing the display options of the map list, you can change the color composite. This is very suitable for sampling and on-screen digitizing. For more information, refer to How to create a color composite.

Furthermore, all maps in a map list can be displayed one after the other in one map window, as a slide show. This is very suitable to display a time series. Choose Visualization, Show Map List as Slide Show from the Operations menu, or use the right mouse button on a map list in the Catalog and choose Visualization, as Slide Show from the context-sensitive menu. For more information on slide shows, see How to display a map list as a slide show or Map window: advanced functionality.

To edit a map list
To edit a map list, select Edit Object from the Edit menu in the Main window and select a map list, or use the right mouse button on a map list in the Catalog and choose Edit from the context-sensitive menu. In the appearing Edit Map List dialog box you can change the selection of raster maps that are included in the map list. See also How to edit a map list.

Operations on map lists
Usually, you will create a map list during sampling to obtain a sample set with which you can perform a supervised classification in a later stage. Sampling however is an interactive process and not a genuine operation.

Operations on a map list return multivariate statistical information (a matrix) on the values of the different raster maps in a map list. Only the Principal Components and Factor Analysis operations result in an output map list.

You can start an operation on a map list for instance by selecting an operation from the Operation-list or from the Operations menu. For other methods, see How to start operations. Subsequently, fill out the appearing dialog box of that operation and click the OK button; this generates an ILWIS expression on the command line. Output matrices and output raster maps that are obtained through an operation's dialog box are always dependent; however output map lists are not dependent yet. Advanced users can type the complete ILWIS expressions on the command line of the Main window or create a script to execute a series of expressions.

Technical information
A map list consists only of an ASCII object definition file (.MPL). The object definition file contains further references to raster maps that are included in the map list.

By viewing the properties of a map list, you can see the size of the object and the time it was last changed. In case you calculated a variance-covariance matrix or a correlation matrix, the properties dialog box contains an extra button Additional Info, which shows you the Optimum Index Factor (OIF) of combinations of three bands in the map list. For more information, see How to calculate the Optimum Index Factor.

Limitations
In principle, the maximum number of raster maps that can be included in a map list is 5440. In practice however limitations will occur by memory and available disk

space. For operations that result in a matrix, a map list cannot contain more than 80 raster maps.

4.2 Service objects

4.2.1 Representations

A representation defines the manner in which classes of a map with a class domain, a group domain or a picture domain, or the values of a map with a value domain or the image domain should be represented on the screen and on a printer.

A representation stores colors or ranges of colors for the classes in a class domain, or for specific values or ranges of values for the values in a value domain. Furthermore, for polygon classes, colors, hatching and patterns can be stored; for segment classes, colors, line types and line widths, etc. can be stored; and for point classes, colors, symbols, symbol sizes, etc. can be stored.

A representation is a service object for a domain, i.e. a domain uses a certain representation.

Maps with a Bool, the `Bit`, an ID, or the Color domain do *not* have a stored representation on disk:

- For maps with a Bool domain or the `Bit` domain, you can interactively select colors in the Display Options dialog box of the map each time you display the map.
- Maps with an ID domain can be shown in 7, 15, or 31 colors to inspect the map itself. In most cases however, you will display an ID map by one of its attributes as stored in the map's attribute table; to do so, select the Attribute check box in the Display Options dialog box of the ID map and select an attribute column.
- Raster maps with the Color domain store colors in each pixel, so no representation is needed.

Furthermore, special options for point symbols available in the Display Options dialog box of a point map.

☞ You can always create a number of representations for a certain domain, for instance by clicking the create button next to the Representation list box in the Display Options dialog box of a map. Then, to display the map with a certain representation, in the map's Display Options dialog box, select the representation you want to use.

Types of representations

There are three types of representations:

- **representation class**: for the classes in a class domain, the groups a group domain, or the items in a picture domain. For each class in the domain, a representation class contains: colors for mapping units in raster maps; colors, hatching or patterns for polygons; colors, line types, line widths, etc. or equally spaced symbols for segments; colors, symbol type, symbol sizes, etc. for points.

- **representation gradual**: defines colors for ranges of values (as *percentages*) in a value map. Any map with a value domain can be displayed with an existing representation gradual. Maps, which use system domain `Value,` can only be displayed with a representation gradual; you can also create a representation gradual for such maps. Examples of a representation gradual are `Gray` and `Pseudo`. To change the domain of a map, which uses system domain Value to a user-defined value domain, in order to create a representation value, see How to change the domain of maps.
- **representation value**: defines colors for ranges of values (as *values*) in a value map. A representation value can be created for maps with the `Image` domain (interactive slicing), for maps with a *user-defined value domain* (e.g. a `Height` domain), and for maps with a system value domain except system domain `Value`. You can create various value representations for individual value domains.

General use of representations

- All maps which use the same class domain (e.g. `Landuse`) will by default use the same representation class: these maps will thus by default be displayed in the same colors. You can also create or select another representation class, which fits the classes in the domain.
- All maps which use the same value domain (e.g. `Height`) will by default use the same representation value: these maps will thus by default be displayed in the same colors. Except for maps, which use system domain `Value`, you can also create or select another representation value, which fits this domain. Furthermore, you can select a representation gradual e.g. `Pseudo`.
- Maps, which use the system domain `Value,` can only be shown in a representation gradual. You can select any existing representation gradual, or create your own representation gradual.

Names of representations

Any representation name must start with a character from A to Z. For the other characters in an object name, you are allowed to use: characters A through Z, numbers 0 through 9, and underscores. Object names cannot exceed 8 characters.

☞ To show representations in the Catalog, refer to How to customize the Catalog.
☞ To view which representation is linked to a certain domain, check the properties of the domain.

To create a representation

1. Creating a representation class:
   When creating a new class or group domain, a representation class is automatically created with default colors; these can of course be edited.

   The easiest way to create another representation class for a map with a class, group or picture domain, is to click the create button in the Display Options dialog box of the map. Then a new representation class is created for the domain that is used by the map.
   The Representation Class editor will be opened.

In the Representation Class editor, choose whether you want to adapt the representation for the raster form, the polygon form, the segment form or the point form. Then, edit the colors, etc. of each class for instance by double-clicking individual classes. You can also select multiple classes and choose Edit Multiple Items from the Edit menu to assign one color or a smooth range of colors to the selected classes.

2. Creating a representation value:
When creating a new value domain, a representation value is not automatically created. You can always show a map with a value domain with an existing representation gradual, e.g. `Pseudo`.

The easiest way to create a representation value for a map with a value domain (except the system domain `Value`) is to click the create button in the Display Options dialog box of the map. Then a new *representation value* is created for the domain that is used by the map. The Representation Value/Gradual editor will be opened.

In the Representation Value/Gradual editor, you can insert limits, i.e. *boundary values* for a representation value, you can choose or create a color for each limit, and you can indicate whether to stretch between the limits or to use the color of the upper of lower boundary.

3. Creating a representation gradual:
When you click the create button in the Display Options dialog box of a map which uses the system domain `Value`, a new *representation gradual* is created with default colors. Furthermore, a representation gradual will be created when you select system domain `Value` in the Create Representation dialog box. The Representation Value/Gradual editor is opened.

In the Representation Value/Gradual editor, you can insert limits, i.e. *boundary percentages* for a representation gradual, you can choose or create a color for each limit, and you can indicate whether to stretch between the limits or to use the color of the upper of lower boundary.

When a map (e.g. `Map1`) is currently using system domain `Value`, and thus a representation gradual, and you want to use a representation value instead, you have to change the domain of the map with system domain `Value` to a user-defined domain:
- Perform a simple mapcalc statement like `Map2 = Map1;`
- Create a new value domain in the Raster Map Definition dialog box by clicking the create domain button;
- Then display `Map2` and in the map's Display Options dialog box, click the create representation button. You have created a representation value and you can assign colors to specific values in the map.

For more information, see How to change the domain of a map.

4. Other ways to create representations:
Furthermore, you can create a representation through the File menu of the Main window, by double-clicking the New Repr item in the Operation-list, or via the Properties dialog box of a domain. For more information, see How to create a representation.

To view and edit a representation
To edit an existing representation, you can for instance double-click a representation in the Catalog. When a map is displayed in a map window, you can also open the Edit menu in the map window and choose Representation. Depending on the type of representation you selected, the Representation Class editor or the Representation Value/Gradual editor is opened. For more information, see How to edit a representation or the respective editors.

Advanced users may wish to open and/or edit a representation class as a table. For more information, see How to open objects as a table. System representations and read-only representations cannot be edited.

Technical information
A representation consists of an ASCII object definition file (.RPR); in case of a representation class also a binary data file (.RP#) is available. The object definition file has no further references to other objects.

By viewing the properties of a representation, you can see for instance the type of the representation and you can find out to which domain the representation is linked. A representation is a property of a domain.

A **representation class** stores, for each class in the domain:
- a color (raster form);
- a color, hatching or pattern (polygon form);
- a color, a line type (e.g. single, double, triple, dot, dash dot, blocked), line width, optional support line, optional background color for the support line, and a symbol font and a symbol when a line should be represented by symbols (segment form);
- symbol type, symbol, size, line width, color, fill color, rotation (point form).

Advanced users may wish to open and edit a representation class as a table.

A **representation gradual** stores:
- limits, i.e. boundary values *as percentages* of the values in the domain,
- a color for each limit,
- for each range of percentages in between two limits, whether to stretch colors or to use the color of the upper of lower boundary (slice), and
- when you stretch the colors in between two limits: the number of steps to be used for stretching.

A **representation value** stores:
- limits, i.e. boundary values *as values* of the domain,

- a color for each limit,
- for each range of values in between two limits, whether to stretch colors or to use the color of the upper of lower boundary (slice), and
- when you stretch the colors in between two limits: the number of steps to be used for stretching.

System representations
The following system representations are *representations gradual*:

`Blue`	From black to blue in 16 stretch steps.
`Clrstp6`	Six times from black to a color (6 colors between blue to red), each time in 8 stretch steps.
`Clrstp8`	Eight times from black to a color (8 colors between blue to red), each time in 8 stretch steps.
`Clrstp10`	Ten times from black to a color (10 colors between blue to red), each time in 8 stretch steps.
`Clrstp12`	Twelve times from black to a color (12 colors between blue to red), each time in 8 stretch steps.
`Cyan`	From black to cyan in 16 stretch steps.
`FineGray`	From black to white in 60 stretch steps.
`Gray`	From black to white in 30 stretch steps.
`Green`	From black to green in 16 stretch steps.
`Inverse`	From white to black in 30 stretch steps.
`Magenta`	From black to magenta in 16 stretch steps.
`Pseudo`	From blue to green to red in 6 x 9 stretch steps.
`Red`	From black to red in 16 stretch steps.
`Yellow`	From black to yellow in 16 stretch steps.

There is one system *representation class* available:

ColorCmp	Designed for output raster maps of the Color Composite operation created with the Standard option; 216 colors.

4.2.2 Georeferences

A georeference defines the relation between rows and columns in a raster map and XY-coordinates. The location of pixels in a raster map is thus defined by a georeference. It is advised that raster maps of the same area use the same georeference. A georeference uses a coordinate system, which may contain projection information. Polygon, segment and point maps merely use a coordinate system. A georeference is a service object, usually for several raster maps.

Georeference types
There are five main types of georeferences:

- **georeference corners**: a North-oriented georeference to be used during rasterization of vector data or as the North-oriented georeference to which you want to resample maps;
- **georeference tiepoints**: a non-North-oriented georeference to add coordinates to a satellite image or to a scanned photograph, a scanned map, etc. without using a DTM;

- **georeference direct linear**: to add coordinates to a scanned photograph while using a DTM;
- **georeference orthophoto**: to add coordinates to a scanned aerial photograph while using a DTM and camera parameters;
- **georeference 3D**: to create a three dimensional view of maps.

☞ For a correct behaviour of a georef direct linear, and a georef orthophoto, it is essential that you have marked the 'Interpolation' check box in the Properties dialog box of your DTM raster map. For more information, refer to the Raster Map Properties dialog box.

Furthermore, system georeference None is available for raster maps that do not have coordinates.

Other types of georeferences are obtained when performing an operation on raster maps:

- georeference factor: created by the Aggregate map and the Densify operation;
- georeference mirrorrotate: created by the MirrorRotate operation;
- georeference submap: created by the Sub-map of raster map and the Glue raster maps operation.

General use of georeferences

When a raster map has a georeference, you can:

- inspect coordinates at the position of the mouse pointer in a map window, at the status line;
- retrieve information at the position of the mouse pointer in a map window from other maps, i.e. functionality of the pixel information window;
- inspect the pixel size of raster maps in meters;
- check if raster maps fit on top of each other (requirement for MapCalc and Cross);
- overlay vector data on raster maps;
- rasterize vector data to raster maps;
- resample raster data which uses a particular georeference to another georeference; create an orthophoto from a scanned aerial photograph which has a georef orthophoto by resampling it to a georef corners;
- screen digitize on satellite images or on scanned photographs which have a georef tiepoints; screen digitize on scanned photographs which have a georef direct linear; monoplot on scanned aerial photographs which have a georef orthophoto.

A map window also uses a georeference. When the map window displays a raster map, this is the georeference of the raster map. When a map window only displays vector data, an internal georeference is used that is created from the coordinate system and coordinate boundaries of the vector maps.

Names of georeferences
Any georeference name must start with a character from A to Z. For the other characters in an object name, you are allowed to use: characters A through Z, numbers 0 through 9, and underscores. Object names cannot exceed 8 characters.

☞ To show georeferences in the Catalog, refer to How to customize the Catalog.
☞ To view which georeference is used by a raster map, check the properties of the raster map.

To create a georeference
A georeference can be created:
- during rasterization, click the create georeference button in a Rasterize dialog box; the Create Georeference Corners dialog box will appear; you can create a georef corners.
- when a satellite image, a scanned photograph, or a scanned map is displayed in a map window, choose the Create Georeference command from the File menu of the map window; the Create Georeference (in map window) will appear; you can create a georef tiepoints, a georef direct linear or a georef orthophoto.
- when you want to create a 3D view of your study area, start the Display 3D operation, and click the create georeference button in the dialog box; the Create Georeference 3D dialog box will appear; you can create a georef 3D.

Other general ways to create a georeference are:
- by choosing the Create Georeference command on the File menu of the Main window,
- by double-clicking the New Georef item in the Operation-list,
- by opening the Properties dialog box of a raster map: click the Edit button in the Properties dialog box and click the create button in the subsequent Edit Properties dialog box.

For more information, refer to How to create a georeference, How to add coordinates to an image, How to screen digitize, How to monoplot, How to create an orthophoto or How to create a 3D view.

To view or edit a georeference
You can view and edit a georef corners, a georef tiepoints, a georef directlinear, a georef orthophoto and a georef 3D. The easiest way to edit a georeference is to double-click a georeference in the Catalog. When a raster map, which has a georeference is displayed in a map window, you can also open the Edit menu in the map window and choose Georeference.

Depending on the type of georeference you open (or the georeference which is used by the raster map in the map window), a dialog box will appear or an appropriate editor will be opened:
- for a georef **corners**: the Edit Georeference Corners dialog box will appear;
- for a georef **tiepoints**, a georef **direct linear**, or a georef **orthophoto**: the Tiepoint editor will be opened. When editing a georef orthophoto, the Locate Fiducial Marks dialog box will appear first;

- for a georef **3D**: the Display Options **-** 3D Grid dialog box appears, in which you can specifying how the 3D view should be displayed in the map window. When the 3D view is displayed in a map window, you can select the Georeference command from the Edit menu of the map window to edit the georeference 3D. The Georeference 3D editor is started.

A georef factor, a georef mirrorrotate or a georef submap cannot be edited. Information on these georeferences can be found in their properties dialog box.

Advanced users may wish to open a georef tiepoints, a georef directlinear or a georef orthophoto as a table.

☞ To have a raster map use another georeference than its current one, perform the Resample operation.

Technical information
A georeference consists of an ASCII object definition file (.GRF); in case of a georef tiepoints, a georef direct linear or a georef orthophoto, also a binary data file (.GR#) is available. The object definition file contains further references to:

- the coordinate system that is used by the georeference;
- in case of a georef tiepoints, georef direct linear or a georef orthophoto, the background map on which tiepoints are positioned;
- in case of a georef direct linear or a georef orthophoto, the DTM from which height values should be obtained.

These are properties of a georeference.

By viewing the properties of a georeference, you can see for instance the type of the georeference and find out which raster maps are using this georeference. By clicking the Additional Info button in the properties dialog box of a georef tiepoints, a georef direct linear or a georef orthophoto, you will get extra information on transformation or orientation results.

The system georeference `None` is available for raster maps that do not have coordinates.
Georeference corners:

Stores minimum and maximum XY-coordinates, and whether these refer to the corners of the corner pixels or to the centers of the corner pixels.

$$Col = a_1 X + b_1$$
$$Row = a_2 Y + b_2$$

- Georeference tiepoints:

Stores a set of tiepoints in RowCol and XY-coordinates and a transformation method. Parameters are found by a least squares method. Height values are not taken into account. Available transformation methods and their formulas are:

- Conformal:
  minimum of two tiepoints required;
  $Col = aX + bY + c_1$
  $Row = bX - aY + c_2$
- Affine:
  recommended for satellite images; minimum of three tiepoints required;
  $Col = a_{11}X + a_{12}Y + b_1$
  $Row = a_{12}X + a_{22}Y + b_2$
- Second order bilinear: minimum of 4 tiepoints required;
  $Col = a_1 + b_1X + c_1Y + d_1XY$
  $Row = a_2 + b_2X + c_2Y + d_2XY$
- Full second order: minimum of 6 tiepoints required;
  $Col = a_1 + b_1X + c_1Y + d_1XY + e_1X^2 + f_1Y^2$
  $Row = a_2 + b_2X + c_2Y + d_2XY + e_2X^2 + f_2Y^2$
- Third order: minimum of 10 tiepoints required;
  $Col = a_1 + b_1X + c_1Y + d_1XY + e_1X^2 + f_1Y^2 + g_1X^3 + h_1X^2Y + i_1XY^2 + j_1Y^3$
  $Row = a_2 + b_2X + c_2Y + d_2XY + e_2X^2 + f_2Y^2 + g_2X^3 + h_2X^2Y + i_2XY^2 + j_2Y^3$
- Projective:
  recommended for normal camera, i.e. small format, photographs; conventional rectification; minimum of 4 tiepoints required;
  $Col = (aX + bY + c) / (gX + hY +1)$
  $Row = (dX + eY + f) / (gX + hY + 1)$

- Georeference DirectLinear:
  Recommended for normal camera, i.e. small format, photographs when a DTM is available; also corrects for tilt and relief displacement; Direct Linear Transformation (DLT). Stores a set of tiepoints in RowCol and XYZ-coordinates. Height values can be supplied by the user, otherwise these are obtained from the DTM. Orientation parameters, i.e. camera position (X_0, Y_0, Z_0) and angles (κ, φ, ω), are calculated from the tiepoints. Minimum of 6 tiepoints required; tiepoints must not be co-planar, i.e. the tiepoints should not be on a (tilted) plane.
- Georeference OrthoPhoto:
  Recommended for photogrammetric camera aerial photographs when a DTM is available; also corrects for tilt and relief displacement; Differential rectification. Stores fiducial marks, principal distance, and a set of tiepoints in RowCol and XYZ-coordinates. Height values can be supplied by the user, otherwise these are obtained from the DTM. The user needs to specify fiducial marks and principal distance; orientation parameters, i.e. camera position (X_0, Y_0, Z_0) and angles (κ, φ, ω), are calculated from the tiepoints (outer orientation). Minimum of 3 tiepoints required.
- Georeference 3D:
  Stores a number of 3D view parameters: view point, location height, scale height, horizontal and vertical rotation, distance, view angle.

- Georeference factor:
  Stores a factor of the pixel size of another raster map and horizontal and vertical offset of another raster map.
- Georeference mirrorrotate:
  Stores whether another georeference is mirrored horizontally, vertically, diagonally or is transposed, or rotated 90°, 180°, or 270°.
- Georeference submap:
  Stores offset and size related to another georeference.

4.2.3 Coordinate system

A coordinate system contains information on the kind of coordinates you are using in your maps; you may for instance use user-defined coordinates, coordinates defined by a national standard or coordinates of a certain UTM zone. A coordinate system defines the possible XY- or LatLon-coordinates that can be used in your maps.

Point, segment and polygon maps always have a coordinate system. Raster maps have a georeference, which uses a coordinate system. A coordinate system is a service object for point, segment and polygon maps, and for georeferences of raster maps.

In ILWIS, XY-coordinates are supposed to be in meters and the 90° angle between the positive X-axis and the positive Y-axis is counter-clockwise.

Coordinate system types
There are five main types of coordinate systems:

- **coordinate system boundary only**: to define XY-coordinates for maps by only specifying the boundaries of your study area. This type of coordinate system should only be used when you are sure that you will not use projections at all; furthermore, *maps using a coordinate system boundary only, cannot be transformed into any other coordinate system*.
- **coordinate system projection**: to define XY-coordinates for maps by specifying the boundaries of your study area and when you want to have the possibility to add projection information, ellipsoid information and/or datum information. You can add the projection information later on or right away. Maps with different coordinate systems and different projections can be transformed into one another.
- **coordinate system latlon**: to define LatLon-coordinates for maps by specifying the boundaries of your study area in Latitudes and Longitudes and when you want to have the possibility to add ellipsoid information and/or datum information. You can add the ellipsoid information and/or datum information later on or right away.
- **coordinate system formula**: when you obtained data which is using different XY-coordinates than the coordinate system of your project, and when you know the relation between the two coordinate systems. You can create a coordinate system formula for maps with artificial coordinates, i.e. starting at (0,0) or digitized in millimeters. The coordinate system formula uses a 'related'

coordinate system; this is the coordinate system with correct coordinates. When you have defined the formula and when the map with artificial coordinates uses the newly created coordinate system formula, then you can transform the map to the correct coordinate system.

- **coordinate system tiepoints**: when you obtained data which is using different XY-coordinates than the coordinate system of your project, and when you do not know the relation between the two coordinate systems. You can create a coordinate system tiepoints for maps with artificial coordinates, i.e. starting at (0,0) or digitized in millimeters. The coordinate system tiepoints uses a 'related' coordinate system; this is the coordinate system with correct coordinates. When you have specified the tiepoints and transformation method, and when the map with artificial coordinates uses the newly created coordinate system tiepoints, then you can transform the map to the correct coordinate system.

Furthermore, two coordinate systems are available in the \SYSTEM directory:

- coordinate system `Unknown`: when you do not care about coordinates.
- coordinate system `LatLon`: when you obtained raster or vector data which are supposed to use LatLon-coordinates on a sphere (world-wide).

General use of coordinate systems

- For general analysis purposes, it is advised to use **one** coordinate system for all your maps. This coordinate system should be wide enough to cover all X- and Y-coordinates that should be stored in your maps.
- Coordinate systems enable you to transform vector maps from one coordinate system to the other. There are 3 possibilities:
  - When maps use a coordinate system projection or a coordinate system latlon and when this coordinate system actually contains projection, ellipsoid or datum information, then you can transform the map to any other coordinate system of type projection or latlon which contains projection, ellipsoid or datum information.
  - When maps use a coordinate system formula (which uses a related coordinate system with correct coordinates), then you can transform the map to the related coordinate system.
  - When maps use a coordinate system tiepoints (which uses a related coordinate system with correct coordinates), then you can transform the map to the related coordinate system.
- A map window has the capability to transform vector maps on the fly or temporarily. When a point, segment or polygon map with a certain coordinate system is displayed in a map window, you can drag or add another coordinate system to the map window. If a transformation between both coordinate systems is possible, the map window will use the newly added coordinate system; the vector maps displayed by the map window will be shown in the new coordinate system, i.e. the maps are temporarily transformed.
- When you use pixel information and the map is currently using one coordinate system, you can add another coordinate system to the pixel info window. When a transformation is possible between both coordinate systems, the pixel information window will also show you the coordinates in the new coordinate

system. The pixel information window can also retrieve information from maps with another 'transformable' coordinate system.

- The following maps can be displayed with a graticule:
  - maps which use a coordinate system projection with a projection,
  - maps which use a coordinate system latlon,
  - maps which use a coordinate system formula, where the related coordsys is a coordsys projection with a projection or a coordsys latlon,
  - maps which use a coordinate system tiepoints, where the related coordsys is a coordsys projection with a projection or a coordsys latlon.
- When resampling raster maps from one georeference to another georeference, which uses another coordinate system, then a transformation is automatically performed.

☞ When you receive data in different projections (and you have created correct coordinate systems with correct projections for these maps), it is advised to use the Transform operations to transform vector data from the current coordinate system into one other coordinate system, or to use the Resample operation to resample pixels of a raster map the current georeference into one other georeference (with another coordinate system).

Names of coordinate systems

Any coordinate system name must start with a character from A to Z. For the other characters in an object name, you are allowed to use: characters A through Z, numbers 0 through 9, and underscores. Object names cannot exceed 8 characters.

To create a coordinate system

To create a coordinate system, you can select the Create Coordinate System command on the File menu of the Main window or double-click the NewCrdSys item in the Operation-list. The Create Coordinate System dialog box will appear: you can create any type of coordinate system.

When one or more vector maps which do not have correct coordinates yet are displayed in a map window, you can also select the Create Coordinate System command on the File menu of the map window. The Create Coordinate System (in map window) dialog box will appear: you can create a coordinate system formula or a coordinate system tiepoints.

Other ways to create a coordinate system are:

- by clicking the create button in the dialog box when creating a point, segment or polygon map, or a georeference,
- by opening the Properties dialog box of a vector map: click the Edit button in the Properties dialog box and click the create button in the subsequent Edit Properties dialog box.

For more information, refer to the Create Coordinate System dialog box. It is advised that within one project, all maps use the same coordinate system.

To view or edit a coordinate system

The easiest way to view and/or edit an existing coordinate system, is to double-click a coordinate system in the Catalog. When one or more raster and/or vector maps are displayed in a map window, you can also open the Edit menu in the map window and choose Coordinate System.

Depending on the type of coordinate system you open (or the type of coordinate system that is currently used by the map window), a dialog box will appear or an appropriate editor will be opened:

- for a coordsys **boundary only**: the Edit Coordinate System Boundary Only dialog box will appear;
- for a coordsys **projection**: the Edit Coordinate System Projection dialog box will appear;
- for a coordsys **latlon**: the Edit Coordinate System LatLon dialog box will appear;
- for a coordsys **formula**: the Edit Coordinate System Formula dialog box will appear;
- for a coordsys **tiepoints**: the Coordinate System Tiepoints editor will be opened.

Of course, to open and edit a coordinate system, you can also click a coordinate system with the right mouse button in the Catalog and choose Open from the context-sensitive menu, use the Edit Object command on the Edit menu in the Main window, double-click the Edit item in the Operation-list, etc.

☞ To show coordinate systems in the Catalog, refer to How to customize the Catalog.
☞ To view which coordinate system is used by a georeference of a raster map or by a point, segment or polygon map, check the properties of a georeference or the properties of a point, segment or polygon map.
☞ Advanced users may wish to open a coordinate system tiepoints as a table.

Technical information

A coordinate system consists of an ASCII object definition file (.CSY); in case of a coordsys tiepoints also a binary data file (.CS#) is available. The object definition file stores the coordinate boundaries and, if available, projection information, a formula, etc.

By viewing the properties of a coordinate system, you can see for instance the type of the coordinate system, the boundary coordinates of the coordinate system, and find out by which vector maps and by which georeferences this coordinate system is used. For a coordinate system formula and a coordinate system tiepoints, also the related coordinate system is listed. Furthermore, for a coordinate system tiepoints, you can find the transformation results by clicking the Additional Info button.

Introduction on Projections

A projection defines the relation between the map coordinates (X,Y) and the geographic coordinates latitude and longitude (φ, λ).

The Earth's surface is curved, however in maps it is presented as a flat surface. Therefore, the display of an area on a map will always lead to some deformation or distortion; there is no 'perfect' projection. If you show only a small part of the Earth, like a town, the distortion will be almost insignificant. If, on the other hand, a map shows a continent, deformations and distortions will be a major problem. To correctly represent the curved Earth's surface on a flat map, you need a special projection. The geographic coordinates are converted to a metric coordinate system, measuring the X- and Y-directions in meters. Each projection has unique equations for the transformation from geographic to metric coordinates.

Because of the earth's rotation, the shape of the earth is not a perfect sphere. The earth is flattened towards the poles: the equatorial axis (line from the center to the Equator) is longer than the polar axis. The shape of the earth can be represented by an ellipsoid, or as it is sometimes called, a spheroid (shapes that are generated by revolving an ellipsis around its minor axis). The choice of the ellipsoid which fits best a certain region of the earth surface to be mapped depends on the surface curvature and undulations in that region. Hence every country has its own 'best fit' ellipsoid.

General characteristics of projections

Projection types
Based on the shape of the projection surface, one can classify the projections in azimuthal, conical and cylindrical projections. Therefore, the cone or cylinder needs to be 'unrolled' to form a plane map.

1. Cylindrical projections
   Cylindrical projections may be imagined as the projection to a plane that is wrapped around the globe in the form of a cylinder (see Figure 1). After unrolling, the outline of the world map would be rectangular in shape; the meridians are parallel straight lines which cross at right angles by straight parallel lines of latitude. Examples: Mercator, Plate Carree.

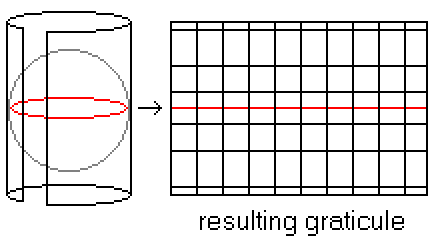


Figure 1: Principle of cylindrical projections.

2. Azimuthal projections
   Azimuthal projections may be imagined as the projection on a plane tangent to the globe (Figure 2). The characteristic outline of the world map would be circular. If the pole is the central point, the meridians are straight lines, spaced at

their true angles intersecting at this center point. Parallels are represented as concentric circles. Examples: Gnomonic, Stereographic.

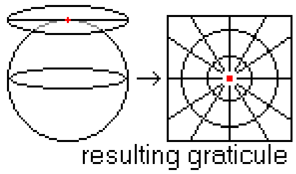


Figure 2: Principle of azimuthal projections.

3. Conical projections
   Conical projections may be imagined as the projection to a plane that is wrapped like a cone around the globe (Figure 3). After unrolling the outline of the world would be fan shaped. The meridians are represented as straight lines and parallels as concentric circles. Only the parallels where the cone touches the globe have the same length as on earth.

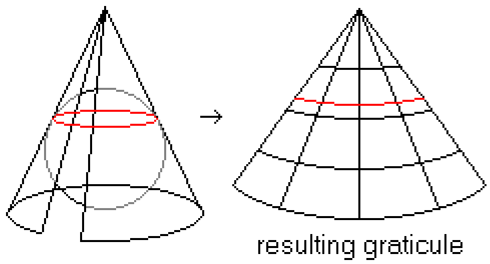


Figure 3: Principle of conical projections.

Aspect

Furthermore, projections can be subdivided according to the direction in which a cylinder, plane or cone is oriented with respect to the globe, the so-called aspect. In the text above, it is assumed that the projection only touches the Earth. However, it is also possible to use a secant cylinder, plane or cone, which intersects the sphere. Figures 4, 5, and 6 show some aspect types for different types of projections.

- For cylindrical projections a normal aspect is a cylinder that touches the equator. A transverse aspect is a cylinder that touches the poles.
- Similarly the normal and secant aspects of azimuthal projections can be visualized.
- The aspect may also be oblique; in that case the cylinder, plane or cone is not horizontally or vertically oriented, but something in between.

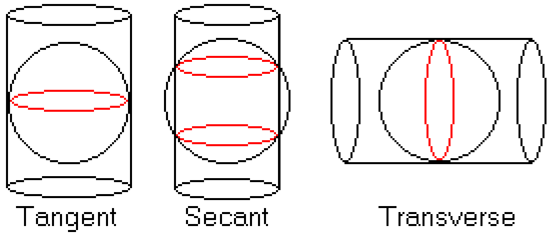


Figure 4: Different aspects for cylindrical projections.

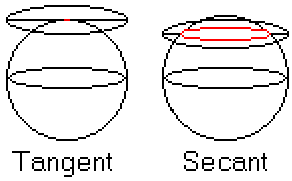


Figure 5: Different aspects for azimuthal projections.

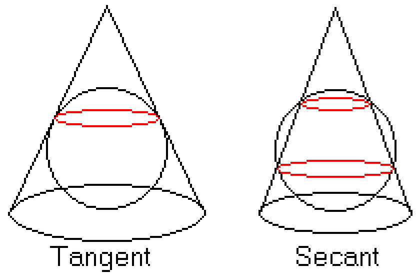


Figure 6: Different aspects for conical projections.

Projection characteristics

As mentioned before a map projection always results in some deformation or distortion. Depending on the type of projection, these distortions will be different. This is indicated by the characteristics of a projection:

- Conformality: a conformal map is one in which all angles are indicated correctly. Angles between two points in a map with a conformal projection and between two points on Earth are the same. As all angles are maintained, the shape of the objects is also preserved. Examples: Mercator, Stereographic.
- Equivalence: a map with an equivalence property is an equal-area map. Although the shape of objects is distorted, the areas of all regions are shown in the same proportion to their true areas. Examples: Mollweide, Sinusoidal.
- Equidistance: an equidistance map has the characteristic that in one direction the distances are preserved.

Available projections

Map projections are named according to the class, the aspect, the property, the name of the originator and the nature of any modification. For an overview of available projections, refer to the Select Projection dialog box. For hints on what projection to use, refer to Suggested projections.

Coordinate system Formula

You can create a coordinate system formula for maps with artificial coordinates, i.e. starting at (0,0) or digitized in millimeters. The coordinate system formula uses a 'related' coordinate system; this is the coordinate system with correct coordinates. When you have defined the formula and when the map with artificial coordinates uses the newly created coordinate system formula, then you can transform the map to the correct coordinate system.

Transformation formulae:

In the formulae below, $\begin{pmatrix} x \\ y \end{pmatrix}$ is used for the related coordinates, $\begin{pmatrix} X_{out} \\ Y_{out} \end{pmatrix}$ for the coordinates of the coordsys formula which you are creating, $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ for the origin in the related coordsys, and $\begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$ for the origin of the coordsys formula which you are creating.

Conformal:

$$\begin{pmatrix} X_{out} \\ Y_{out} \end{pmatrix} = k \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} + \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$$

k = scaling
φ = rotation

Differential scaling:

$$\begin{pmatrix} X_{out} \\ Y_{out} \end{pmatrix} = \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} + \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$$

k_1 = X-scaling
k_2 = Y-scaling

Skew along X-axis:

$$\begin{pmatrix} X_{out} \\ Y_{out} \end{pmatrix} = \begin{pmatrix} 1 & \tan\alpha \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} + \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$$

α = skew

Skew along Y-axis:

$$\begin{pmatrix} X_{out} \\ Y_{out} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \tan\beta & 1 \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} + \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$$

β = skew

Affine:

$$\begin{pmatrix} X_{out} \\ Y_{out} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} + \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$$

a_{11}, a_{12}, a_{21}, a_{22} = matrix coefficients

User-defined expression:

$X_{out} = f_x (x - x_0, y - y_0) + X_0$
$Y_{out} = f_y (x - x_0, y - y_0) + Y_0$

4.3 Annotation

4.3.1 Annotation Text

An Annotation Text object, also called an annotation text layer, is designed to display and store multiple texts at multiple positions. While simple annotation types can only be stored by saving a map window as a map view, an Annotation Text object can be stored itself.

When creating an Annotation text object, you can base the texts that will appear in the text object on an existing point, segment, or polygon map. If you do so, the text object will contain a text (class name, ID or value) for each point, segment or polygon in the selected map.

An Annotation Text object can be edited with the Annotation Text editor or in a table window. In the Annotation Text editor, you can easily insert more text items, change and refine the position of texts (move), make text duplicates, and specify fonts, font sizes, appearance (bold, italics, underline), colors, rotations, etc. for (multiple) selected texts.

To create an annotation text object
In a map window, open the File menu and choose Create Annotation Text. You can also open the Layers menu, choose Add Annotation, Text Layer and then click the Create button in the appearing Add Annotation Text dialog box.

To display annotation text
The easiest way to display an existing annotation text object in a map window is to drag the annotation text object from the Catalog to a map window. In a map window, you can also open the Layers menu and choose Add Annotation, Text Layer and then select an annotation text object in the appearing Add Annotation Text dialog box.

To edit annotation text
Annotation text objects can be edited in the Annotation Text editor or in a table window. In the annotation text editor, you can insert new texts and delete selected existing texts, move one or more selected texts to another position, make text duplicates, and specify fonts, sizes, colors, etc. for (multiple) selected texts.

- When you create a new annotation text object, the annotation text editor is directly opened.
- To edit an existing annotation text object in the annotation text editor, first display the annotation text object in a map window, then open the Edit menu in the map window, choose Edit Layer and select the annotation text layer.
- To edit an annotation text object in a table window, you can simply double-click the annotation text object in the Catalog, or open the File menu in the Main window, choose Open As Table and select the annotation text object.

For more information, refer to the Annotation Text editor.

Technical information
An annotation text object consists of an ASCII object definition file (.ATX) and a binary data file (.AT#). The object definition file contains information on the working scale and information on columns in the table. The object definition file contains no further references to other objects.

4.3.2 Simple Annotation

You can add simple annotation to a map window in the form of texts, legends, boxes, North arrow, scale bar, grid lines, graticule, or bitmaps or metafiles (pictures) from disk. Furthermore, any picture or bitmap that was copied to the clipboard from other Windows application programs can be pasted into ILWIS. You can also paste the contents of another ILWIS map window as annotation into a current map window; for instance as an inset-map giving an overview of a larger area.

These types of annotation can be edited in the Annotation editor. To store this type of annotation, you need to save the map window in which they are displayed as a map view.

☞ You can create an Annotation Text object when you want to store multiple texts at multiple positions. Annotation Text objects are stored as objects and can thus be added to multiple map windows. Texts in an annotation text object can be edited in the Annotation Text editor: you can for instance easily move texts, specify fonts, font size, colors, etc.

Annotation is generally used on top of one or more data layers (displayed maps).

Creating annotation
There are several ways to add annotation to a map window:

- from the Layers menu, choose Add Annotation, and select the type of annotation you want, or
- from the Layers menu, choose Layer Management. In the appearing Layer Management dialog box click the Add Annotation button, and select the type of annotation you want, or
- from the Edit menu, choose Annotation. The Annotation editor is opened directly.

Before annotation is shown, you have to specify some display options for the annotation layer. Then, the Annotation editor is opened. In the Annotation editor, you can use the same possibilities as above, but you can also click one of the add annotation buttons in the button bar (the status line gives an explanation).
For more information, see also How to add annotation to a map window.

Editing annotation
You can edit annotation in the Annotation editor. To open the Annotation editor, choose the Annotation command on the Edit menu of a map window. The Annotation editor is automatically started when a new annotation layer is added to a map window.

In the Annotation editor, texts, legends, boxes, scale bar, North arrow, and bitmaps or pictures can be moved to another position by drag and drop. The size of these types of annotation can be adjusted by dragging the size handles of a selected object. Further, the display options of these types of annotation can be changed by double-clicking. Grid lines and a graticule have a fixed position in a map window according to the georeference of the map window and can thus not be moved or double-clicked. You can always change display options of an annotation layer by choosing the Display Options command from the Layers menu, or through Layer Management. For more information, refer to the Annotation editor or to How to edit annotation.

☞ To create more space in a map window for annotation, select the Extend Window command on the Options menu.

Saving annotation
Annotation should be saved with the contents of your map window as a map view. Either in the Annotation editor or in the map window, open the File menu, and choose the Save View or the Save View As command. All annotation layers and all data layers in the map window are stored with their current display settings. Specify a name for the map view and a title. The title will appear in the title bar when you redisplay the map view.

Annotation types

Annotation Text object
An Annotation Text object, also called an annotation text layer, is designed to display and store *multiple texts at multiple positions*. While other annotation types can only be stored by saving a map window as a map view, an Annotation Text object can be stored by itself.

When you create an annotation text object based on an existing map, you can easily obtain texts for all points, segment or polygons of the selected map. it is advised to create an Annotation Text object. An Annotation Text object can be edited in the Annotation Text editor: you can reposition selected texts, specify fonts, font sizes, appearance (bold, italics, underline), colors, rotations, etc. Texts in an Annotation Text object can only consist of one line.
For more information, see 4.3.1 ILWIS objects: Annotation Text.

Single text
A single text annotation can be used to display *one text at one position* in a map window, for instance a title. Each single text has to be defined in a Display Options - Text dialog box. The text may actually consist of multiple text lines; you can use the Enter key or the Shift+Enter key. Single texts can only be stored by saving a map window as a map view. Single texts can be rotated, and font and colors are user-defined.

Legend
Legends are more or less automatically created from the domain that a map is using and from the representation used by that domain. A legend can be created for maps using a class domain, the image domain or a value domain.

For a legend of a *class map*, color boxes and class names appear in principle for all items in the class domain. The user can hide class names from the legend and manipulate the order in which class names appear in the legend. Further, font and text color are user-defined and the legend can be made transparent or not.

For a legend of an image or a value map, a vertical color bar with gradual colors as used in the displayed map is automatically created. Font and color for the text explaining some of the colors in the color bar are user-defined, and the legend can be made transparent or not. See also Display Options - Legend (Image/Value) dialog box.

North-arrow
A North-arrow points to the North of a map; the North is determined from the georeference of a map window.

In the Display Options - North Arrow dialog box, you can choose between three types of North arrows. Furthermore, you can specify the line width and color. Optionally, text such as N, E, S, W can be displayed with the North arrow. The font and color for the text is of course user-defined.

Scale bar and scale information
A scale bar and scale information (e.g. 1 : 100000) can be shown based on the georeference of a map window.

In the Display Options - Scale Bar dialog box, you can choose whether to create a scale bar as blocks of alternating colors or as a line with tick marks. Furthermore, you can specify the distance intervals and colors used in the scale bar, and the font and colors used for the text below the scale bar.

Optionally, you can also obtain scale information. When you display scale information, the value may seem odd in the map window, however, when you print the map on a certain scale, the scale value is automatically adjusted to the scale that you selected for printing.

Box
Boxes can be used to obtain for instance borders around a legend, a title, etc. In the Display Options - Box dialog box, you can choose between two types of boxes, and you can select a line type and a color of the borders of the box. When you select line type Single, you can also specify the line width. By default, the inside of a box is transparent, but you can also fill a box with a color. You can size a box by dragging one of the size handles of the box in the Annotation editor.

To obtain borders exactly around a mapped area, use the Draw Border check box in the Display Options - Grid Lines dialog box.

Grid lines
Grid lines, coordinate ticks, X- and Y-coordinates and a border fitting a map can be shown. Grid lines are parallel to the X- and Y-axes of the coordinate system.

When a map window displays:
- a North-oriented map (e.g. raster map with a georef corners), the grid lines will be straight and exactly horizontal and vertical;
- a raster map which has a georef tiepoints with a conformal, affine or projective transformation, the grid lines will be straight but not exactly horizontal and vertical;
- a raster map which has a georef tiepoints with a higher order transformation, or a georef direct linear, georef orthophoto, the grid lines will be curved.

In the Display Options - Grid Lines dialog box, you can specify the spacing of the grid lines in meters. Furthermore, you can select a line type and a color for the grid lines.

X- and Y-coordinates of the grid lines can be shown as text inside or outside the map area, and can appear in a short form. The font and color used for the coordinate texts are of course user-defined.

Graticule
A graticule draws parallels and meridians; you require a coordinate system that has projection information.

In the Display Options - Graticule dialog box, you can specify the spacing of the graticule in degrees, minutes and seconds. Furthermore, you can specify a line type and a color for the graticule. Latitudes and Longitudes of the graticule cannot be shown yet as text.

Bitmaps from disk
A bitmap is a standard Windows graphic file format in 'raster' form. Bitmaps can be added as annotation to an ILWIS map window. Bitmaps are stored in device independent bitmap (DIB) format, i.e. bitmaps are not dependent on the graphic board. Bitmaps can use a 2, 16, 256 color palette, or are stored using 24 bits per pixel. The default file extension for bitmaps is .BMP.

When using 256 color bitmaps, and when a lot of windows with many colors are open already, you may loose the colors of pasted bitmaps. When this occurs, you might consider using 16 color bitmaps.

Windows metafile or pictures from disk
A Windows metafile, often called a picture, is a standard Windows graphic file format in 'vector' form. Metafiles or pictures can be added as annotation to an

ILWIS map window. A metafile consists of a set of Windows drawing instructions. The default extension for Windows metafiles is .WMF.

ILWIS maps as inset-maps in other map windows
Prepare an ILWIS map as usual, copy it into the clipboard and paste as a picture into another ILWIS map window. The pasted layer is an annotation layer.

Pasting pictures and bitmaps from clipboard
Any picture or bitmap in the clipboard (copied into the clipboard from any Windows application), can be pasted into an ILWIS map window. You can use the clipboard viewer in Windows to view what is in the clipboard.

Chapter 5

Editors

Note: This chapter is written as an enhancement to Chapter 5: Editors in the ILWIS 2.1 Reference Guide. It does *not* completely replace the mentioned chapter: instead you are invited to use both as one.

5.1 Representation Class editor

Functionality

With the representation class editor, you can change the representation as used by a class domain, a group domain, or a picture domain. In the representation class editor, you see all the class names of the domain to which this representation is linked and the way in which these classes will be represented on the screen and on a printer.

For each class as available in the domain, you can edit:

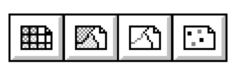
- colors (for mapping units in raster maps);
- colors, hatching or patterns (for mapping units in polygon maps);
- colors, line types, line widths, etc. or use equally spaced symbols (for segments);
- colors, symbol types, symbol sizes, etc. (for points).

The Representation Class editor consists of an editor window with the class names and their colors, a menu bar, a button bar, and Red, Green and Blue color slide bars at the bottom of the window.

To start the Representation Class editor, see How to create a representation and How to edit a representation.

☞ You can always create a number of representations for a certain domain, for instance by clicking the create button next to the Representation list box in the Display Options dialog box of a map. Then, to display a map with a certain representation, in the map's Display Options dialog box, select the representation you want to use.

General working of the Representation Class editor

1. Select a map type:
   - Click one of the buttons in the button bar:  for Raster, Polygon, Segment, or Point respectively, or
   - Open the Options menu and select either Raster, Polygon, Segment or Point.

Keep in mind that when changing a color of a class in one 'mode' (e.g. Polygon), then the color of this class will also change for all other modes (e.g. Raster, Segment).

2. Select a single class in the editor window and subsequently edit the color and optionally other representation attributes of this class:
   - Double-click a class or its color box in the editor window, or
   - Select a class or its color box and choose Edit Item from the Edit menu, or
   - Select a class or its color box and press the Enter key on the keyboard.

   The Edit Repr Item (raster), Edit Repr Item (polygon), Edit Repr Item (segment) or the Edit Repr Item (point) dialog box will appear.

   Depending on whether you selected to edit the raster form, polygon form, segment form or point form of the representation, the Edit Repr. Item dialog box will allow you to edit:
   - for raster: colors
   - for polygon: fill colors, hatchings or patterns;
   - for segment: colors, line types (e.g. single, double, triple, dot, dash dot, blocked), line widths, optional support lines, or a symbol font and a symbol which will be used to draw lines;
   - for point: simple symbols or symbols from any installed symbol font, symbol sizes, colors, etc.

   Furthermore, you can use the Red, Green and Blue color slide bars at the bottom of the editor window to edit the colors of a single selected class.

3. Select multiple classes in the editor window and subsequently assign a single color to all selected classes or assign a smooth range of colors to the selected classes.
   - Select multiple classes in the editor window by using the Shift and/or Ctrl key, and
   - Open the Edit menu and choose Edit Multiple Items, or press the Enter key on the keyboard. The Edit Multiple Items dialog box will appear.

   Assigning a range of colors can be used:
   - to obtain smooth color changes between multiple classes, for instance from yellow to red, through orange, or
   - to obtain similar colors for related classes in a domain, e.g. various forest types in similar shades of green, etc.

   When assigning a color range, the order of the classes in the domain will be used. For more information, see the Edit Multiple Items dialog box.

To interactively change colors of maps

1. For any raster, polygon, segment or point map with a class or group domain:
   - Display the map in a map window and make sure that the Info check box in the map's Display Options dialog box is marked;

- Open the Layers menu in the map window and choose the Double-Click Action command; in the appearing dialog box, select the Edit Representation option;
- Then, double-click the pixels, polygons, segments or points in the map: each time a Edit Repr Item dialog box will appear in which you can directly change colors, polygon patterns, segment line types, line widths, point symbols, etc.
- The map window will directly show the map in the changed representation.
- Continue double-clicking units in the map and editing the representation of that class until satisfied.

2. For any raster, polygon, segment or point map with a class, group or value domain:
   - Display the map in a map window;
   - Open the Edit menu in the map window and choose the Representation command;
   - Depending on the type of representation that the domain of this map is using, the representation class editor as described above, or the representation value/gradual editor is opened;
   - In the appearing representation editor window, edit one or more colors, etc;
   - To see the changes in the map window, press the Redraw  button in the map window;
   - Continue editing colors in the representation editor and redrawing the map window until satisfied.

   In this way, you can work with a representation editor and a map window simultaneously.

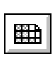
☞ Advanced users may wish to edit a representation class in table form; click a representation class with the right mouse button in the Catalog, and choose Open As Table from the context-sensitive menu.

Button bar

The buttons in the button bar of the Representation Class editor allow you to select a map type. This shows the representation as if this representation was used by that map type.

Every class name has a color. Each pixel, polygon, segment, point with that class name will use that color. Polygons, segments, and points have some other representation characteristcs as well.

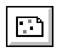
The following actions can be performed:

 Raster Map: Edit the colors of this representation as used to display a raster map.

 Polygon Map: Edit the colors, hatchings or patterns of this representation as used to display a polygon map.

 Segment Map: Edit the colors, line types, line widths, etc. of this

representation as used to display a segment map.


Point Map: Edit the colors, symbols, symbol sizes, etc. of this representation as used to display a point map.

The button bar can be hidden or shown with the Button Bar command on the Options menu.

5.2 Georeference Tiepoints editor

Functionality

General information

With the Tiepoint editor you can edit:

- a georeference **tiepoints**: to add coordinates to a satellite image or to a scanned photograph without using a DTM, or
- a georeference **direct linear**: to add coordinates to a scanned photograph while using a DTM, or
- a georeference **orthophoto**: to add coordinates to a scanned aerial photograph while using a DTM and camera parameters.

For more information on georeferences, see ILWIS objects : georeferences.

The Tiepoint editor shows:

- in a map window: a *raster map*, for instance a satellite image, a scanned photograph or a scanned map. This is the background map as specified during the creation of the georeference. By inserting tiepoints, also called ground control points, you can establish relationships between the pixels of that map (row,col) and real world XY- or XYZ-coordinates.
- *existing tiepoints in a table in a secondary window*: for each existing tiepoint, its Row and Column values in the raster map, its real world X- and Y-coordinates, and the residuals. When editing a georef direct linear or a georef orthophoto, also Z-values (heights) are shown: column Z_dtm displays height values as obtained from the Digital Terrain Model (DTM); in case you know more exact height values for control points, you can enter these yourself in column Z.

The tiepoint editor has a menu bar, a button bar, a status line and a context-sensitive menu.

Purpose of a georeference tiepoints, direct linear, orthophoto
A georeference **tiepoints** is mostly used to add coordinates to a *satellite image* or to another map, which is not North-oriented and/or in which the pixels do not represent exactly square areas on the ground. You can also use a georef tiepoints to add coordinates to a scanned photograph when a DTM of the area is not available. When working with multi-spectral images, you should add a created georef tiepoints to all bands of the image by editing the properties of the bands. This type of georeference can be used to add coordinates to satellite imagery and for subsequent screen digitizing or to resample the image to another georeference (e.g. to a georef corners) for overlay operations.

A georeference **direct linear** is mostly used to add coordinates to a *scanned photograph* which was taken with a *normal camera*, and when you have a DTM to also correct for tilt and relief displacement. A georef direct linear performs a Direct Linear Transformation (DLT). This type of georeference can for instance be used to add coordinates to small format aerial photographs and for subsequent screen digitizing or to resample the photograph to another georeference (e.g. to a georef corners) for overlay operations.

A georeference **orthophoto** is mostly used to add coordinates to a *scanned aerial photograph with fiducial marks*, taken with a photogrammetric camera with known principal distance, and when you have a DTM to also correct for tilt and relief displacement. A georef orthophoto performs a Differential rectification. This type of georeference can be used to add coordinates to professional near vertical aerial photographs and further monoplotting on the photograph or for creating an orthophoto (resampling).

When you added tiepoints to an image or a scanned photograph, you can:
- Display any type of vector data on top of the map;
- Create or update vector data using the georeferenced image or photo as background (screen digitizing);
- Use the map in pixel info;
- When you have raster maps of different sources or images of different dates and you want to perform raster operations to combine these maps or images (e.g. MapCalc, Cross), first create a georeference tiepoints for each set of maps/images, then use the Resample operation and resample the maps/images preferably to a georeference corners;
- When you want to combine rasterized vector maps with satellite data, you can rasterize the vector data on the georeference tiepoints of the satellite images. In case you prefer North-oriented raster maps, rasterize the vector maps with a georeference corners, and Resample the images with the georeference tiepoints to this georeference corners.

Using the Tiepoint editor

Inserting tiepoints
In the tiepoint editor, tiepoints can be added to a map in several manners:
- first, click at a recognizable point in the map without coordinates,
- the Add Tiepoint dialog box appears. In this dialog box, the row and column values at the position of the click are filled out.
  When you already have some tiepoints, the dialog box will also come up with a suggestion for the XY-coordinates. This suggestion is the result of the calculation with the existing tiepoints and using the current transformation method. This suggestion is a measure of the quality of the current tiepoints. The suggestion is merely a suggestion, it is advised to enter your own XY-coordinates to prevent false accuracy.

Then:

- click at the same position in a map which already has correct XY-coordinates and which is displayed in another map window (Master/Slave), or
- digitize the same point in an analog paper map on a digitizer, or
- read the correct XY-coordinates for this point from a table or an analog paper map, and type these XY-coordinates in the dialog box.

When editing a georef direct linear or a georef orthophoto, you can optionally specify a Z-value for the inserted tiepoint; otherwise the height value as found in the DTM is used.

The inserted tiepoint appears in the map window and in the tiepoint table.

For a more detailed description on inserting tiepoints (for instance Master/Slave), refer to How to use the Georeference Tiepoints editor.

(De)selecting tiepoints

Column `Active` in the tiepoint table indicates whether or not a tiepoint is used in the transformation. Include tiepoints by the putting True in column `Active`; exclude tiepoints by putting False in column `Active`. You can type T or F in column `Active` or use the space bar to switch between these.

Changing colors and symbol size of tiepoints and/or fiducial marks

To change the colors or symbol size of tiepoints or to change the colors of fiducial marks, choose Customize from the Edit menu.

Deleting tiepoints

To delete a tiepoint, choose Delete Tiepoint from the Edit menu, or click the Delete Tiepoint button in the button bar. Furthermore, after selecting a tiepoint in the tiepoint table, you can press Del on the keyboard, or use the right mouse button and select Delete from the context-sensitive menu.

Georef Tiepoints transformation method

When editing a georef tiepoints, one of the following transformation methods can be selected: conformal, affine, second order bilinear, full second order, third order and projective.

- For satellite images an *affine transformation* will usually do;
- For a scanned photograph (without DTM), a *projective transformation* is recommended.

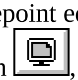
Tiepoint requirements

- Mathematical minimum number of tiepoints required:
  - Georef tiepoints: conformal 2; affine 3; second order 4; full second order 6; third order 10; projective 4.
  - Georef direct linear: 6
  - Georef orthophoto: 3
- You should always insert more tiepoints than is mathematically required.
- Tiepoints should be well spread over the map (XY-direction).

- For a georef direct linear, the tiepoints should also be well spread in Z-direction and they should not be co-planar, i.e. in Z-direction, the tiepoints should not be on a (tilted) plane.

Inspecting DRow, DCol and Sigma
Columns DRow and DCol show the difference between calculated Row and Col values and actual Row and Col values in pixels. Very good control points have DRow and DCol values less than 2. Sigma is calculated from these values and the degrees of freedom, and gives a measure for the overall accountability or credibility of the active tiepoints.

☞ When you have already a vector map of the area of the image, it can be handy to display this vector map on top of the image while working with the tiepoint editor.

When you now add tiepoints to the image and press the Redraw button , you can judge the quality of the tiepoints.

☞ For a georef tiepoints which should be used by *all bands of a satellite image*: after you have finished adding tiepoints to a background map in the Tiepoint editor, you have to add the created georef tiepoints to all bands of the image. This can be done by opening the Properties dialog boxes of the bands, click the Edit Properties button and select the correct georef tiepoints.

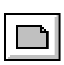
☞ When finished creating a georeference for an image or a photograph, you can directly create a segment map (choose File, Create in the map window) and start screen digitizing on the image or photo which now uses the created georef tiepoints, georef direct linear or georef orthophoto. When you created a georef orthophoto, the screen digitizing is called monoplotting.

☞ When finished editing a georef tiepoints, a georef direct linear or a georef orthophoto, detailed information on transformation or orientation results can be viewed in the Properties dialog box of the georeference: click the Additional Info button.

Button bar

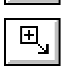
The button bar of the georeference tiepoints editor largely resembles the button bar of a map window but there are some extra buttons to add a tiepoint, to delete a tiepoint, to choose a transformation method and to exit the editor.

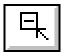
The following actions can be performed:


Entire Map: Displays the entire map in the map window.


Zoom In: Zooms in on a selected spot or area in the map. The mouse pointer changes into a magnifying-glass. Click to zoom in on a spot of interest, or drag the pointer from one corner of interest to another (the mouse pointer changes into a little box).


Zoom Out: Zooms out directly from the center of the displayed map with a factor 2.

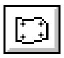

Larger: Enlarges the size of the window with a factor 2. If this would not fit on your screen, this button is empty.

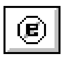

Smaller: Reduces the size of the window with a factor 2.


Redraw: Redraws the contents of the map window.


Add tiepoint: Opens the Add Tiepoint dialog box in which you can enter Row and Column numbers and XY-coordinates for a new tiepoint from the keyboard. You can also add tiepoints by clicking in the map window, or by typing the Row and Column numbers and XY-coordinates directly in the tiepoint table.


Delete tiepoint: Opens the Delete Tiepoint dialog box in which you can specify the the number of the tiepoint you want to delete. You can also delete one or more tiepoints by selecting records in the tiepoint table, then press the Del key on the keyboard or choose Delete from the Edit menu.


Transformation method: Opens the Transformation dialog box in which you can select a transformation method with which the relation between Row and Column values and the XY-coordinates of your tiepoints should be calculated. This button is only available when editing a georef tiepoints.


Exit editor: Leaves the editor and keeps all changes.

The button bar can be hidden or shown with the Button Bar command on the Options menu. If the action of a certain button cannot be performed at a certain moment, the button appears gray.

5.3 Coordinate system Tiepoints editor

Functionality

General information

With the Coordinate system Tiepoints editor you can edit a coordinate system **tiepoints**: to add coordinates to a vector map. For more information on coordinate systems, see ILWIS objects : coordinate systems.

The Coordinate system Tiepoints editor shows:

- in a map window: *one or more vector maps*. This is the background map as specified during the creation of the coordinate system. By inserting tiepoints, you can establish relationships between the current XY-coordinates in the map and related XY-coordinates.
- *existing tiepoints in a table in a secondary window*: for each existing tiepoint, its current XY-coordinate (X and Y) in the vector maps, its related XY-coordinates (RelX and RelY), and the residuals (DX and DY).

The coordinate system tiepoints editor has a menu bar, a button bar, a status line and a context-sensitive menu.

Purpose of a coordinate system tiepoints
A coordinate system **tiepoints** is mostly used to add correct XY-coordinates to a vector map which has wrong or artificial coordinates, e.g. maps starting at (0,0), maps digitized in millimeters, or maps that were were imported from other packages and do not have correct coordinates.

When working with multiple imported vector maps of the same kind, you should add a created coordsys tiepoints to all similar vector maps, by editing the properties of these maps.

When you added a coordsys tiepoints to a vector map, you can:
- Display the vector map on top of any map with correct coordinates;
- Use the map and the coordinate system in Pixel Info;
- To actually change the XY-coordinates of your vector maps, first create a coordinate system tiepoints, then transform the vector maps to the correct coordinate system.

Using the Coordinate system Tiepoint editor

Inserting tiepoints
In the tiepoint editor, tiepoints can be added to a map in several manners:
- first, click at a recognizable point in the map with wrong coordinates,
- the Add Tiepoint dialog box appears. In this dialog box, the current XY-coordinates at the position of the click are filled out.
  When you already have some tiepoints, the dialog box will also come up with a suggestion for the related XY-coordinates. This suggestion is the result of the calculation with the existing tiepoints and using the current transformation method. This suggestion is a measure of the quality of the current tiepoints. The suggestion is merely a suggestion, it is advised to enter your own related XY-coordinates to prevent false accuracy.

Then:
- click at the same position in a map which already has correct XY-coordinates and which is displayed in another map window (Master/Slave), or
- digitize the same point in an analog paper map on a digitizer, or
- read the correct XY-coordinates for this point from a table or an analog paper map, and type these XY-coordinates in the dialog box.

The inserted tiepoint appears in the map window and in the tiepoint table.

(De)selecting tiepoints
Column `Active` in the tiepoint table indicates whether or not a tiepoint is used in the transformation. Include tiepoints by the putting True in column Active; exclude tiepoints by putting False in column `Active`. You can type `T` or `F` in column `Active` or use the space bar to switch between these.
Changing colors and symbol size of tiepoints:

To change the colors or symbol size of tiepoints or to change the colors of fiducial marks, choose Customize from the Edit menu.

Deleting tiepoints
To delete a tiepoint, choose Delete Tiepoint from the Edit menu, or click the Delete Tiepoint button in the button bar. Furthermore, after selecting a tiepoint in the tiepoint table, you can press Del on the keyboard, or use the right mouse button and select Delete from the context-sensitive menu.

Coordsys Tiepoints transformation method
When editing a coordinate system tiepoints, one of the following transformation methods can be selected: conformal, affine, second order bilinear, full second order, third order and projective. An affine transformation will usually do.

Changing the boundaries of the coordinate system
When creating a coordinate system tiepoints in a map window, then for the boundaries of the new coordsys tiepoints, the boundaries of the map window are automatically used. Usually, you will not have to change these values.

Tiepoint requirements
- Mathematical minimum number of tiepoints required: conformal 2; affine 3; second order 4; full second order 6; third order 10; projective 4.
- You should always insert more tiepoints than is mathematically required.
- Tiepoints should be well spread over the map (XY-direction).

Inspecting DX, DY and Sigma
Columns DX and DY show the difference between calculated related XY-coordinates and actual XY-coordinates in meters. Sigma is calculated from these values and the degrees of freedom, and gives a measure for the overall accountability or credibility of the active tiepoints.

Menu commands

File	Exit Editor
Edit	Copy Paste Delete Add Tiepoint ... Delete Tiepoint ... Transformation ... Boundaries ... Customize ...
Layers	*see map window*
Options	*see map window*

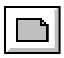
Context sensitive menu

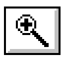
Select Area
Add Tiepoint
Delete Tiepoint
Transformation
Customize
Exit Editor

Button bar

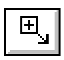
The button bar of the coordinate system tiepoints editor largely resembles the button bar of a map window but there are some extra buttons to add a tiepoint, to delete a tiepoint, to choose a transformation method and to exit the editor.

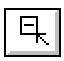
The following actions can be performed:

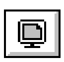

Entire Map: Displays the entire map in the map window.


Zoom In: Zooms in on a selected spot or area in the map. The mouse pointer changes into a magnifying-glass. Click to zoom in on a spot of interest, or drag the pointer from one corner of interest to another (the mouse pointer changes into a little box).


Zoom Out: Zooms out directly from the center of the displayed map with a factor 2.

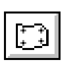

Larger: Enlarges the size of the window with a factor 2. If this would not fit on your screen, this button is empty.

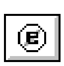

Smaller: Reduces the size of the window with a factor 2.


Redraw: Redraws the contents of the map window.


Add tiepoint: Opens the Add Tiepoint dialog box in which you can enter Row and Column numbers and XY-coordinates for a new tiepoint from the keyboard. You can also add tiepoints by clicking in the map window, or by typing the Row and Column numbers and XY-coordinates directly in the tiepoint table.


Delete tiepoint: Opens the Delete Tiepoint dialog box in which you can specify the the number of the tiepoint you want to delete. You can also delete one or more tiepoints by selecting records in the tiepoint table, then press the Del key on the keyboard or choose Delete from the Edit menu.


Transformation method: Opens the Transformation dialog box in which you can select a transformation method with which the relation between Row and Column values and the XY-coordinates of your tiepoints should be calculated. This button is only available when editing a georef tiepoints.


Exit editor: Leaves the editor and keeps all changes.

The button bar can be hidden or shown with the Button Bar command on the Options menu. If the action of a certain button cannot be performed at a certain moment, the button appears gray.

5.4 Annotation Text editor

Functionality

The Annotation Text editor allows you to edit text items stored in an Annotation Text object.

With the Annotation Text editor, you can:
- insert new texts and delete selected existing texts,
- move one or more selected texts to another position,
- make text duplicates, and
- specify fonts, sizes, colors, etc. for (multiple) selected texts.

Additionally, you can cut, copy and paste text items to and from the clipboard, and change the working scale.

The Annotation Text editor has a menu bar, a context-sensitive menu, and a button bar. All changes that you make in the annotation text object are continuously stored.

You can also edit an annotation text object in table form. This may provide quicker, easier or more precise possibilities to edit texts, positions and other settings like font, font sizes, colors, etc. To open an annotation text object as a table, you can double-click the object in the Catalog, or open the File menu in the Main window, choose Open As Table and select an annotation text object.

☞ While annotation text is edited in the Annotation Text editor, other annotation types like single texts, legends, boxes, North arrows, scale bars, grid lines, graticule, bitmaps or metafiles are edited with the Annotation editor.

To start the Annotation Text editor
The Annotation Text editor is automatically started when you create a new annotation text object. For more information, see the Create Annotation Text dialog box.

To edit an annotation text object, which is displayed in a map window, open the Edit menu, choose Edit Layer and select the annotation text layer. For more information, see ILWIS objects : annotation text objects.

To insert more annotation text items
In the Create Annotation Text dialog box, you will usually base a new annotation text object on an existing polygon, segment or point map. You will then automatically obtain texts (class names, IDs, or values) of all polygons, segments or points in that map.

To add more or other text items

- double-click at a new position in the editor window;
- locate the mouse pointer at a position where you want to insert a new text and press the Ins key on the keyboard;
- open the Edit menu and choose the Add Text command;
- press the right mouse button and select Add Text from the context-sensitive menu.

The Add Text / Edit Text dialog box will appear. When you use the Edit menu or the context-sensitive menu, you may have to position the text later on.

To change the text string, font, font size, colors, etc, of text items

To change text and settings of a single text:

- double-click a single text item;
- select a single text item, press the right mouse button and choose Edit from the context-sensitive menu;
- select a single text item, and press the Enter key.

The Add Text / Edit Text dialog box will appear in which you can edit the existing text, the font in which this text item appears, the font size, the color, etc.

To change settings of multiple selected texts:

- select multiple text items, press the right mouse button and choose Edit from the context-sensitive menu;
- select multiple text items, and press the Enter key.

The Edit Texts dialog box will appear in which you can edit the the font in which all these text items appear, the font size, the color, etc.

To change only one setting for one or more selected text items: choose the Change command from the Edit menu or from the context-sensitive menu. You can either change the font, the font size, whether the selected text(s) appear in bold, italics or underlined (toggles), the color, the justification, whether rectangles around the text should be transparent (toggle) and the rotation. For font, font size, color, justification and rotation, a little dialog box will appear; the other settings are toggles without dialog box.

To change only one setting for one or more selected text items, you can also use the following short cut keys on the keyboard:

`Shift + F`	to select another font;
`Shift + S`	to change the font size;
`Shift + B`	toggle between bold / not bold;
`Shift + I`	toggle between italics / not italics;
`Shift + U`	toggle between underlined / not underlined;
`Shift + C`	to select another color;
`Shift + J`	to select another justification method;
`Shift + T`	toggle between transparent / not transparent;
`Shift + R`	to select another rotation.

To select multiple text items

- open the Edit menu and choose Select All: all text items will be selected, or

- drag a box around the text items which you want to select, or
- press and hold the Shift key down and click the text items which you want to select.

To position annotation text items
- click on a single text item and drag it to another position;
- double-click a single text item and fill out XY-coordinate or RowCol numbers in the Add Text / Edit Text dialog box, or
- select multiple text items and drag them to another position.

To cut, copy and paste selected text items
Select one or more text items, and:
- press Ctrl+X (cut) to remove the selected items from the editor window and copy them into the clipboard;
- press Ctrl+C (copy) to copy the selected items to the clipboard;
- press Ctrl+V (paste) to paste the contents (first line) of the clipboard into the editor window, at the current position of the mouse pointer.

You can also use the Cut, Copy and Paste commands on the Edit menu or on the context-sensitive menu.

To make duplicates of annotation text items
- Select one or more text items,
- press and hold the Ctrl key down, then
- press and hold the left mouse button down, and
- move the mouse pointer to another position.

To delete annotation text items
Select one or more text items, and:
- press the Del key on the keyboard;
- select Delete from the Edit menu or from the context-sensitive menu.

To change the working scale
You can change the working scale, which determines the relation between font sizes on your screen and on the printer. The working scale is usually the scale on which you want to print your maps later on. To change the working scale: open the File menu and choose Options. The Options dialog box will appear in which you can specify another value for your working scale.

Subsequently, a question will be asked: `Update all font sizes?`
- answer `Yes` if you want to keep your texts in the same proportions;
- answer `No` if you want to keep the font sizes as specified before; then, if you Redraw the editor window, the texts may appear larger or smaller depending on whether you specified a larger or smaller value for the working scale.

To exit the Annotation Text editor
- double-click the Control-menu box, or
- in the button bar, click the Exit Editor button , or

- from the File menu, choose Exit Editor.

All changes that you make in the annotation text object are continuously stored.

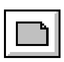
Menu commands

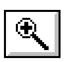
File	Options	
	Exit Editor	
Edit	Cut	
	Copy	
	Paste	
	Delete	
	Select All	
	Add Text ...	
	Edit ...	
	Change	font
		font size
		bold
		italic
		underline
		color
		justification
		transparency
		rotation.
Layers	*see pixel editor*	
Options	*see pixel editor*	

Button bar

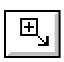
The button bar of the annotation text editor largely resembles the button bar of a map window; there is an extra button to exit the editor.

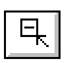
The following actions can be performed:

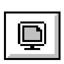

Entire Map: Displays the entire map in the map window.

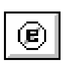

Zoom In: Zooms in on a selected spot or area in the map. The mouse pointer changes into a magnifying-glass. Click to zoom in on a spot of interest, or drag the pointer from one corner of interest to another (the mouse pointer changes into a little box).


Zoom Out: Zooms out directly from the center of the displayed map with a factor 2.


Larger: Enlarges the size of the window with a factor 2. If this would not fit on your screen, this button is empty.


Smaller: Reduces the size of the window with a factor 2.


Redraw: Redraws the contents of the annotation text editor.


Exit editor: Leaves the annotation text editor and keeps all changes.

The button bar can be hidden or shown with the Button Bar command on Options menu. If the action of a certain button cannot be performed at a certain moment, the button appears gray.

Chapter 6

Map & Table Calculation

Note: This chapter is written as an enhancement to Chapter 6: Map & Table calculation in the ILWIS 2.1 Reference Guide. It does *not* completely replace the mentioned chapter: instead you are invited to use both as one.

6.1 Map calculation

6.1.1 Logical operators

Truth table of logical AND
The result of A and B is true only if both expressions A and B are true. If either expression A or expression B is true, false is returned. If A is true and B is undefined, or vice versa, then undefined is returned.

The truth table of the logical AND is:

AND	T	F	?
T	T	F	?
F	F	F	F
?	?	F	?

Truth table of logical OR
The result of A or B is true if one or both of the expressions a and b is true.

The truth table of the logical OR is:

OR	T	F	?
T	T	T	T
F	T	F	?
?	T	?	?

Truth table of logical XOR
The result of A xor B is true if only one of the expressions A or B is true. When both expressions are true or when both expressions are false, false is returned. When either expression A or B is undefined, undefined is returned.

The truth table of the logical XOR is:

XOR	T	F	?
T	F	T	?
F	T	F	?
?	?	?	?

Truth table of logical NOT
The result of not A is true if expression A is false. If expression A is true, false is returned. If expression A is undefined, undefined is returned.

The truth table of the logical NOT is:

NOT	
T	F
F	T
?	?

6.2 Table calculation

6.2.1 Aggregating values

Several 'functions' are available to aggregate values of a value column. Aggregation means that you get one aggregate value, for instance the average or the sum, of a whole column, or one value per group of class names. In this way, you can for instance calculate the total area of each class.

The following aggregation 'functions' are available:

- average,
- count,
- minimum,
- median,
- maximum,
- predominant,
- standard deviation, and
- sum.

These functions can be used:

- to aggregate values of a *whole column* (using any aggregation function).
  You will obtain one output value; for all records, the same aggregation answer appears.
- to aggregate values of a column per group, i.e. aggregate values by the classes or ID of another column (using any aggregation function): use a 'group by' column. For all records that have the same class or ID in the selected 'group by' column, the same aggregation answer will appear. The 'group by' column is usually a column with a class, or ID domain.

- to aggregate values while taking into account weights: use a weight column. In this way, you can calculate weighted averages, etc. The weight column is a column with a value domain.
  - When you do not select a 'group by' column, you will obtain one output value.
  - When you do select a 'group by' column, you will obtain answers per group.

Aggregation results can be written either:

- into the same table, in a new column, or
- in a new table, in a new column (only possible through the menu), or
- in another existing table, in a new column.

Aggregations can be performed by choosing the Aggregation from the Columns menu in a table window, or by typing an expression on the command line of a table window.

Aggregation through a dialog box
Generally, you will aggregate columns via the menu in the table window. To aggregate the values of a column:

- In a table window which contains the column which values you want to aggregate, open the Columns menu and choose the Aggregation command. The Aggregate Column dialog box appears.
- Fill out the Aggregate Column dialog box:
  - Select the value column that contains the data you want to aggregate.
  - Select the aggregation function that you want to use: Average, Count, Maximum, Median, Minimum, Predominant, Standard Deviation or Sum.
  - If you want to aggregate values by classes or IDs in a Group By column, select the Group By check box, and select the 'group by' column; else the values of the complete column will be aggregated.
  - If you want to use weight values during the aggregation, select the Weight check box and select the weight column; else all records are treated equally.
  - If you want the aggregation results written into another table; select the Output Table check box and type a table name; else the results will be written into a column in the current table.
  - Type a name for the output column that will contain the results of the aggregation.

For examples of aggregations, click the links in the text on aggregations from the command line as given below.

Aggregation through the command line
To aggregate the values of a column you may also type a statement on the command line of a table window.

The syntax for aggregation on the command line is:
`ColumnAggregateAvg(`*`col`*`)`
calculates the average value of values in column *col*

`ColumnAggregateAvg(`*`col, g`*`)`
: calculates the average value of values in column *col* per group *g*

`ColumnAggregateAvg(`*`col, g, w`*`)`
: calculates the average value of values in column *col* per group *g* using weights *w*

`ColumnAggregateAvg(`*`col , , w`*`)`
: calculates the average value of values in column *col* using weights *w*

`ColumnAggregateCnt(`*`col, g`*`)`
: counts the number of times that column *col* is not undefined, optionally per group *g*

`ColumnAggregateMax(`*`col, g`*`)`
: determines the maximum value of column *col*, optionally per group *g*

`ColumnAggregateMed(`*`col, g, w`*`)`
: calculates the median value of column *col*, optionally per group *g*, and optionally using weights *w*

`ColumnAggregateMin(`*`col, g`*`)`
: determines the minimum value of column *col*, optionally per group *g*

`ColumnAggregatePrd(`*`col, g, w`*`)`
: determines the predominant value of column *col*, optionally per group *g*, and optionally using weights *w*

`ColumnAggregateStd(`*`col, g, w`*`)`
: calculates the standard deviation of column *col*, optionally per group *g*, and optionally using weights *w*

`ColumnAggregateSum(`*`col, g`*`)`
: calculates the sum of column *col*, optionally per group *g*

Notes

- The use of a group column *g* and/or a weight column *w* is optional:
  - Aggregations with parameters (*col*) return one aggregation result for all entries of column *col*.
  - Aggregations with parameters (*col*, *g*) return one aggregation result for all entries with the same class name or ID in group column *g*.
  - Aggregations with parameters (*col*, *g*, *w*) return one weighted aggregation result for all entries with the same class name or ID in group column *g*.
  - Aggregations with parameters (*col* , , *w*) return one weighted aggregation result for all entries of column *col*.
- Parameter *col* refers to nothing else than a column name. This means that within the brackets no other expressions can be used.
- Parameter *g* is a column with a class or ID domain; parameter *w* is a column with a value domain.
- For ILWIS 1.41 users: argument *g* can be considered as the key column.

Aliases

Instead of the long '`ColumnAggregate`*`AggFunc`*', you can also use the following aliases (the links provide examples):

`AGGAVG(col, g, w)`	calculates the average value of *col*, optionally per group *g* ,and optionally using weights *w*
`AGGCNT(col, g)`	counts the number of times that column *col* is not undefined, optionally per group *g*
`AGGMAX(col, g)`	determines the maximum value of *col*, optionally per group *g*
`AGGMED(col, g, w)`	calculates the median value of *col*, optionally per group *g*, and optionally using weights *w*
`AGGMIN(col, g)`	determines the minimum value of *col*, optionally per group *g*
`AGGPRD(col, g, w)`	determines the predominant value of *col*, optionally per group *g*, and optionally using weights *w*
`AGGSTD(col, g, w)`	calculates the standard deviation of *col*, optionally per group *g*, and optionally using weights *w*
`AGGSUM(col, g)`	calculates the sum of *col*, optionally per group *g*

Example 1

Two simple aggregations are:

```
Avg1 = AGGAVG(Area)
Avg2 = AGGAVG(Area,Landuse)
```

Parcel	Landuse	Area	Avg1	Avg2
00123	Residential	4000	10000	5000
00124	Residential	3500	10000	5000
00125	Commercial	17500	10000	17500
00126	Residential	7500	10000	5000
00127	Industrial	20000	10000	20000
01272	Institutional	12500	10000	12500
04625	Residential	5000	10000	5000

Column `Avg1` contains the average of the area of all parcels.

Column `Avg2` contains the averages of parcel areas per land use class: Residential, Commercial, Industrial and Institutional. For class Residential:
(4000+3500+7500+5000) / 4 = 5000

Example 2 (advanced)

When you are currently in one table (Province) and you want to retrieve aggregated values from another table (Municip), you may use:

```
Population = AGGSUM(Municip.pop, municip.prov)
```

Province.tbt

Dom. Province	any info
Prov1	1000
Prov2	2000
Prov3	5000
Prov4	4000
Prov5	2000
....	

Municip.tbt

Dom. Municip	Population	Province
Municip1	9920	Prov1
Municip2	4131	Prov1
Municip3	2161	Prov2
Municip4	4918	Prov1
Municip5	10461	Prov3
....		

Province.tbt after aggregation

Dom. Province	any info	Population
Prov1	1000	18969
Prov2	2000	2161
Prov3	5000	10461
Prov4	4000	?
Prov5	2000	?
...		

This expression will only work when the GroupBy column (Municip.prov) has the same domain as the current table (Province). It is usually easier to perform a join operation through the Join dialog box. This will give the same result as the expression above, and is even more flexible. For more information, see Table calculation : Join columns.

Example 3 (advanced)

To write output values from a current table (Parcel) into in a new column of another existing table (Landuse), use the following syntax:

```
Landuse.AvgParcelSize = AGGAVG(Parcel.Area, Parcel.Landuse)
```

Parcel.tbt

Dom. Parcel	Landuse	Area
00123	Residential	4000
00124	Residential	3500
00125	Commercial	17500
00126	Residential	7500
00127	Industrial	20000
01272	Institutional	12500
04625	Residential	5000
...		

Landuse.tbt

Dom. Landuse	any info
Residential	1000
Commercial	2000
Industrial	5000
Institutional	4000
...	

Landuse.tbt after aggregation

Dom. Landuse	any info	AvgParcelSize
Residential	1000	5000
Commercial	2000	17500
Industrial	5000	20000
Institutional	4000	12500
...		

The average of `Area` in table `Parcel` grouped by Landuse, is written into column `AvgParcelSize` in table `Landuse`.

When you want to retrieve for instance area values from a histogram, you have to specify the extension of the histogram as *tablename.ext.column*.

6.2.2 Joining columns of other tables

The join operation enables you to read a column from a second table and join it into the current table. To do so, you need a link between the tables. This link is made via the domain of the tables or via the domain of columns in the tables. When a column is used to make the link, this column is called a *key column*. When the domain of a table is used to make the link, you do not have to specify it.

Joining two tables use the same domain
Do not specify key columns. The tables have a one to one relation.

Joining column in the current table uses the same domain as the second table
Only specify a key column from the current table, i.e. key1 or the *Key column*. The relation between the column of the current table and the second table is a many to one relation.

Joining current table uses the same domain as a column in the second table
Only specify a key column from the second table, i.e. key2:

- when the classes or IDs in the key column of the second table are not unique (i.e. one to many relation), then the values that you want to join need to be aggregated and you have to specify key2 as the *Group By column* by which the values to be joined will be grouped during aggregation. Optionally, you can select a column from the second table as a weight column.
- when the classes or IDs in the key column of the second table are unique (i.e. one to one relation): specify key2 as the *Via Key column.*

Joining column in the current table and second table use the same domain
Specify a key column from the current table, i.e. key1, as well as a key column from the second table, i.e. key2:

- when the classes or IDs in the key column of the second table are not unique (i.e. many to many relation), then the values that you want to join need to be aggregated and you have to specify key2 as the *Group By column* by which the values to be joined will be grouped during aggregation. Optionally, you can select a column from the second table as a weight column.
- when the classes or IDs in the key column of the second table are unique (i.e. many to one relation): specify key2 as the *Via Key column*.

Notes

- When you need to specify a key column from the first table, i.e. key1; this is called the Key column.
- When you need to specify a key column from the second table, i.e. key2, you can either do this by specifying a Group By column (you will aggregate the values to be joined) or by specifying a Via Key column.
- Links between tables are always through class or ID domains.
- For more information on column aggregations, refer to Table calculation : aggregating values.

Joining through a dialog box
Generally, you will join columns via the menu in the table window. To join a column from another table into the current table:

- In a table window which displays the table into which you want to join one or more columns of another table, open the Columns menu and choose the Join command. The Join Column dialog box appears.
- Fill out the Join Column dialog box:
  - Select the table from which you want to join a column into the current table.

- Select a column from the second table, i.e. the column, which you want to join into the current table.
- If necessary, select the Key check box and fill out a column name of the current table (key1) which will be used to make a link to the second table; else deselect the Key check box.
- If necessary, select the Aggregation check box and select a column from the second table for the Group By column (key2). Furthermore, select an aggregation function and, optionally, select a weight column from the second table which contains the weight values to be used during the aggregation. Otherwise, fill out a column name from the second table as the Via Key (key2 and no aggregation).
- Type the name of the output column that will contain the joined values.

Joining through the command line

To join a column from another table into the current table you may also type a statement on the command line of a table window.

The syntax of the join operation on the command line is:

(1) OutColName = `ColumnJoin`(TableName, ColumnName)
(2) OutColName = `ColumnJoin`(TableName, ColumnName, Key1)
(3) OutColName = `ColumnJoin2ndKey`(TableName, ColumnName, ViaKey)
(4) OutColName = `ColumnJoin2ndKey`(TableName, ColumnName, Key1, ViaKey)
(5) OutColName = `ColumnJoin`*`AggFunc`*(TableName, ColumnName, GroupBy)
(6) OutColName = `ColumnJoin`*`AggFunc`*(TableName, ColumnName, GroupBy, Weight)
(7) OutColName = `ColumnJoin`*`AggFunc`*(TableName, ColumnName, GroupBy, Weight, Key1)

where:

OutColName	is the output column name. Usually, this is the same as the column name that was chosen to be joined into the current table.
`ColumnJoin`	is the command to start the Join operation.
`ColumnJoin2ndKey`	is the command to start the Join operation using a column from the second table to make a link to the current table.
TableName	is the name of the second table from which you want to join a column into the current table.
ColumnName	is a column name from the second table, i.e. the column that you want to join into the current table.
Key1	is a column from the current table; the domain of this column is either the same as the domain of the second table, or as the domain of a column in the second table.
ViaKey	is a column from the second table in which the classes or IDs are unique; the domain of this column is either also the domain of the current table or the same as the domain of a column in the current table. No aggregation will be performed.

`ColumnJoin`*`AggFunc`*
: is the command to start the Join operation with aggregation, i.e. joining while using an aggregation function. Type directly after `ColumnJoin` one of the following aggregation functions: `Avg` | `Cnt` | `Max` | `Med` | `Min` | `Prd` | `Std` | `Sum`. The *AggFunc* part should thus be replaced by one of the aggregation functions.

GroupBy
: is a column from the second table in which the classes or IDs are not unique; the values to be joined will be aggregated according to this GroupBy column.

Weight
: is an optional parameter to specify a column from the second table which contains weight values for the aggregation.

Formula 1 represents the case where both tables have the same domain.
Formula 2 represents the case where a *column* of the current table has the same domain as the second table (i.e. many to one relation).
Formula 3 represents the case where the current table uses the same domain as a *column* in the second table and when the classes or IDs in that column in the second table are unique (one to one relation).
Formula 4 represents the case where a *column* in the current table uses the same domain as a *column* in the second table and when the classes or IDs in that column in the second table are unique (many to one relation).
Formula 5 represents the case where the current table uses the same domain as a column in the second table and when the classes or IDs in that *column* in the second table are not unique (one to many relation). The values will be aggregated during the join.
Formula 6 represents the case where the current table uses the same domain as a *column* in the second table and when the classes or IDs in that column in the second table are not unique (one to many relation). The values will be aggregated during the join while using a weight column.
Formula 7 represents the case where a *column* in the current table uses the same domain as a *column* in the second table and when the classes or IDs in that column in the second table are not unique (many to many relation). The values will be aggregated during the join while using a weight column.

Example 1
The domain of the current table is the same as the domain of another table.

Table Landuse contains landuse classes and also lists the commercial value of these landuse classes.

The (raster or polygon) histogram of the landuse map contains the area of each landuse class.
The column Area from the histogram will be joined into attribute table Landuse.

Landuse.tbt

Dom Landuse	Commval
Residential	1000
Commercial	2000
Industrial	5000
Institutional	4000
Agricultural	2000

Landuse.his/.hsa

Dom Landuse	Area
Residential	9920800
Commercial	4131200
Industrial	2161600
Institutional	4918400
Agricultural	10461600

Landuse.tbt after joining

Dom Landuse	Commval	Area
Residential	1000	9920800
Commercial	2000	4131200
Industrial	5000	2161600
Institutional	4000	4918400
Agricultural	2000	10461600

Open table Landuse as the current table. From the Columns menu in the table, choose Join.
In the Join Columns dialog box:

- Select histogram `Landuse` as the second table,
- Select column `Area` as the column you want to join into the current table,
- Type a name for the output column to contain the joined values, e.g. Area.

Or open table Landuse as the current table, and type the following expression on the command line of the table window:

```
Area = ColumnJoin(Landuse.his.Area)
Area = ColumnJoin(Landuse.hsa.Area)
```

With the first formula, the column `Area` from the raster histogram `Landuse.his` is joined into the attribute table.
With the second formula, the `Area` column from polygon histogram `Landuse.hsa` is joined into the attribute table.

Example 2
The domain of a column in the current table is the same as the domain of the second table.
Table Municip contains a number of municipalities, the population of each municipality and a column indicating whether the municipalities are considered large, medium or small (column MunClass).
Table MuniSubs contains information for large, medium and small municipalities. It contains a column Subsidy, which represents for instance expected subsidy figures for types of municipalities.
The subsidy figures in table MuniSubs will be joined into the Municipality table.

Municip.tbt

Dom. Municip	Population	MunClass
Municip1	99208	MunLarge
Municip2	41312	MunMedium
Municip3	21616	MunSmall
Municip4	49184	MunMedium
Municip5	104616	MunLarge
...		

MuniSubs.tbt

Dom.MunClass	Subsidy
MunSmall	1000
MunMedium	2000
MunLarge	5000
...	

Municip.tbt after joining

Dom. Municip	Population	MunClass	Subsidy
Municip1	99208	MunLarge	5000
Municip2	41312	MunMedium	2000
Municip3	21616	MunSmall	1000
Municip4	49184	MunMedium	2000
Municip5	104616	MunLarge	5000
...			

Open table Municip as the current table. From the Columns menu in the table, choose Join.

In the Join Columns dialog box:

- Select table `MuniSubs` as the second table,
- Select column `Subsidy` as the column to join into the current table,
- Select the Key check box, select column `MunClass`.
- Type a name for the output column to contain the joined values, e.g. Subsidy.

Or open table Municip as the current table, and type the following expression on the command line of the table window:

```
Subsidy = ColumnJoin(RateMuni, Subsidy, MunClass)
```

Example 3

The domain of your current table is the same as the domain of a column in the second table.
Table Province lists for each province some information. The domain of this table is Province; the Province domain contains all provinces of a certain country.
Table Municip contains population figures for each municipality. Furthermore, for each municipality it is known in which province it is. The column Population in the table has a value domain. The column Province has domain Province.
Column Population in table Municip will be joined into the Province table. As a province usually contains more than one municipality, the municipal population figures need to be aggregated during the join.

Province.tbt

Dom. Province	any info
Prov1	1000
Prov2	2000
Prov3	5000
Prov4	4000
Prov5	2000
....	

Municip.tbt

Dom. Municip	Population	Province
Municip1	9920	Prov1
Municip2	4131	Prov1
Municip3	2161	Prov2
Municip4	4918	Prov1
Municip5	10461	Prov3
....		

Province.tbt after joining

Dom. Province	any info	Population
Prov1	1000	18969
Prov2	2000	2161
Prov3	5000	10461
Prov4	4000	?
Prov5	2000	?
....		

Open table Province as the current table. From the Columns menu in the table, choose Join.
In the Join Columns dialog box:

- Select table `Municip` as the second table,
- Select column `Population` as the column to join into the current table,
- The Aggregation check box is already selected:
  - select for the Aggregation function: Sum,
  - select for the GroupBy column: column Province,
- Type a name for the output column to contain the joined values, eg Population.

Or open table Province as the current table, and type the following expression on the command line of the table window:

```
Population = ColumnJoinSum(Municip, Pop, Prov)
```

Example 4

The domain of a column in the current table is the same as the domain of a column in the second table.

From the Columns menu in the table, choose Join. Choose the table name, which you want to use and the column to join into your current table. Then, select a key column in your current table and a column in the second table to group your data. Note that these two columns must have the same domain. You may choose to perform an aggregation on the values. The join will take place as described before.

6.2.3 Creating and running scripts (example)

By creating and applying a script, you can perform a series of ILWIS operations. This is comparable to using batch files in ILWIS version 1.4.

With a script, MapCalc and TabCalc expressions can be performed, and any ILWIS operation. Further, some extra commands are possible to show objects, or to perform some file management. For more information on script syntax, see Appendices : operators and functions in MapCalc and TabCalc, Appendices : ILWIS expressions and Appendices : ILWIS script language (syntax).

To create a script

- In the Main window, open the File menu, and choose Create Script, or
- In the Operation-list, double-click the item New Script.

The Create Script dialog box appears in which you can type your script expressions.

Example

To calculate a slope maps in percentages and in degrees:

1. Create a script (e.g. 'Slopes')
2. In the dialog box where the script is defined: type, to insert a comment line:
   ```
   // script to calc slope maps in percentages and degrees
   ```
3. To use operation InterpolContour to create an interpolated height map from segment contour lines; type:
   ```
   %2 = MapInterpolContour(%1,geo)
   ```
   Perform a contour interpolation on segment map `%1`, use existing georeference 'geo', and write the output to map `%2`.
   Advanced users may wish to define a georeference corners with a script command; see Appendices : ILWIS script language (syntax).
4. To use filter dfdx on the interpolated contour map to calculate height differences in X-direction; type:
   ```
   %3 = MapFilter(%2, dfdx)
   ```
   Filter map `%2` with the `dfdx` filter and write the output to map `%3`.
5. To use filter dfdy on the interpolated contour map to calculate height differences in Y-direction; type:
   ```
   %4 = MapFilter(%2, dfdy)
   ```
   Filter map `%2` with the `dfdx` filter and write the output to map `%4`.

6. To calculate a slope map from these, type:
```
%5 = 100 * HYP(%3,%4) / PIXSIZE(%2)
```
   HYP is an internal Mapcalc/Tabcalc function;
   %3 and %4 are the output maps from the filtering;
   function PIXSIZE returns the pixel size of raster map %2;
   %5 is the output map name of the map containing slope value in percentages.
7. To convert the percentage values into degrees, type:
```
%6 = RADDEG(ATAN(%5/100))
```
   Function ATAN and RADDEG are internal MapCalc/TabCalc functions.
8. After running the script (see step 9), the output maps will be available as dependent maps. The expression by which a map is created is stored in the map's object definition files. The data file for an output map will be calculated when you double-click an output map in the Catalog.
   To have the script calculate the data files for the output maps, you may add the following lines to your script:
```
calc %2.mpr
calc %5.mpr
calc %6.mpr
```
   In fact, by adding only `calc %6.mpr`, all maps, which are part of the process to calculate map %6 will be calculated as well.
9. To run the script, type on the command line in the Main window:
```
run Slopes Contour.mps DEM.mpr DX DY SlopePct SlopeDeg
```
   In script Slopes, %1 is filled out as Contour.mps, %2 as DEM.mpr, %3 as DX, %4 as DY, %5 as SlopePct, and %6 as SlopeDeg.

Of course, you can also use objects names inside the script instead of parameters %1 (Contour, segment map), %2 (DEM, Digital Elevation Model), etc. For more information on running scripts, see also How to run scripts.

The result of running this script are maps SLOPEPCT and SLOPEDEG which are slope maps in percentages and in degrees.

Mind: the following slope values are the same: 30°= 58%, 45°= 100%, 60°= 173%, 80°= 567%. As you see, slope values in the SLOPEPCT map can be greater than 100%.

Additionally, you can prepare representations for both maps with the Representation Value/Gradual editor.
You can also create two domain Groups to classify both output maps, e.g.:
classes 0-10%, 10-25%, 25-50%, 50-100%, >100% for the slope map in percentages and
classes 0-6°, 6-12.5°, 12.5-22.5°, 22.5-45°, >45° for the slope map in degrees.

Use these domain groups in the Slicing operation.

Chapter 7

Operations

Note: This chapter is written as an enhancement to Chapter 7: Operations in the ILWIS 2.1 Reference Guide. It does *not* completely replace the mentioned chapter: instead you are invited to use both as one.

7.1 Visualization

7.1.1 Show map list as Color Composite

Select a map list in the Show Map List as Color Composite dialog box, and display three maps present in this map list together as an interactive color composite.

One map will be displayed in shades of red, one in shades of green and one in shades of blue. Putting three images together in one color composite can give a better visual impression of the reality on the ground, than by displaying one band at a time. Examples of color composites are false color (or IR) images and 'natural color' images.

By using an interactive color composite, you can easily change intervals, select other bands, etc. The resulting color composite is displayed in a map window, which can be saved as a map view. Interactive color composites are very suitable to be used as a background during sampling or during screen digitizing.

Your graphics board needs to be configured to use more than 256 colors, for instance High Color 16-bit, or True Color 24-bit (see Display Settings in Windows' Control Panel).

☞ When you want to create a permanent color composite, which can for instance be used as a raster drape over a 3D model, choose Color Composite from the Operations, Image Processing menu.

7.2 Raster Operations

7.2.1 Cross

Functionality

The Cross operation performs an overlay of two raster maps. Pixels on the same positions in both maps are compared; the occurring combinations of class names, identifiers or values of pixels in the first input map and those of pixels in the second input map are stored. These combinations give an output cross map and a cross table. The cross table includes the combinations of input values, classes or IDs, the number of pixels that occur for each combination and the area for each combination.

Input map requirements

- Both maps should have the same georeference.
- No restrictions on domain types.

Domain and georeference of output map and table

Cross creates an Identifier domain for the output map and table. This output domain obtains the same name as the output map and is filled with the combinations of class names, IDs or values of both input maps. When an input map has a class or ID domain in which the class names or IDs have codes, then these codes will appear in the output domain.
The output map uses the same georeference as the input maps.

Example

Input map 1:

B	B	C	D
A	A	C	D
A	A	C	C
B	C	D	A

Input map 2:

S	S	S	S
R	R	S	S
T	R	S	T
T	R	T	T

Output cross map:

BS	BS	CS	DS
AR	AR	CS	DS
AT	AR	CS	CT
BT	CR	DT	AT

In the picture of the output map above, read B×S instead of BS, etc.

Output cross table

Domain	Map1	Map2	NPix	Area
A * R	A	R	3	…
A * T	A	T	2	…
B * S	B	S	2	…
B * T	B	T	1	…
C * R	C	R	1	…
C * S	C	S	3	…
C * T	C	T	1	…
D * S	D	S	2	…
D * T	D	T	1	…

The cross table lists

Domain	the combination of class names, identifiers, values or group names of the first map with the second map's class names, IDs or values is returned as the output domain for the cross table. If class names or IDs in the input map have codes, then these codes will appear in the output domain.
Map1	the class name, identifier, or value of pixels in the first input map.
Map2	the class name, identifier, or value of pixels in the second input map.
NrPix	the number of pixels that occur as a combination.
Area	the areas of combinations as: NrPix * pixel size * pixel size.

Combinations with undefined values

When you use the Cross operation through the Cross dialog box, combinations with undefined values will by default not appear in the output cross table, i.e. undefined values are ignored. If an output cross table should also list combinations with undefined values, clear one or both of the *Ignore Undefs* check boxes in the dialog box. The combination Undefined value in the first input map and Undefined value in the second input map will never be listed.

Dialog box

Dialog box options

First map:	Select the first input raster map. Open the drop-down list box by clicking it and select a map, or directly drag a map from the Catalog into this box.
Ignore undefs:	Select this check box to ignore undefined values in the first input map: in the output cross table, combinations with undefined values in the first map will not be listed. Clear this check box when combinations with undefined values in the first map should be listed in the cross table.
Second map:	Select the second input raster map.
Ignore undefs:	Select this check box to ignore undefined values in the second input map; in the output cross table, combinations with undefined values in the second map will not be listed. Clear this check box when combinations with undefined values in the second map should be listed in the cross table.
Output table:	Type a name for the output cross table which will contain the combinations of the two input maps. This name will also be used for the output domain.
Show:	Select this check box if you want the output cross table to be directly displayed. Clear this check box if you do not want to see this table immediately: you simply define how the output table (and map) should be created.
Description:	Optionally, type a description for the output cross table (and map).
Output map:	Select this check box if you want to obtain a cross map containing the combinations of the two input maps. Subsequently, type a name for the output raster map. Clear this check box if an output cross map is not desired.

A dependent table is created; optionally, a dependent map can be created as well. Furthermore, an ID domain is created (same name as cross table) which contains the combinations of domain items of both input maps.

Command line

Cross can be directly executed by typing one of the following expressions on the command line of the Main window:

OUTTABLE=	`TableCross`(FirstInputMap, SecondInputMap)
OUTTABLE=	`TableCross`(FirstInputMap, SecondInputMap, OUTMAP)
OUTTABLE=	`TableCross`(FirstInputMap, SecondInputMap [, OUTMAP] , `IgnoreUndefs`)
OUTTABLE=	`TableCross`(FirstInputMap, SecondInputMap [, OUTMAP] , `IgnoreUndef1`)
OUTTABLE=	`TableCross`(FirstInputMap, SecondInputMap [, OUTMAP] , `IgnoreUndef2`)

where:

OUTTABLE	is the name of your output cross table.
OUTMAP	is an optional parameter to create an output cross map.
`TableCross`	is the command to start cross and produce an output cross table (optionally with an output cross map).
FirstInputMap	is the name of your first input raster map (any domain).
SecondInputMap	is the name of your second input raster map (any domain).
`IgnoreUndefs`	an optional parameter to ignore undefined values in both maps; in the output cross table combinations with undefined values will not appear. When the parameter is not used, the output cross table will also show combinations with undefined values.
`IgnoreUndef1`	an optional parameter to ignore undefined values of the first input map.
`IgnoreUndef2`	an optional parameter to ignore undefined values of the second input map.

Alias:

OUTMAP=	`MapCross`(FirstInputMapName, SecondInputMapName, OUTTABLE)

OUTMAP	is the name of your output cross map.
`MapCross`	is the command to start cross and produce an output cross map and an output cross table.

For other parameters, see above.

When the definition symbol `=` is used, a dependent output table and/or a dependent map are created; when the assignment symbol `:=` is used, dependency links are immediately broken after the output table and/or map have been calculated.

7.2.2 Glue raster maps

Functionality

The Glue raster maps operation glues or merges two or more georeferenced input raster maps into one output raster map. The output map then comprises the total area of all input maps. The domains of the input maps are merged when needed. With the Glue raster maps operation, you can thus merge two or more adjacent or partly overlapping raster maps (i.e. make a mosaic) or glue smaller raster maps onto a larger one.

When the input raster maps have attribute tables, also the tables will be automatically merged; for more information see the Glue tables operation.

Input maps: In the dialog box, you can select 2, 3, or 4 input raster maps. On the command line, you can specify as many input maps as you like. The input maps may be purely adjacent to one another, partly overlapping, or totally overlapping. When the input maps are (partly) overlapping, the input maps form a pile of maps on top of each other.

Map on top: When the pile of input maps are (partly) overlapping the same area, you have to decide which map should be considered as the map on top. When for a pixel a value is found in the map on top, that value will appear in the output map. When the undefined value is found in the map on top, the operation will look in the map 'below' it. Undefined pixels thus act as being transparent and provide 'openings' to enable the operation to find a value in the map below the current one.

In the dialog box, you can select the check box 'Last Map on Top' to order the input maps as:
- the map selected last on top,
- the map selected one but last below that one,
- until the map selected first.

On the command line, you can use the `Replace` option to this end.

Then, for each pixel, the operation will do:
- when a value is encountered in the map you selected last: that value will appear in the output map;
- when the undefined value is encountered in the map you selected last: the operation will look in the map that was selected one but last;
  - when a value is encountered in the map you selected one but last: that value will appear in the output map;
  - when the undefined value is encountered in the map you selected one but last: the operation will in the map that was selected second but last, etc.

Only when no value is found at all, the output pixel will be assigned the undefined value. Note: when the output map uses the Image domain, this means value 0; when the output map uses a `Picture` domain or the `Color` domain, this means color (0,0,0), i.e. black.

First input map
The georeference of your first input map will be used to construct a new georeference for the output map. The output georeference will always be sized in such a way that all input georeferences will fit in it. However, the georeference of your first input map will directly determine the pixel size and the coordinate system for the output georeference. The output georeference will obtain the same name as the output raster map; the output georeference is a georeference of type submap.

Domain combinations
You can always merge maps that use the same domain, and maps that use domains of the same type. In some cases, you can also merge maps of different domain types. The list below shows the possible combinations of input domain types and also shows the output domain type.

In	In	Out
Image	Image	Image
Image	Value	Value
Image	Color	Color
Value	Value	Value
Value	Color	Color
Class	Image	Color
Class	Value	Color
Class	Class	Class
Class	Color	Color
ID	ID	ID
Picture	Picture	Picture
Picture	Color	Color
Color	Color	Color
Bool	Bool	Bool
Bit	Bool	Bool
Bit	Bit	Bit

For more information, see also the section Domain of output map below.

Georeferences
When all input raster maps use the same georeference, the output raster map will also use that georeference. When the input georeferences are different, a new georeference (type submap) will be automatically created. The output georeference is based on the georeference of your first input map but the georef size will be extended so that all input georeferences fit in the output georeference. The georeference of your first input determines furthermore the pixel size and coordinate system of the new georeference. If necessary, all input maps will be automatically resampled to the output georeference; the resampling is done according to the Nearest Neighbour method. See also the section Georeference of output map below.

Example

To show a certain `landuse` type (of a class map) on a satellite image:

- Select an image that you want to use as background and stretch the image with the Stretch operation.
- To obtain for instance a `landuse` map showing only agricultural units, perform MapCalc statements like:
  ```
  Agri= iff ((landuse="agriculture") or
  (landuse="agriculture (irrigated)"), landuse, "?")
  ```
  Explanation: if map landuse is classified as agriculture, then have these classes remain, else assign undefined.
- Use the Glue Raster Maps operation to merge the map Agri with your stretched image. The results may look like below.
- The result of the Glue Raster Map operation is shown in Figure 1 below.

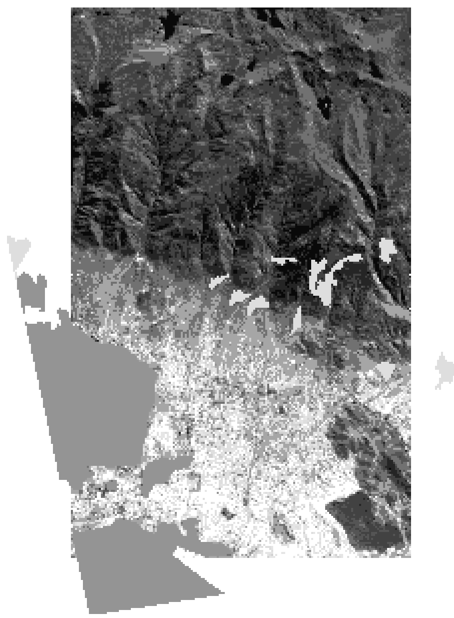


Figure 1: Agricultural land use over a satellite image. The dark gray part is agriculture with irrigation; the light gray part is rainfed agriculture.

For presentation and printing purposes, you could of course also use the Mask Polygons operation, use as mask `agri*` . Then, display the output polygon map on top of the satellite image in a map window.

Input map requirements

The domains of the input maps should be mergeable, see the list of domain combinations above.

All input raster maps must have a georeference, which is not georeference `None`.

Domain of output map
The domain type for the output map can be found in the list of domain combinations above. Here some extra information will be given on how the system deals with the domains for the output maps.

- The output map will use *system domain Image* when you are combining images with each other.
- The output map will use *system domain Value* when you are combining value maps with value maps or images with value maps. The smallest minimum and the largest maximum of all input maps determine the value range for the output map; the precision is the smallest precision of the input maps.
- The output map will use a *class domain* when all input maps use a class domain. When all input maps use the same class domain, then the output map will also use this class domain. When the input maps use different class domains and when the class names in the input domains are different, then all class names of all input domains will be merged automatically into a new output domain. You can choose whether the new domain should be stored as a separate object, or whether it should be stored by the output map (internal domain). Also the representations of the input class domains will be merged into a new output representation; the new representation is either stored as a separate object or by the output map (internal representation).
- The output map will use an *ID domain* when all input maps use an ID domain. The same procedure is followed as for class maps. There are no representations involved.
- The output map will use a *Picture domain* when all input maps use a Picture domain. When all input maps use the same Picture domain, the output map will also use this Picture domain. When the input maps use different Picture domains and when the colors in the input Picture domains are different, then the colors of all input domains will be merged automatically into a new output Picture domain. You have to keep in mind though that a Picture domain can only contain a maximum of 256 colors. Thus, when the total number of all input colors is 256 or less, the colors of the input pictures will be retained in the output picture. The new Picture domain will always be stored by the output map (internal domain) and not as a separate object.
- The output map will use *system domain Color* when you are combining images, value maps, class maps or pictures with a map that uses the Color domain, and when you are combining class maps with images or class maps with value maps. For class maps and pictures, the output colors will be retrieved from the representation of the input domains; for images and value maps, the output color will be retrieved from system representation Gray is used.

Georeference of output map
When all input maps use the same georeference, that georeference will also be used for the output map. When the input maps use different georeferences, then a new georeference (type submap) will be automatically created for the output map. The new georeference will always use the pixel size and coordinate system of your first input map; the size of new the georeference will be such that all input maps fit in it. The new georeference will obtain the same name as the output map. If necessary, all

input maps are resampled with the Nearest Neighbour method to this output georeference.

Usually, the input georeferences will use the same coordinate system, which covers the whole area already. In case input georeferences use different but compatible coordinate systems (e.g. different projections), the coordinate system of the first input map is used for the output georef, and a coordinate transformation is performed for the other input maps.

☞ When input maps are resampled to a new output georeference, this will be done with the Nearest Neighbour method. When working with value maps with different georeferences, and when you would like to calculate with the output value map of the Glue Raster maps operation, you will obtain better interpolation results from the Glue Raster maps operation, when:

- you first make one new large georeference,
- resample your maps to this new georeference by using the Resample operation with the Bilinear or BiCubic interpolation method, and
- only when all maps use the same new large georeference, use the Glue Raster Maps operation.

☞ When you want to combine class maps with ID maps, you first have to convert either the class domain into an ID domain or the ID domain into a class domain. In fact, it is quite easy to convert Class and ID domain(s) to each other: open the Domain Properties dialog box of the domain that you want to convert, and click the Convert to Classes or the Convert to IDs button. When converting IDs to classes, you can create a representation class for your map; when converting from classes to IDs, you will loose your representation class.

☞ When you want to combine a class map with one image, it is advised to first stretch the image; this will improve the contrast in the output map with the `Color` domain.

☞ When you want to combine a class map with multiple images, it is advised to first use the glue raster maps operation only with all images, then stretch the output image and finally use the glue raster maps operation again to merge the class map in it. This will improve the contrast in the output map with the `Color` domain.

Notes

- When merging class maps or ID maps, by default an internal domain is created for the output map to reduce the number of separately stored domains. Internal domains are stored internally in the output map. If you like, you can select the New Domain check box in the dialog box and specify a new name for the output domain if you want the output domain to be stored as a separate object.
- The maximum number of columns for any output map is 64000 for a 1-byte output map, 32000 for a 2-byte output map, 16000 for 4-byte output map and 8000 for an 8-byte output map.

For more information on internal domains and representations, refer to How to open internal domains/representations.

Dialog box

Dialog box options

Number of input maps:	Select the number of raster maps that you want to glue or merge together (2, 3, or 4). In the dialog box, the number of input raster maps is limited to four. On the command line, this limitation is not present.
First input map:	Select the first input raster map. Open the list box and select the desired input map, or drag a raster map directly from the Catalog into this box.
Second input map:	Select the second input raster map.
Third input map:	Optionally, select a third input raster map.
Fourth input map:	Optionally, select a fourth input raster map. You can always glue raster maps of the same domain type.
Last map on top:	Select this check box when for overlapping pixels the values, class names, IDs, or colors of the last map should be used. Then, for overlapping pixels, the values of the last map will appear on top of the other maps; undefined values in the last map will act as openings for the map under it, etc. Clear this check box when for overlapping pixels, the pixels of the first map should be used.
New domain:	Only when merging Class or ID maps that do not use the same domain: select this box if you want to store the output domain as a separate object (recommended). Subsequently, type a name for the new domain. Clear this check box to obtain an output Class or ID domain, which is stored by the output map (internal domain).
Output raster map:	Type a name for the output raster map that will contain all input maps.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created. Furthermore, a new georeference is automatically created for the output map (same name as output map), and, when needed and specified, a new domain is also created.

Command line

The Glue raster maps operation can be directly executed by typing one of the following expressions on the command line of the Main window:

OUTMAP = `MapGlue`(FirstInputMapName, SecondInputMapName [, MoreInput Maps])
OUTMAP = `MapGlue`(FirstInputMapName, SecondInputMapName [, MoreInput Maps] , `Replace`)
OUTMAP = `MapGlue`(FirstInputMapName, SecondInputMapName [, MoreInput Maps] , NewDomain)
OUTMAP = `MapGlue`(FirstInputMapName, SecondInputMapName [, MoreInput Maps] , NewDomain, `Replace`)

where:

OUTMAP	the name of your output raster map.
`MapGlue`	the command to start the Glue raster maps operation.
FirstInputMapName	the name of the first input raster map.
SecondInputMapName	the name of the second input raster map.
MoreInputMaps	optionally, you can specify more input raster map names, delimited by commas.
`Replace`	an optional parameter to use for overlapping pixels the values, class names, IDs or colors of the last input map. When this parameter is not used, then the values, class names, IDs or colors of the first input map will be used for overlapping pixels.
NewDomain	an optional parameter in case of merging Class or ID maps that do not have the same domain, to specify a name for the new output domain in which all input domain items will be merged. When input Class or ID domains are not the same and this parameter is not specified, the new output domain in which all input domain items are merged will be stored by the output map (internal domain).

On the command line, you can specify as many input raster maps as you like; i.e. you can merge as many raster maps as you like. When using the dialog box of this operation, only two, three or four input maps can be merged at a time.

When the definition symbol `=` is used, a dependent output map is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output map has been calculated. When Class or ID maps are merged that do not have the same domain, and the NewDomain parameter is used, a new output domain is also created. Furthermore, a georeference submap is created with the same name as the output map.

Algorithm

First, the operation checks whether the georeferences and the domain types of the input maps are compatible. If so, the operation can be performed. For more information on mergeable domain types, see Glue raster maps: functionality.

Output georeference

A new georeference submap is created with the same name as the output map.

- The pixel size of the first input map is used and the georeference is extended in X and Y direction so that all input maps fit into the output georeference.
- If needed, all input maps are resampled to this output georeference. The nearest neighbour resampling method is used in the same way as when resampling the values of a single raster map to a new georeference (see Resample).
- Usually, the input georeferences will use the same coordinate system, which covers the whole area already. In case the input georeferences use a different but compatible coordinate system (e.g. different projections), the coordinate system of the first input map is used and a coordinate transformation is performed for the other maps.

Output domain

- When merging maps with the same domain, the output map will also use that domain.
- When merging value maps, the value range for the output map is determined by the smallest minimum and the largest maximum of all input maps; the precision is the smallest precision of the input maps.
- When merging a value map with an Image, the output map always uses system domain `Value`. The smallest minimum and the largest maximum of the input maps determine the value range for the output map; the precision is the best precision of the input maps.
- When merging Class maps or when merging ID maps that do not have the same domain, a new output domain is created which contains all domain items, i.e. all classes or all IDs. The new domain is either stored as a separate object, or is stored by the output map (internal domain).
- When merging Pictures, a new output domain and a new representation are created which are both stored by the output map (internal domain and internal representation); the representation contains all colors of the input maps with a maximum of 256.
- When merging an image or a class, ID, or value map with a `Color` map, the output map will use the Color domain
- When merging a class map with an image or when merging a class map with a value map, the output map will use the `Color` domain.

Overlapping pixels

- By default, the last map appears on top of the first map.
- Undefined pixels in the map on top are considered transparent and act as openings.

7.3 Image Processing

7.3.1 Filter

Command line

The Filter operation can be directly executed by typing the following expression on the command line of the Main Window.

OUTMAP= `MapFilter`(InputMapName, *FilterName* | *FilterExpression*)

where:

OUTMAP	is the name of your output map.
`MapFilter`	is the command to start the Filter operation.
InputMapName	is the name of your input map.
FilterName	`avg3x3` | `binmajor` | `conn8to4` | `d2fdx2` | `d2fdxdy` | `d2fdy2` | `dfddn` | `dfdup` | `dfdx` | `dfdy` | `dilate4` | `dilate8` | `edgesenh` | `inbnd4` | `inbnd8` | `laplace` | `lifegame` | `majority` | `majundef` | `majzero` | `med3x3` | `med5x5` | `outbnd4` | `outbnd8` | `peppsalt` | `shadow` | `shrink4` | `shrink8` | *name of user-defined linear filter*
FilterExpression	`FilterLinear`(*rows,cols,expression*) | `Average`(*rows,cols*) | `RankOrder`(*rows,cols,rank[,threshold]*) | `Median`(*rows,cols[,threshold]*) | `Majority`(*rows,cols*) | `ZeroMajority`(*rows,cols*) | `UndefMajority`(*rows,cols*) | `Pattern`(*threshold*) | `FilterStandardDev`(*rows,cols*)

☞ For more information and some examples of using standard and user-defined filters on the command line, refer to Filters : user-defined filters.

When the definition symbol `=` is used, a dependent output map is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output map has been calculated.

7.3.2 Filter types

Based on the algorithm used by filters, the following types of filters can be distinguished:

- Linear filters
  - Gradient or derivative filters
- Rank order and median filters
- Majority filters
- Binary filters
- Pattern filters

- Standard deviation filters

7.3.2.1 Linear filters

Linear (convolution) filters consist of a matrix with values and a gain factor. When considering a linear filter of size 3x3, the 9 matrix values are multiplied with 9 pixel values in the input raster map, this is summed and then multiplied with the gain factor. The result is assigned to the center pixel in the output map.

Standard linear filters
The following standard filters are linear filters,: AVG3X3, EDGESENH, LAPLACE, and SHADOW. Also the standard gradient filters are linear filters: DFDX, DFDY, DFDDN, DFDUP, D2FDX2, D2FDY2, D2FDXDY.

User-defined linear filters
You can create, edit and store your own linear filters, for instance from the Filter: dialog box, through the File menu in the Main window or by clicking the NewFilter item in the Operation-list. For more information, refer to the Create Filter and Edit Filter dialog boxes. You need to specify the size of your filter, fill out the values in the matrix and specify a gain factor. Experienced users may wish to experiment with the definition of a linear filter by an expression on the command line. For more information on using standard and user-defined linear filters, refer to User-defined linear filters.

Furthermore, you can interactively define an Average filter in the Filter dialog box or by an expression on the command line. For more information, refer to User-defined average filters.

7.3.2.2 Gradient or derivative filters (technical information)

Linear (convolution) filters consist of a matrix with coefficients and a gain factor. When considering a linear filter of size 3×3, the 9 matrix coefficients are multiplied with 9 pixel values in the input raster map, this is summed and then multiplied with the gain factor. The result is assigned to the center pixel in the output map.

The following standard filters are known as gradient filters or derivative filters: DFDX, DFDY, DFDN, DFDUP, D2FDX2, D2FDY2, D2FDXDY. For each group of pixel values considered, they calculate the first or second derivative in one or more directions. Derivative filters are often used in relation to slope calculations.

The standard derivative filters mentioned above have a typical size of 1×5, 5×1, or 5×5. Of course, you can also create your own linear gradient filters.

This topic is intended for people who would like to understand the mathematical background of the coefficients in the matrices of derivative filters. First, 3×3 filters will be described, then 5×5 filters.

3×3 filters

Introduction

The simplest mathematical situation is represented when using a 3×3 filter, however these filters may not be exact enough to calculate slopes. A 3×3 filter uses the 9 input values to calculate a value for the center pixel in the output map.

To do calculations with a 3×3 filter, a local coordinate system (X,Y) is defined around the current center pixel, as:

(-1, 1)	(0, 1)	(1, 1)
(-1, 0)	(0, 0)	(1, 0)
(-1, -1)	(0, -1)	(1, -1)

Both X and Y can have the values -1, 0, and 1.

When calculating the first derivative only in the x-direction, y remains 0, and x can have value -1, 0, or 1.

To calculate derivatives, a continuous function is needed. The input pixel values are to be described by a function `f(x)` where:

f_0 = input pixel value of the center pixel; x=0
f_{-1} = input pixel value of the pixel to the left of the center pixel; x=-1
f_1 = input pixel value of the pixel to the right of the center pixel; x=1

Formulas

As continuous function, a polynomial function is used. With 9 known values, a second order polynomial can be fitted through these points.

$$f_{xy} = a_{00} + a_{10}\,x + a_{20}\,x^2 + a_{01}\,y + a_{11}\,xy + a_{21}\,x^2y + a_{02}\,y^2 + a_{12}\,xy^2 + a_{22}\,x^2y^2 \quad (1)$$

When we are only interested in the first derivative in x-direction, formula 1 can be simplified by substituting y with 0 to:

$$f_x = a_0 + a_1\,x + a_2\,x^2 \quad (2)$$

The function for the first derivative equals:

$$df/dx = f'x = a_1 + 2a_2\,x \quad (3)$$

The second derivative equals:

$$d^2f/dx^2 = f''x = 2a_2 \quad (4)$$

Because we are interested in the derivatives at the central pixel where x=0, in formulas 3 and 4, x can be substituted with 0:

$df/dx = f'_0 = a_1$ (5)
$d^2f/dx^2 = f''_0 = 2a_2$ (6)

To find f_{-1}, f_0 and f_1, in formula 2, x is substituted with values -1, 0, and 1:

$f_{-1} = a_0 - a_1 + a_2$ (7)
$f_0 = a_0$ (8)
$f_1 = a_0 + a_1 + a_2$ (9)

Then, by elimination, a_1 and a_2 are found:

$a_1 = (f_1 - f_{-1}) / 2$ (10)
$a_2 = (f_{-1} - 2f_0 + f_1) / 2$ (11)

A 3×3 first derivative filter for the x-direction will thus read:

0	0	0
-1	0	1
0	0	0

Gain factor = 1/2 = 0.5

A 3×3 second derivative filter for the x-direction will thus read:

0	0	0
1	-2	1
0	0	0

Gain factor = 1/2 = 0.5

To calculate the second derivative in both the x-direction and the y-direction, $f''=d^2f/dxdy$, formula 1 is needed. After the substitution of all 9 coordinates in the equation, and solving them, the results are:

$d^2f / dxdy = a_{11}$ (12)
$a_{11} = (f_{1,1} + f_{-1,-1}) - (f_{-1,1} + f_{1,-1}) / 4$ (13)

A 3×3 second derivative filter for both the x-direction and the y-direction will thus read:

-1	0	1
0	0	0
1	0	-1

Gain factor = 1/4 = 0.25

5×5 filters

Calculation of matrix coefficients for 5×5 filters follows the same method as for 3×3 filters. Although 5×5 filters are a little bit more complicated, they will produce more accurate results.

Again a local coordinate system is used around the current center pixel as:

(-2,2)	(-1,2)	(0,2)	(1,2)	(2,2)
(-2,1)	(-1,1)	(0,1)	(1,1)	(2,1)
(-2,0)	(-1,0)	(0,0)	(1,0)	(2,0)
(-2,-1)	(-1,-1)	(0,-1)	(1,-1)	(2,-1)
(-2,-2)	(-1,-2)	(0,-2)	(1,-2)	(2,-2)

Both x and y can have the values -2, -1, 0, 1, and 2.

The polynomial function f_x and its derivatives are:

$$fx = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x \quad (14)$$
$$df/dx= f'x = a_1 + 2a_2 x + 3a_3 x^2 + 4a_4 x^3 \quad (15)$$
$$d^2f/dx^2 = f''x = 2a_2 + 6a_3 x + 12a_4 x^2 \quad (16)$$

By substituting x in formulas 15 and 16 with 0, the previous equations 5 and 6 are obtained again:

$$df/dx = f'_0 = a_1 \quad (5)$$
$$d^2f/dx^2 = f''_0 = 2a_2 \quad (6)$$

Substituting x in formula 14 with values -2, -1, 0, 1, and 2 gives f_{-2}, f_{-1}, f_0 , f_1 and f_2:

$$f_{-2} = a_0 - 2a_1 + 4a_2 - 8a_3 + 16a_4 \quad (17)$$
$$f_{-1} = a_0 - a_1 + a_2 - a_3 + a_4 \quad (18)$$
$$f_0 = a_0 \quad (19)$$
$$f_1 = a_0 + a_1 + a_2 + a_3 + a_4 \quad (20)$$
$$f_2 = a_0 + 2a_1 + 4a_2 + 8a_3 + 16a_4 \quad (21)$$

With some restructuring, the following equations are obtained:

$$f_1 + f_{-1} = 2a_0 + 2a_2 + 2a_4 \quad (22)$$
$$f_1 - f_{-1} = 2a_1 + 2a_3 \quad (23)$$
$$f_2 + f_{-2} = 2a_0 + 8a_2 + 32a_4 \quad (24)$$
$$f_2 - f_{-2} = 4a_1 + 16a_3 \quad (24)$$

The matrix coefficients for a 1×5 filter, which calculates the first derivative in x-direction, are found by the elimination of a_3: subtracting equation 23 eight times from equation 25. This results in:

$$a_1 = (f_{-2} - 8f_{-1} + 8f_1 - f_2) / 12 \quad (26)$$

A 1×5 first derivative filter for the x-direction will thus read:

1	-8	0	8	-1

Gain factor = 1/12 = 0.0833333

This is exactly what ILWIS uses for the DFDX filter.

The matrix coefficients for a 1×5 filter, which calculates the second derivative in x-direction, are found by the elimination of a_4: subtracting equation 22 sixteen times from equation 24 and substituting f_0 for a_0. This results in:

$$a_2 = (-f_{-2} + 16f_{-1} - 30f_0 + 16f_1 - f_2) / 24 \qquad (27)$$

A 1×5 second derivative filter for the x-direction will thus read:

-1	16	-30	16	-1

Gain factor = 1/24 = 0.0416667

This is exactly the D2FDX2 filter.

7.3.2.3 Rank order and median filters

Rank order filters have a certain size, but do not have any matrix values or gain factor. A rank order filter of size 3x3 for example, examines 9 pixel values of the input map at a time, sorts the values from small to large, and selects for the output value that value which is encountered at a certain rank order number. So one value of the pixel values examined becomes the output value, without any calculation performed on the values itself.
When a threshold is set, the value of the center pixel will only be replaced with the new value if the difference between the original and new value is smaller than or equal to the threshold.

Standard rank order filters
Standard rank order filters are the median filters: MED3x3, MED5x5. For each 9 pixels considered, the MED3 filter always assigns the value of rank 5 to the center pixel in the output map. For each 25 pixels considered, the MED5 filter always assigns the value of rank 13 to the center pixel in the output map. The median filters can for instance be used to smooth an image.

User-defined rank order and median filters
In the Filter dialog box as well as on the command line, you can define your own rank order filter and median filter. For a rank order filter, specify the size of the filter, the rank order number and an optional threshold. In this way you can obtain for example the minimum or maximum value of a number of pixels. For median filters, specify the size of the filter and an optional threshold. For more information and some examples, refer to User-defined rank order and median filters.

7.3.2.4 Majority filters

For each group of pixels considered in the input map, a majority filter assign the predominant (=mostly frequently occurring) value or class name of these to the center pixel in the output map. Undef-majority filters only do this when the center pixel in the input map is undefined; zero-majority filters only do this when the value of the center pixel in the input map is 0.

Undef-Majority filters are often used as a post-classification operation to reduce the number of undefined pixels in the output map of an image classification.

Standard majority filters
For each 3×3 pixels considered, the standard MAJORITY filter assigns the predominant value or class name to the central pixel in the output map. If no predominant value is found, for instance when all 9 input pixels have a different value or class name, the value or class name encountered first is used as output.

The standard undef-majority filter (MAJUNDEF) will only assign the predominant value or class name to the central pixel in the output map if the central pixel in the input map is undefined.

The standard zero-majority filter (MAJZERO) will only assign the predominant value or class name to the central pixel in the output map if the central pixel in the input map has value zero.

User-defined majority filters
In the Filter dialog box as well as on the command line, you can define your own majority filters: specify the size of the filter and whether a condition has to be used. For more information, refer to user-defined majority filters.

7.3.2.5 Binary filters

Binary filters regard the input map as a binary map. This means that zero values are regarded as zero, and all other values as one. Depending on the central pixel value and its 8 neighbours, the filter produces a zero or one as output values. Binary filters are widely used for morphologic filtering.

The 9 binary pixels examined by a binary filters are put in a special order or bit position as below (where 0 means last position, 1 the one but last position, etc.):

5	6	7
4	8	0
3	2	1

This results in a number of 9 binary digits (when only lower right pixel is true: 000000010).

Thus depending on the position of true pixels a unique number is obtained. To decide whether the central pixel should be assigned a 0 or 1, this number is looked up in a table, which is present in each binary filter itself.

Standard binary filters
The standard binary filters are: BINMAJOR, CONN8TO4, DILATE4, DILATE8, SHRINK4, SHRINK8, INBND4, INBND8, OUTBND4, OUTBND8, PEPPSALT, LIFEGAME.

User-defined binary filters
There are no possibilities in ILWIS to define your own binary filters. However, you could copy an existing binary filter (*.FIL), edit it with an ASCII editor like Notepad and thus create your own binary filter.
Only Pattern filters, which work more or less the same as binary filters, can be user-defined via the Filter dialog box or the command line.

7.3.2.6 Pattern filters

With a pattern filter you can detect: **areas** where pixels have more or less the same value, **locations** where the values of all neighbours are largely different from the center pixel (outliers), and the **directions** in which differences between a center pixel and its neighbours are found. A pattern filter always works in a 3×3 environment and works on images and other raster maps with a value domain.

Whether or not any of the 8 neighbours is considered to have more or less the same value as the center pixel, is determined by the threshold value that you have to specify. When the absolute difference between a neighbour and the center pixel is smaller than or equal to the threshold value, the answer is true. For each true neighbour, a certain bit is set. The value assigned to the center pixel in the output map is the bit-wise combination of all true neighbours. For more information, refer to Example pattern filters.

The pattern filter works according to the following rules:

- All neighbours are false: the final output value is 0, which means that there are large differences between the central pixel and all its neighbours (outlier).
- All neigbours are true: the output value is 255, which means that there are small differences between the central pixel and its neighbours (area),
- All other output values bit-wise represent the directions in which differences are found.

Standard pattern filters
There is no standard pattern filter stored on disk.

User-defined pattern filters
A pattern filter always works in a 3×3 environment. In the Filter dialog box as well as on the command line, you can define a pattern filter; specify the threshold value, which has to be used. For more information and some examples, refer to User-defined pattern filters.

7.3.2.7 Standard deviation filters

For each group of pixels in the input map, a standard deviation filter calculates the standard deviation and assigns this value to the center pixel in the output map.

Standard deviation filters can be useful for radar images. The interpretation of radar images is often difficult: you can not rely on spectral values because of backscatter (return of the pulse sent by the radar). This often causes a lot of 'noise'. By using a standard deviation filter, you may be able to recognize some patterns.

The formula to calculate the standard deviation reads:

$$\text{stddev} = \sqrt{\frac{\sum (x_i - \overline{x})^2}{r \cdot c - 1}}$$

Where:

x_i are the individual pixel values of the input map considered by the filter is the mean of the pixel values considered by the filter
r is the size of the filter in rows
c is the size of the filter in columns

Standard standard deviation filters
There is no standard standard deviation filter stored on disk.

User-defined standard deviation filters
In the Filter dialog box as well as on the command line, you can define a standard deviation filter: specify the size of the filter. For more information and some examples, refer to User-defined standard deviation filters.

7.3.2.8 User-defined filters

Besides using the ILWIS standard filters, you can:

- create, edit and store your own linear filters,
- define a filter in the Filter dialog box by specifying some parameters,
- define a filter on the command line by an expression.

Creating linear filters
You can create, edit and store your own linear filters, for example for example by clicking the create button in the Filter dialog box, by selecting Create Filter command from the File menu of the Main window, or by double-clicking the NewFilter item in the Operation-list.
You can specify the size of the linear filter, insert your own values in the matrix and specify a gain factor. For more information, refer to the Create Filter dialog box and the Edit Filter dialog box.

Defining a filter in the Filter dialog box
In the Filter dialog box, you can select any standard (predefined) filter. Furthermore, you can define the following filters according to your wishes: Average filter, Rank Order filter, Median filter, Majority filter, Pattern filter, and Standard Deviation filter. You can specify the filter size, a rank or the threshold.

Defining a filter by an expression on the command line
Finally, advanced users can define an Average filter, a Rank Order filter, a Median filter, a Majority filter, a Pattern filter, a Standard Deviation filter or even a Linear filter by typing an expression on the command line of the Main window.

Restrictions of user defined filters
These restrictions apply to all user defined filters, except the pattern filter.

- Both the number of rows and the number of columns defining the filter size must be odd values.
- The product of *rows*×*cols* defining the filter size may not exceed 8000.

Restrictions of user defined filters (specific)

- When using the rank order filter, the minimum rank is 1 and the maximum rank cannot exceed the total size of the filter (*rows*×*cols*).

7.3.2.9 User-defined linear filters

Create, edit and store a linear filter on disk

A user-defined linear filter can be created:

- by clicking the create button in the Filter dialog box, or
- by choosing the Create Filter command from the Main window File menu, or
- by double-clicking the NewFilter command in the Operation-list.

The Create Filter dialog box and the Edit Filter window will appear. You can specify the size of the filter, fill out values in the matrix, and specify a gain factor.

Edit an existing user-defined linear filter

An existing user-defined linear filter can be edited:

- by clicking your filter with the right mouse button in the Catalog, and subsequently selecting the Open command from the context-sensitive menu,
- by choosing the Edit Object command from the Edit menu in the Main window, and subsequently selecting your filter.

The Edit Filter window will appear.

The standard linear filters cannot be edited.

User-defined average filters

Additional possibilities are provided to for user-defined average filters, via the Filter dialog box or via the command line, see Filters : user-defined average filters.

Selecting an existing linear filter in the Filter dialog box

In the Filter dialog box, you can select any standard linear filter or any linear filter you created yourself.

- Open the Filter dialog box,
- for Filter Type, select: Linear,
- for Filter Name, select: AVG3X3, D2FDX2, D2FDXDY, D2FDY2, DFDDN, DFDUP, DFDX, DFDY, EDGESENH, LAPLACE, Shadow, or your own linear filter.

Using existing linear filters on the command line

To use an existing linear filter from the command line, type the following expression on the command line of the Main window.

OUTMAP = `MapFilter`(InputMapName, *FilterName*) where:

OUTMAP	is the name of your output map.
`MapFilter`	is the command to start the Filter operation.
InputMapName	is the name of your input map.
FilterName	is either the name of a standard filter on disk, then fill out one of the following standard linear filter names: `av3x3` | `d2fdx2` | `d2fdxdy` | `d2fdy2` | `dfddn` | `dfdup` | `dfdx` | `dfdy` | `edgesenh` | `laplace` | `shadow` or the name of a linear filter which you created yourself.

Defining a linear filter on the command line (advanced)
Advanced users may wish to experiment with the definition of a linear filter by an expression. The syntax for the command line is:

OUTMAP= `MapFilter`(InputMapName, `FilterLinear`(*rows*, *cols*, expression))

where:

OUTMAP	is the name of your output map.
`MapFilter`	is the command to start the Filter operation.
InputMapName	is the name of your input map.
`FilterLinear`	is the command to define a linear filter.
rows	are the number of rows of your linear filter.
cols	are the number of columns of your linear filter.
expression	is an expression in which you can use x, y, and r to calculate the linear filter's matrix values, where:
x	distance to the center cell of the matrix in x-direction; the center cell of the matrix has position (0,0); the distance in x-direction increases to the right.
y	distance to the center cell of the matrix in y-direction; the center cell of the matrix has position (0,0); the distance in y-direction increases downwards.
r	Euclidean distance to the center cell of the matrix: $\sqrt{(x^2+y^2)}$

Examples of linear filter expressions are for instance:

```
FilterLinear(5,5,2x+y)
```

Defines a 5 by 5 shadow filter where the illumination is from the north-west.

```
FilterLinear(5,5,-2x-y)
```

Defines a 5 by 5 shadow filter where the illumination is from the south-east.

```
FilterLinear(5,5,iff(r>3,0,4-r))
```

Defines an average filter in which the matrix values decrease linearly, starting from the center cell of the matrix towards the borders of the matrix, according to Euclidean distance.

7.3.2.10 User-defined average filters

The standard average filter is AVG3X3; it calculates the average value of 9 pixel values considered. In the Filter dialog box as well as on the command line, you may define other average filters of any user-defined size.

Using the Filter dialog box

1. To use the standard Average filter AVG3X3:
   - Open the Filter dialog box,
   - for Filter Type, select: Linear,
   - for Filter Name, select: AVG3X3.

2. To define your own average filter:
   - Open the Filter dialog box,
   - for Filter Type, select: Average,
   - specify the size of the filter.

See also the example below.

Using the command line (advanced)

1. To use the standard Average filter AVG3X3, type on the command line of the Main window:

   OUTMAP = `MapFilter`(InputMapName,AVG3X3)

   where:

OUTMAP	is the name of your output map.
`MapFilter`	is the command to start the Filter operation.
InputMapName	is the name of your input map.
`Avg3x3`	is the name of the standard average filter.

2. To define your own Average filter, type the following expression on the command line of the Main window:

   OUTMAP = `MapFilter`(InputMapName,`Average`(*rows* , *cols*))

   where:

OUTMAP	is the name of your output map.
`MapFilter`	is the command to start the Filter operation.
InputMapName	is the name of your input map.
`Average`	is the command to define an average filter.
rows	are the number of rows of your average filter.
cols	are the number of columns of your average filter.

Example

To use an average filter which considers each **5 vertically neighbouring pixels**:

- in the Filter : dialog box, choose for Filter type: Average, specify 5 for the number of rows and 1 for the number of columns,
- on the command line, use a filter definition of: `Average(5,1)`.

7.3.2.11 User-defined rank order and median filters

Standard rank order filters are MED3x3 and MED5x5; these filters sort 9 respectively 25 pixel values and assign the 5th respectively the 13th value (median value) to the center pixel in the output map. In the Filter dialog box as well as on the command line, you may define other rank order and median filters of any user-defined size and assigning any user-defined rank order value to the central pixel.

Using the Filter dialog box

1. To use one of the standard rank order filters:
   - Open the Filter dialog box,
   - for Filter Type, select: Rank Order,
   - select the Predefined check box,
   - for Filter Name, select: Med3x3 or Med5x5.

2. To define your own rank order filter:
   - Open the Filter dialog box,
   - for Filter Type, select: Rank Order,
   - clear the Predefined check box,
   - specify the size of the filter,
   - specify the rank order value which should be assigned to the central pixel,
   - optionally specify a threshold: if the absolute difference between the result and the original value is larger than the threshold, the original value is kept; if the difference is smaller than or equal to the threshold, the ranked value is assigned.

   This ensures that larger differences between neighbouring pixels remain in the output, while small variations are removed.

3. To define your own median filter:
   - Open the Filter dialog box,
   - for Filter Type, select: Median,
   - specify the size of the filter,
   - optionally specify a threshold.

See also the example below.

Using the command line (advanced)

1. To use a standard Rank order filter, type one of the following expressions on the command line of the Main window:

OUTMAP = `MapFilter`(InputMapName, `Med3x3`)
OUTMAP = `MapFilter`(InputMapName, `Med5x5`)

where:

OUTMAP	is the name of your output map.
`MapFilter`	is the command to start the Filter operation.
InputMapName	is the name of your input map.
`Med3x3`	is the name of the standard Median 3x3 filter.
`Med5x5`	is the name of the standard Median 5x5 filter.

2. To define your own Rank order filter, type one of the following expressions on the command line of the Main window:

 OUTMAP = `MapFilter`(InputMapName, `Rankorder`(*rows*,*cols*,*rank*))
 OUTMAP = `MapFilter`(InputMapName, `Rankorder`(*rows*,*cols*,*rank*,*threshold*))

 where:

OUTMAP	is the name of your output map.
`MapFilter`	is the command to start the Filter operation.
InputMapName	is the name of your input map.
`Rankorder`	is the command to define a rank order filter.
rows	are the number of rows of your rank order filter.
cols	are the number of columns of your rank order filter.
rank	is the rank order number that determines the output value.
threshold	when a threshold is set, the value of the center pixel will only be replaced by the new value if the difference between the original and new pixel value is smaller than or equal to the threshold.

3. To define your own Median filter, type one of the following expressions on the command line of the Main window:

 OUTMAP = `MapFilter`(InputMapName,`Median`(*rows*,*cols*))
 OUTMAP = `MapFilter`(InputMapName,`Median`(*rows*,*cols*,*threshold*))

 where:

OUTMAP	is the name of your output map.
`MapFilter`	is the command to start the Filter operation.
InputMapName	is the name of your input map.
`Median`	is the command to define a median filter.
rows	are the number of rows of your median filter.
cols	are the number of columns of your median filter.
threshold	when a threshold is set, the value of the center pixel will only be replaced by the new value if the difference between the original and new pixel value is smaller than or equal to the threshold.

Examples

1. To obtain the minimum of each 3×3 neighbouring pixels:
   - in the Filter dialog box, select for the Filter type Rank Order,
     clear the Predefined check box,
     specify 3 for the number of rows and 3 for the number of columns,
     specify 1 for the rank,
     and optionally specify a threshold.
   - on the command line, use a filter definition of: `Rankorder(3,3,1)`

Rank 1 means the smallest value of each 9 (3 by 3) pixel values examined will be used for the output pixel.

2. To obtain the maximum of each 7 horizontally neighbouring pixels:
   - in the Filter dialog box,
     select for the Filter type Rank Order
     clear the Predefined check box,
     specify 1 for the number of rows and 7 for the number of columns,
     specify 7 for the rank,
     and optionally specify a threshold.
   - on the command line, use a filter definition of: `Rankorder(1,7,7)`

   Rank 7 means that the largest value of each 7 (1 by 7) pixel values examined is used as output value.

3. To obtain the median value of each 5 horizontally neighbouring pixels:
   - in the Filter dialog box,
     select for the Filter type Median,
     clear the Predefined check box,
     specify 1 for the number of rows and 5 for the number of columns,
     and optionally specify a threshold.
   - on the command line, use a filter definition of: `Median(1,5)`.

☞ With Neighbourhood operators similar operations can be performed as with rank order filters: for example, the minimum, maximum, sum and predominant pixel value of each 9 pixels considered can be assigned as output value to the central pixel. For more information, refer to MapCalc special: neighbourhood operations.

☞ With the Aggregate Map operation, similar results can be obtained. Aggregate Map however moves from block to block in the map, while the Filter operation moves pixel by pixel.

7.3.2.12 User-defined majority filters

The standard majority filters are Majority, MajUndef, and MajZero; they work in a 3 by 3 environment and assign the predominant value of the surrounding pixels to the central pixel. In the Filter dialog box as well as on the command line, you may define other majority filters of any user-defined size.

Using the Filter dialog box

1. To use one of the standard Majority filters:
   - Open the Filter dialog box,
   - for Filter Type, select: Majority,
   - select the Predefined check box,
   - for Filter Name, select: Majority, MajUndef or MajZero.

2. To define your own Majority filter:
   - Open the Filter dialog box,
   - for Filter Type, select: Majority,
   - clear the Predefined check box,

- specify the size of the filter,
- in case you want to replace only central values which are undefined, select the Undefined check box.

See also the example below.

Using the command line (advanced)

1. To use a standard Majority filter, type one of the following expressions on the command line of the Main window:

   OUTMAP = `MapFilter`(InputMapName, `Majority`)
   OUTMAP = `MapFilter`(InputMapName, `MajUndef`)
   OUTMAP = `MapFilter`(InputMapName, `MajZero`)

   where:

OUTMAP	is the name of your output map.
`MapFilter`	is the command to start the Filter operation.
InputMapName	is the name of your input map.
`Majority`	is the name of the standard Majority filter.
`MajUndef`	is the name of the standard Undef-Majority filter.
`MajZero`	is the name of the standard Zero-Majority filter.

2. To define your own Majority filter, type one of the following expressions on the command line of the Main window:

   OUTMAP = `MapFilter`(InputMapName, `Majority`(*rows,cols*))
   OUTMAP = `MapFilter`(InputMapName, `UndefMajority`(*rows,cols*))
   OUTMAP = `MapFilter`(InputMapName, `ZeroMajority`(*rows,cols*))

   where:

OUTMAP	is the name of your output map.
`MapFilter`	is the command to start the Filter operation.
InputMapName	is the name of your input map.
`Majority`	is the command to define a Majority filter.
`UndefMajority`	is the command to define an Undef-Majority filter.
`ZeroMajority`	is the command to define a Zero-Majority filter.
rows	are the number of rows of your majority filter.
cols	are the number of columns of your majority filter.

Example

To use a majority filter which considers each **5 horizontally neighbouring pixels**:

- in the Filter dialog box, choose for Filter type: Majority,
  clear the Predefined check box, and
  specify 1 for the number of rows and 5 for the number of columns,
- on the command line, use a filter definition of: `Majority(1,5)`

7.3.2.13 User-defined pattern filters

With a pattern filter, you can detect areas where pixels have more or less the same value, outliers where the values of all neighbouring pixels are very different from the center pixel, and the directions in which differences between neighbouring pixel values are found. Whether or not 3×3 neighbouring pixels are considered to have more or less the same values, is determined by the threshold value that you have to specify.

There is no standard pattern filter stored on disk. A pattern filter always works in a 3×3 environment. In the Filter dialog box as well as on the command line, you can define the threshold value, which has to be used by the pattern filter.

Using the Filter dialog box

To use a pattern filter:

- Open the Filter dialog box,
- for Filter Type, select: Pattern,
- specify a threshold value which determines whether the absolute differences between the central pixel and its neighbours are considered large (edge) or small (area).

Using the command line (advanced)

To define a pattern filter, type the following expression on the command line of the Main window:

OUTMAP = `MapFilter`(InputMapName, `Pattern`(*threshold*))

where:

OUTMAP	is the name of your output map.
`MapFilter`	is the command to start the Filter operation.
InputMapName	is the name of your input map.
`Pattern`	is the command to define a pattern filter.
threshold	specify a value for the threshold which determines whether the absolute differences between the central pixel and its neighbours are considered large (outlier or direction of edge) or small (area).

7.3.2.14 User-defined standard deviation filters

There is no standard standard deviation filter stored on disk. You always have to define a standard deviation filter yourself, either through the Filter dialog box or by an expression on the command line of the Main window.

Using the Filter dialog box

To define your own standard deviation filter:

- Open the Filter dialog box,
- for Filter Type, select: Standard Deviation,
- specify the size of the filter.

Using the command line (advanced)
To define your own Standard Deviation filter, type the following expression on the command line of the Main window:

OUTMAP = `MapFilter`(InputMapName, `FilterStandardDev`(*rows*,*cols*))

where:

OUTMAP	is the name of your output map.
`MapFilter`	is the command to start the Filter operation.
InputMapName	is the name of your input map.
`FilterStandardDev`	is the command to define a standard deviation filter.
rows	are the number of rows of your standard deviation filter.
cols	are the number of columns of your standard deviation filter.

Example
To use a standard deviation filter which considers each **7 vertically neighbouring pixels**:

- in the Filter: dialog box, choose for Filter type: Standard Deviation, specify 7 for the number of rows and 1 for the number of columns,
- on the command line, use a filter definition of: `FilterStandardDev(7,1)`

7.3.3 Color composite

Functionality

A color composite is created by combining 3 raster images (bands/maps). One band is displayed in shades of red, one in shades of green and one in shades of blue.

A color composite can be created:

- to serve as a background image during sampling and subsequent image classification,
- for visual interpretation purposes, a printed color composite may be useful as a field map, but you can also use a color composite as a background image during on-screen digitizing, or
- for illustration purposes, for instance by using a color composite as a drape over a 3D model.

☞ This operation creates a permanent color composite map. When your graphics board is configured to use more than 256 colors, you can also interactively display a color composite by selecting the Show MapList as Color Composite command from the Operations, Visualization menu. (To customize the color depth to for instance High Color 16-bit or True Color 24-bit, use Display Settings in Windows' Control Panel). By creating an interactive color composite (very suitable for sampling or on-screen digitizing), you can easily change intervals, select other bands, etc. Another advantage is that besides maps with the Image domain, also maps with a Value domain are accepted. The resulting color composite is displayed in a map window. To store such a color composite, you can save the map window as a map view.

General information on color composites

- A color composite gives a visual impression of 3 raster bands. Putting the three bands together in one color composite map can give a better visual impression of the reality on the ground, than by displaying one band at a time. Examples of color composites are false color (or IR) images and 'natural color' images.
- The input pixel values of each band are measures of the amount of reflection in a certain wavelength interval. The values in the output color composite map just refer to certain colors; the output values themselves have no meaning.
- Before creating a color composite, you might filter the bands in order to increase sharpness of features of interest.

The Color composite operation offers several ways to create color composites:

- **Standard**:
  - Linear stretching: input values are linearly stretched; user-defined input intervals;
  - Histogram equalization: input values are equally divided over output colors; user-defined input intervals;
- **Dynamic** (Heckbert): input values are automatically distributed over a user-defined number of colors.
- **24 Bit RGB**:
  - Linear stretching: input values are linearly stretched; user-defined input intervals;
  - Histogram equalization: input values are equally divided over output colors; user-defined input intervals;
- **24 Bit HSI**: input values are interpreted as hue, saturation and intensity.

The different methods of creating a color composite are merely a matter of scaling the input values over the output colors. See below Color Composite: algorithm for the exact methods.

Input map requirements

The three input maps should use the Image domain. A georeference is not required for the input maps. If the maps do have a georeference, all input maps should use the same georeference.

Domain of output map

For a **standard** color composite: the operation always uses system Picture domain `ColorCmp` for the output color composite. This domain always uses system representation `ColorCmp`.
For a **dynamic** color composite: the operation creates a new domain (type Picture) for the output color composite and a new representation for this domain. This output domain and representation are always stored within the output map (internal domain and internal representation).
For a **24-bit** colors composite: the operation always uses the Color domain for the output color composite.

Georeference of output map

The output color composite always uses the same georeference as the input maps.

- ☞ If the user is interested in the image as a whole, it is best to use the Dynamic option. This usually results in a composite with good contrast. The Dynamic option does not take into account the structure of the input bands. Therefore, if the user is interested in specific intervals of the input bands, it is better to use the Standard option, using linear stretching. However, if the pixels that are of less interest can be masked, the user can calculate a Dynamic composite using only the pixels that are of interest.
- ☞ When there are more bands available than can be used to create a color composite (e.g. 7 TM-bands), you can first calculate the Optimum Index Factor (OIF); this may help you to decide which bands to select for a color composite.
- ☞ The reverse process of creating a color composite is color separation.
- ☞ For more information on internal domains and representations, refer to How to open internal domains/representations.

Dialog box

Dialog box options for 24 Bit Color Composite

24 bit: Select this check box if you want to create a map that can be displayed in 24-bit graphic mode (domain Color). Clear this check box if you want to create a color composite, which uses a Picture domain.

RGB: Creates a 24-bit color composite with Red, Green and Blue bands as input. See Standard Color Composite options below.

HSI: Creates a 24-bit color composite with Hue, Saturation and Intensity bands as input.

You should only select these 24-bit options, if your graphic board is configured to use more than 256 colors, for instance High Color 16-bit or True Color 24-bit (see Display Settings in Windows' Control Panel).

Dialog box options for Standard Color Composite

Standard: Select the Standard option button when you want to use the standard color composite representation, which has six shades of red, six shades of green and six shades of blue (total 216 colors).

Linear Stretching: Select Linear Stretching if you want to obtain intervals of equal length (in terms of input values) for the output colors. Histogram equalization: Select Histogram Equalization if you want to obtain an equal number of pixels for the different output colors.

Percentage: Select this check box to define input intervals by a percentage of pixels to be ignored on both sides of the input map's histogram. Clear this check box to specify input intervals by a minimum and maximum value of each input map.

Dialog box options for Dynamic (Heckbert) Color Composite

Dynamic:	Select the Dynamic option button when you want an automatic division of input values over a number of output colors.
Colors:	Enter a value for the number of colors of which the dynamic color composite should consist (integer value between 2 and 255).

General dialog box options

Red Band:	Select a raster map to be displayed in shades of red. Open the list box and select the appropriate raster map, or directly drag a raster map from the Catalog into this box.
Green Band:	Select a raster map to be displayed in shades of green.
Blue Band:	Select a raster map to be displayed in shades of blue.
Output raster map:	Type a map name for the output color composite.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created. When the Dynamic option is used, the output map will use an internal Picture domain, which has an internal representation.

☞ This operation creates a permanent color composite map. When your graphics board is configured to use more than 256 colors, you can also interactively display a color composite by selecting the Show MapList as Color Composite command from the Operations, Visualization menu. (To customize the color depth to for instance High Color 16-bit or True Color 24-bit, use Display Settings in Windows' Control Panel). By creating an interactive color composite, you can easily change intervals, select other bands, etc. The resulting color composite is displayed in a map window. To store such a color composite, you can save the map window as a map view.

Command line

Typing one of the following expressions on the command line of the Main Window can directly create a color composite:

OUTMAP=	`MapColorComp`(MapList, *range1*, *range2*, *range3*)
OUTMAP=	`MapColorCompLinear`(MapList, *range1*, *range2*, *range3*)
OUTMAP=	`MapColorCompHistEq`(MapList, *range1*, *range2*, *range3*)
OUTMAP=	`MapHeckbert`(MapList, *NrColors*)
OUTMAP=	`MapColorComp24`(MapList)
OUTMAP=	`MapColorComp24`(MapList, *range1*, *range2*, *range3*)

OUTMAP= `MapColorComp24Linear`(MapList)
OUTMAP= `MapColorComp24Linear`(MapList, *range1, range2, range3*)
OUTMAP= `MapColorComp24HistEq`(*MapList,range1, range2, range3*)
OUTMAP= `MapColorComp24HSI`(MapList)

Where:

OUTMAP	is the name of your output color composite.
`MapColorComp`	is the command start the Color composite operation using linear stretching.
`MapColorCompLinear`	is the command start the Color composite operation using linear stretching.
`MapColorCompHistEq`	is the command to start the Color composite operation using histogram equalization.
`MapHeckbert`	is the command to start the Color composite operation using the Heckbert dynamic algorithm.
`MapColorComp24`	is the command to start the 24-bit Color composite operation using linear stretching.
`MapColorComp24Linear`	is the command to start the 24-bit Color composite operation using linear stretching.
`MapColorComp24HistEq`	is the command to start the 24-bit Color composite operation using histogram equalization.
`MapColorComp24HSI`	is the command to start the 24-bit Color composite operation using hue, saturation, and intensity.
MapList	is the name of an existing map list which contains 3 raster maps with the `Image` domain, or the definition of a map list as: `mlist`(*ImageRed, ImageGreen, ImageBlue*). For more information, see the examples below.
range1..3	for each band, the intervals of input values to be used as *min:max* or as *perc*
min:max	minimum and maximum value determining range of input values, e.g. `20:200`; integer values between 0 and 255.
perc	percentage of input values to be ignored on both sides of the maps' histogram during linear stretch or histogram equalization; 0 ≤ real value < 50. When no ranges are specified for the MapColorComp24 or MapColorComp24Linear commands then all input values are used, i.e. no stretching. The MapHeckbert and the MapColorComp24HSI commands always use all input values.
NrColors	For Dynamic (Heckbert) color composites: the number of colors present in the output map (2-255).

All maps in the input map list must have the same georeference.

When the definition symbol `=` is used, a dependent output map is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output map has been calculated.

Examples
A complete expression to calculate a color composite of bands tm4, tmb3, and tmb2, using linear stretching and ignoring 1% of all input values, might thus read:

```
OUTMAP = MapColorComp(mlist(tmb4,tmb3,tmb2),1,1,1)
```

A complete expression to calculate a dynamic color composite of bands tm4, tmb3, and tmb2, using 200 colors, might thus read:

```
OUTMAP = MapHeckbert(mlist(tmb4,tmb3,tmb2),200)
```

Algorithm

Standard color composites

Linear stretching of interval
The specified interval range per band is linearly divided into 6 classes of equal length with numbers 0 to 5. Since this is done for three bands, the number of possible combinations is 6×6×6 = 216. This is the number of different colors that will appear in the color composite.

Histogram equalization
The specified interval range (lower and upper boundary) per band is divided into 6 classes numbered 0 to 5, each of which has an equal area under the histogram. The number of different output colors is 6×6×6 = 216, just as it is for linear stretching.

Output colors
Each output color obtains an internal number of the system picture domain `ColorCmp`; the value is calculated as:

```
output = 36×red + 6×green + blue.
```

This implies that the output map will contain internal values between 0 and 215 (since 215 = 5 × 36 + 5 × 6 + 5). The values of this `ColorCmp` domain are always linked to system representation `ColorCmp`.

Dynamic color composites
A dynamic color composite is calculated using the Heckbert Quantization Algorithm. The Heckbert algorithm produces a color composite on the basis of the amount of variation in pixel values in the three input maps.

This algorithm first builds a three dimensional histogram, indicating how 'popular' any given value is in the images. All values fall in one box or cube. This histogram is then subdivided into smaller boxes or cubes: a division is made in the middle of the axis, which has the largest variation. This process continues until as many boxes are created, as there are output colors (number defined by the user, maximum 255). This algorithm attempts to create boxes, which have approximately equal popularity in the image. Then, colors are assigned to represent each box.

1. First, the input map values at 1% and 99% are determined. See also the Histogram operation.
2. The band with the largest variation in pixel values is selected, the total number of pixels for this band is calculated, and the band is divided into 2 halves, each containing half of the total number of pixels. The division leaves the other 2 bands intact. The result after one division is 2 so-called 'boxes': one band divided over the 2 boxes, and the other 2 bands complete in both boxes.
3. The program then searches for the next (part of a) band with the largest variation in pixel values. The total number of pixels on that (part of the) band is calculated, the (part of a) band is divided into 2 halves, so that each new part of the band contains half of the total number of pixels, and the other 2 bands are left as they were. The result after 2 divisions is 3 'boxes': each with (parts of) the red, the green, and the blue band.
4. The division process is repeated until the total number of 'boxes' reaches the number of user-defined colors desired for the output map.
5. Then colors are assigned to all boxes. For each box, weighted averages are calculated of the parts of the red band, green band and blue band covered by that box; the outcome values are the Red, Green and Blue values for that box. The calculation is repeated for all boxes.

An example of the first and second division is given in Figures 1, 2 and 3 below.

Example Heckbert algorithm
Three 1% histograms are calculated before the first division (see Figure 1).

Figure 1 shows that the pixel values range:

- from 31 to 98 in the first input map
- from 22 to 100 in the second input map and
- from 21 to 66 in the third input map.

Thus, the largest variation is found in the second input map. The total number of pixels on this band is divided in 2 so-called 'boxes': the first division is at pixel value 65 of band 2.

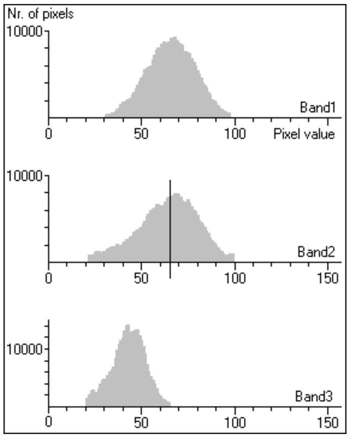


Figure 1: 1% histograms for Heckbert Color Composite

The result of the first division is represented in Figures 2 and 3. Figure 2 shows the histograms of one box; Figure 3 the histograms of the other. Enlarge the Help window by dragging its borders when you cannot see both figures next to each other.

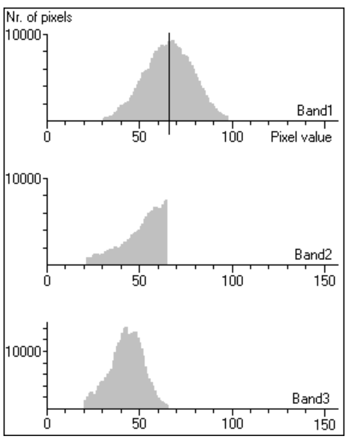


Figure 2:

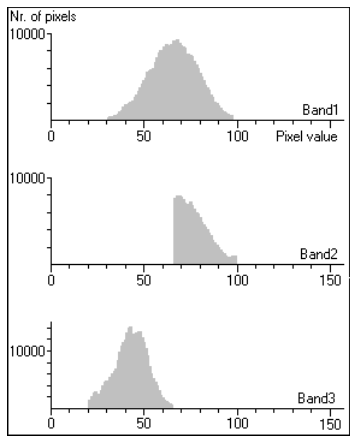


Figure 3:

Figure 2 shows that the pixel values range in box 1:

- from 31 to 98 in the first input map
- from 22 to 65 in the second input map, and
- from 21 to 66 in the third input map

Figure 3 shows that the pixel values range in box 2:

- from 31 to 98 in the first input map
- from 66 to 100 in the second input map, and
- from 21 to 66 in the third input map.

The next largest variation is found in the first input map in box 1. The total number of pixels on this band divided into 2 parts: the next division takes place at pixel value 66.

Output domain for dynamic composites
The operation creates a new domain (type Picture) for the output color composite and a new representation for this domain. This output domain and representation are always stored within the output map (internal domain and internal representation).

24-bit RGB color composites

Linear stretching of interval
Each input band is stretched to values between 0 and 255 using linear stretching using the user-defined intervals of the histograms.

Histogram equalization
Each input band is stretched to values between 0 and 255 using histogram equalization using the user-defined intervals of the histograms.

Output colors
The results are combined in a map with 4 bytes per pixel (4 bytes is 32 bits, thus 8 bits are not used). Each pixel in this map contains the red, green, and blue intensities of values between 0 and 255. It means that the possible number of output colors is $256 \times 256 \times 256 = \pm 16$ million.

24-bit HSI color composites

In this case no stretching is performed. In the output map, for each pixel the hue, saturation and intensity is converted to red, green, and blue intensities. The following relations exist:

$$\text{Hue} = 255/2\pi \times \text{arctan2}\ (\ ½\sqrt{3} \times (\text{Green-Blue})\ ,\ \text{Red} - (\text{Green+Blue}) / 2\ ) \times 240/255$$
$$\text{Saturation} = \sqrt{}\ (\text{Red}^2 + \text{Green}^2 + \text{Blue}^2 - \text{Red}\times\text{Green} - \text{Red}\times\text{Blue} - \text{Green}\times\text{Blue}) \times 240/255$$
$$\text{Intensity} = 1/3 \times (\text{Red} + \text{Green} + \text{Blue}) \times 240/255$$

Red, green, and blue values range from 0 to 255. Hue, saturation, and intensity values however range from 0 to 240; this range complies with the Windows color scheme definition. In the formulas above, multiplication factor 240/255 is used to obtain that range.

References:

- Heckbert, P., 1982. Color image quantization for frame buffer display. SIGGRAPH '82 Proceedings, p. 297.

7.3.4 Resample

Functionality

The Resample operation resamples a raster map from the map's current georeference to another target georeference. The coordinate of each output pixel is used to calculate a new value from close-by pixel values in the input map. Three resampling methods are available: nearest neighbour, bilinear interpolation, and bicubic interpolation.

In raster operations (e.g. MapCalc, Cross), all input raster maps must have the same georeference. Thus, prior to such operations, use Resample:

- *to combine raster maps from various sources*, when maps use different coordinate systems (projections) or different georeferences (pixel size): resample the maps to one common georeference;
- *to combine satellite imagery of different dates or resolutions*: create a georef tiepoints for each set of images, then resample the images preferably to a georef corners;
- *to combine satellite images with rasterized vector maps*: rasterize the vector data on the georef tiepoints of the satellite images, or, in case you prefer North-oriented raster maps, rasterize the vector maps with a georef corners, and resample the images with the georef tiepoints to this georef corners;
- *to combine scanned photographs with rasterized vector data*, or *to rectify scanned aerial photographs*: create a georef tiepoints, a georef direct linear or a georef orthophoto for the photo, then resample the photo to a georef corners.

You will usually resample raster maps from their current georeference to one common georeference corners.

☞ It is not advisable to use Resample to make a raster map with any georeference use a georeference 3D; use the Apply 3D operation instead.

For more information on georeference types or on creating georeferences, refer to ILWIS objects: georeferences or How to create a georeference. For more information on editing a georef tiepoints, a georef direct linear or a georef orthophoto, refer to Tiepoint editor.

Resampling methods

To resample an image, select an input image which has a georeference (usually a georeference tiepoints), select a resampling method (nearest neighbour, bilinear, bicubic), type an output map name and select the target georeference (usually a georeference corners).

- When using nearest neighbour resampling, the value of the input pixel closest to a new output pixel is used as output value (see Fig. 1);
- when using bilinear resampling, the values of 4 input pixels closest to a new output pixel are used to interpolate output values (see Fig. 2);
- when using bicubic resampling, the values of 16 input pixels closest to a new output pixel are used to interpolate output values.

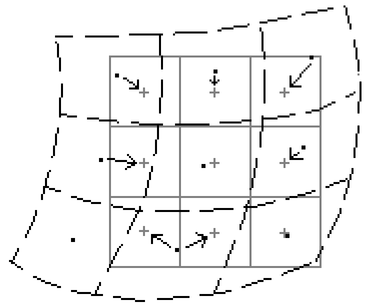


Figure 1: Nearest neighbour resampling. Dashed black lines represent Input pixels, black dots represent the coordinates of input pixels; the rectangular grid represents the output pixels, the plus marks indicate the coordinates of output pixels. The arrows indicate how output values are determined.

As you can see in Figure 1, some values of the input map may be used twice in the output map, while other input values may not be used at all.

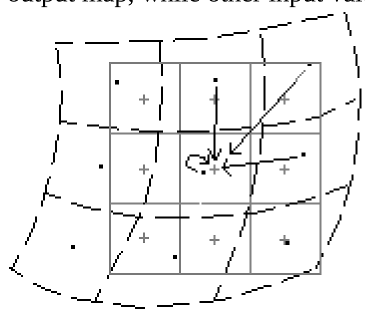


Figure 2: Bilinear resampling. Dashed black lines represent Input pixels, black dots represent the coordinates of input pixels; the rectangular grid represents the output pixels, the plus marks indicate the coordinates of output pixels. The arrows indicate how output values are determined.

Input map requirements
The input map needs to have a georeference; this is usually a georef tiepoints, a georef direct linear or a georef orthophoto.
For nearest neighbour resampling, the input map can have any domain.
For bilinear and bicubic resampling, the input map needs to be a value map.

Domain and georeference of output map
The output map uses the same domain as the input map. If the input map has a value domain, the value range and precision can be adjusted for the output map.

The target georeference for the output map has to be selected or created. You can usually select an existing georeference corners.

Dialog box

Dialog box options

Input raster map:	Select an input raster map. Open the list box and select the desired input map, or drag a raster map directly from the Catalog into this box. The input raster map may not use georef `None`.
Resampling method:	Select a resampling method: Nearest Neighbour, Bilinear, or Bicubic.
Output raster map:	Type a name for the output raster map that will contain resampled pixels.
Georeference:	Select an existing target georeference for the output raster map to which the input map should be resampled; open the list box by clicking it. Or create a new georeference by clicking the little create button.
Value range:	In case the output map uses a value domain: accept the default value range, or specify your own range of possible values in the output map.
Precision:	Accept the default precision of output values, or specify your own precision.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

Command line

The Resample operation can be directly executed by typing one of the following expressions on the command line of the Main window:

OUTMAP = `MapResample`(InputMapName, Georeference,
`NearestNeighbour` | `BiLinear` | `BiCubic`
[, `Patch` | `NoPatch`)]

where:

OUTMAP	is the name of your output raster map.
`MapResample`	is the command to start the Resample operation.
InputMapName	is the name of your input raster map.
Georeference	is the name of an existing target georeference that should be used for the output raster map.
`NearestNeighbour`	is the parameter for Nearest Neighbour resampling.

`BiLinear`	is the parameter for BiLinear resampling.
`BiCubic`	is the parameter for BiCubic resampling.
`Patch`	is an optional parameter to have the operation first patch the input raster map, then resample it and finally unpatch it. This is the default behaviour.
`NoPatch`	is an optional parameter to have the operation directly resample the input raster map without patching. This will take less disk space but usually will take more time.

When the definition symbol `=` is used, a dependent output map is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output map has been calculated.

Algorithm

The Resample operation resamples a raster map from the map's current georeference to another target georeference. The coordinate of each output pixel is used to calculate a new value from close-by pixel values in the input map. Three resampling methods are available: nearest neighbour, bilinear interpolation, and bicubic interpolation.

The resampling process consists of several steps:

- the selected output georeference determines the number of rows and columns in the output map; thus the XY-coordinate for each output pixel is known;
- next these positions are looked up in the original map and, according to the selected interpolation method, 1 (nearest neighbour), 4 (bilinear) or 16 (bicubic) neighbour pixels around this position in the input map are used to calculate a value for the output map.

Nearest neighbour resampling is the fastest method, but results in discontinuities because some input values may be used more than once as output value, while other input values may not be used at all. Bilinear resampling takes much less time than a bicubic resampling. A bilinear interpolation results in discontinuity of the first derivative. A bicubic interpolation remains continuous up to the second derivative.

Nearest neighbour resampling
With nearest neighbour resampling, first the coordinate of each pixel in the output map is determined. Then, for each output pixel, the pixel value of input pixel closest to this coordinate is used as output value.

Figure 1 below shows the position of a 'new' pixel in the output map, and the position and values of 4 surrounding pixels in the input map.

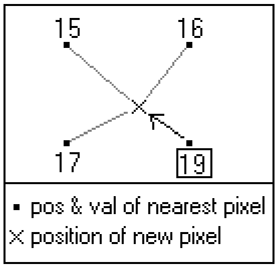


Figure 1: Nearest Neighbour resampling. Dots represent the coordinates of the input pixels; the cross indicates the coordinate of an output pixel. The arrow shows how the value of the nearest input pixel is assigned to the output pixel.

The value for the 'new' pixel in the output map is the value in the input map closest to the new coordinate (19).

Bilinear resampling
First the coordinate of each pixel in the output map is determined. Then the values of 4 surrounding pixels of the input map are used to calculate an interpolated value for each pixel in the output map.

Figure 2 below shows the position of a 'new' pixel in the output map, and the position and values of 4 surrounding pixels in the input map.

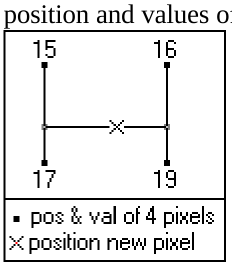


Figure 2: Bilinear interpolation in Bilinear Resampling. The numbered dots represent the coordinates of 4 neighbouring input pixels. The cross represents the coordinate of an output pixel. The straight lines represent the interpolations; intermediate answers are indicated by the unnumbered dots.

The value of the 'new' pixel in the output map is calculated by:

- first 2 interpolations in y-direction (between values 15 and 17, and between values 16 and 19) resulting in two intermediate values which are unnumbered
- then 1 interpolation in x-direction (between the two intermediate values).

A straight line is drawn through each set of 2 points, and from this the value of the third point is known. A bilinear interpolation should not be used when you intend to calculate a derivative of the output map.

Bicubic resampling
With bicubic resampling, first the coordinate of each pixel in the output map is determined; then the values of 16 surrounding pixels of the input map are used to calculate an interpolated value for each pixel in the output map.

Figure 3 below shows the position of a 'new' pixel in the output map, and the position and values of 16 surrounding pixels in the input map.

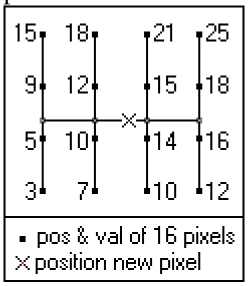


Figure 3: Bicubic interpolation in Bicubic resampling. The numbered dots represent the coordinates of 16 neighbouring input pixels. The cross represents the coordinate of an output pixel. The straight black lines represent the interpolations; intermediate answers are represented by unnumbered dots.

The value of the 'new' pixel in the output map is calculated by:

- first 4 interpolations in y-direction,
- then 1 interpolation in x-direction (between the 4 intermediate values).

A third order polynomial is fitted through each set of 4 known points and from this the value of the fifth point is known. A bicubic interpolation gives a better estimate of the output value than a bilinear interpolation.

7.4 Statistics

7.4.1 Optimum Index Factor

Functionality / Algorithm

The Optimum Index Factor (OIF) is a statistic value that can be used to select the optimum combination of three bands in a satellite image with which you want to create a color composite. The optimum combination of bands out of all possible 3-band combinations is the one with the highest amount of 'information' (= highest sum of standard deviations), with the least amount of duplication (lowest correlation among band pairs).

Preparation

- Create a map list containing the multi-spectral bands of your satellite image.
- Calculate a variance-covariance matrix or a correlation matrix of this map list.
- Open the Properties dialog box of your map list and click the Additional Info button.

The ranked OIF values are shown with the corresponding band combinations. For more information, refer to How to calculate Optimum Index Factor.

Example

Consider an input map list containing 7 bands, named `tmb1`, `tmb2`, `...` `tmb7`. For each combination of three bands in the map list, OIF values are calculated through a simple formula which uses the standard deviations of the bands and correlation coefficients between band pairs (see algorithm below). The OIF values may read:

```
OIF Index Highest Ranking
1:  tmb4    tmb5    tmb6     ( 29.04)
2:  tmb1    tmb5    tmb6     ( 28.58)
3:  tmb3    tmb5    tmb6     ( 27.98)
4:  tmb5    tmb6    tmb7     ( 26.67)
5:  tmb1    tmb4    tmb5     ( 26.42)
6:  tmb2    tmb5    tmb6     ( 26.01)
```

The OIF values suggest that from the 7 bands in the map list, the combination of bands `tmb4`, `tmb5` and `tmb6` is the best statistical choice to create a color composite.

Notes

- By using the three bands with the highest OIF value for a color composite, it is not implied that you will create the 'best' color composite since this greatly depends on the purpose of your work.
- 'Noise' (such as dropouts) in one of the input bands is considered as high variance, hence this band will appear in all high ranking combinations.

Input requirements

To calculate Optimum Index Factors, a map list is required which contains at least 3 raster maps; the raster maps must all use the `Image` domain or the same value domain, and they must have the same georeference. Furthermore, it is necessary that a correlation matrix or a variance-covariance matrix has been calculated for the map list; this will provide the standard deviations and correlation coefficients, which are required for the OIF calculation.
For more information, refer to How to calculate Optimum Index Factor.

Output OIF values

After a variance-covariance matrix or a correlation matrix has been calculated for the input map list, you can display the ranked OIF values and corresponding band combinations by clicking the Additional Info button in the Properties dialog box of the input map list.
The OIF values are stored in the object definition file of the map list (.MPL).

Note: It is not possible to calculate OIF values from the command line.

Algorithm

1. First the number of possible combinations of three bands within the map list is determined as:

$$\binom{N}{3} = \frac{N!}{\left(3!\left(N-3\right)!\right)}$$

where:
N is the total number of bands in the map list.

For 3 bands, there is only 1 combination;
for 4 bands, there are 4 combinations;
for 5 bands, there are 10 combinations;
for 6 bands there are 20 combinations; and
for 7 bands, there are 35 combinations.

2. Then, for each combination of three bands, the OIF is calculated as:

$$OIF = \frac{Std_i + Std_j + Std_k}{\left|Corr_{i,j}\right| + \left|Corr_{i,k}\right| + \left|Corr_{j,k}\right|}$$

where:
Std_i standard deviation of band i
Std_j standard deviation of band j
Std_k standard deviation of band k
$Corr_{ij}$ correlation coefficient of band i and band j
$Corr_{ik}$ correlation coefficient of band i and band k
$Corr_{jk}$ correlation coefficient of band j and band k

3. Finally, the OIF values are ranked.

7.4.2 Point statistics

Point statistics may help to get an impression of the nature of your point data prior to for instance a point interpolation, and to find necessary input parameters for kriging.

For point statistics a point map is required in which:
- the points themselves are values (point map with a value domain), for instance concentration values, or
- the points have an identifier (point map with a Class or ID domain) and values are stored in a column of the attribute table that is linked to the map.

Point statistics creates tables. By displaying the output tables, you can also display graphs.

Spatial correlation: calculates **spatial autocorrelation** (as Moran's I), **spatial variance** (as Geary's c) and semi-variance for point values that are at certain distances towards each other in a point map.
Regarding spatial autocorrelation and spatial variance: the user is encouraged to compare his or her data set with a second data set of the same point locations, but with a set of randomly generated attribute values, approximately in the same range as the measured variable. (Refer to the RND functions in Table Calculation). If the correlation/variance graphs are very much the same for the measured data and the random data, no autocorrelation exists between the data points. Hence, point interpolation is not useful.

By calculating semi-variances, you can display a semi-variogram. By modelling the semi-variogram, you can find the necessary input parameters, such as a model (spherical, exponential, etc.) and sill, range, and nugget values, for a kriging operation.

The Pattern analysis operation is a tool to obtain information on the spatial distribution of points in a point map. The output table contains six columns with the probabilities of finding 1 point (`Prob1Pnt`) within a certain distance from any point in your input map, then 2 points (`Prob2Pnt`), 3 points (`Prob3Pnt`), etc. Another column (
`ProbAllPnt`) contains the sum of `Prob1Pnt`, `Prob2Pnt`, … `Prob(n-1)`, in which n is the number of points in the input map.

By inspecting the graphs of distances against probabilities, you may recognize distribution patterns of your points like random, clustered, regular, paired etc.

7.4.3 Spatial correlation

Functionality

Spatial autocorrelation measures dependence among nearby values in a spatial distribution. Variables may be correlated because they are affected by similar processes, or phenomena, that extend over a larger region. Odland (1988, p.7) mentions that spatial autocorrelation 'exists whenever a variable exhibits a regular pattern over space in which values at a certain set of locations depend on values of the same variable at other locations'.

For example, if the concentration of a certain pollutant is very high at a certain location, it will most likely also be high in the direct surroundings. In other words, the concentration is autocorrelated at small distances. At larger distances, it is less likely that the concentration will be equally high. The correlation will probably be lower, and the variance higher.

By plotting the answers on autocorrelation against the distance classes, you will be able to see until which distance spatial autocorrelation exists between point pairs. This value can be used for the limiting distance in point interpolations such as moving average and moving surface. Furthermore, the user is encouraged to compare his or her data set with a data set consisting of the same point locations, with a set of attribute values, approximately in the same range as the measured variable, but created at random (using one of the RND functions in Table Calculation). If the graphs are very much the same for the measured data and the random data, no spatial autocorrelation exists between the data points. Hence, point interpolation is not useful.

Calculation of the semi-variance is a basic geostatistical measure to determine the rate of change of a regionalized variable along a specific orientation (usually distances). Semi-variance is defined as the sum of the squared differences between pairs of points separated by a certain distance divided by two times the number of points in this distance class. In a graph, semi-variance results can be plotted against distances; this is known as a semi-variogram. By modeling the semi-variogram, you can obtain necessary input information (such as model type, sill, range, and nugget) for a Kriging operation later on. For more information, see the Additional information on semi-variograms below.

General process of this operation

1. First, the distances between all points are calculated.
2. Then, distance classes are determined. This is usually done according to the user-specified lag spacing: in the output table, records will appear for each multiple of the user-specified lag spacing, thus when using a lag spacing of 500 m., the distance values in the output table will be 0, 500, 1000, 1500, etc. However, these distance values in the output table represent the middle value of a distance class, thus for lag spacing 500, distance 500 represents the distance interval of 250-750m, distance 1000 represents the distance interval of 750-1250m, etc.
   When a variable was sampled at regular distances, you can use this distance for the lag spacing.
   On the command line, you can also use a certain expression to obtain log-scaled distance classes.
3. Subsequently, for each distance class, the number of point pairs is counted of which the points have such a distance towards each other.
   Thus, when the user-specified lag spacing is 500 m.:
   - the first record in the output table has value 0 in column Distance; this first distance class is only half a distance class: it contains all point pairs of which the distance of the points towards each other is 0-250 m.;
   - the second record in the output table has value 500 in column Distance; this distance class contains all point pairs of which the distance of the points towards each other is between 250-750 m.;
   - the third record in the output table has value 1000 in column Distance; this distance class contains all point pairs of which the distance of the points towards each other is between 750-1250 m. etc.;
4. Then, for all the point pairs within a certain distance class, the following statistical values are calculated:
   - spatial autocorrelation (as Moran's I)
   - spatial variance (as Geary's c)
   - semi-variance.

The formula to calculate semi-variance reads:

$\gamma = \sum (z_i - z_{i+h})^2 / 2n$

Where:

γ	semi-variance of points that have a certain distance (h) towards each other
z_i	the value of point i
z_{i+h}	the value of a point at distance h from point i
$\sum(z_i - z_{i+h})^2$	the sum of the squared differences between point values of all point pairs that have distance h towards each other
n	the number of point pairs within a distance class

Methods
In the dialog box, you can choose to use either the omnidirectional or the bidirectional method:

- The omnidirectional method simply determines all distances between all point pairs, regardless of any direction, i.e. in all directions. Thus, all point pairs that have a certain distance towards each other will be counted in a certain distance class. Then, Moran's I, Geary's c, and the semi-variance are calculated for all point pairs within each distance class.
- The bidirectional method first counts, just like the omnidirectional method, all pairs of points that have a certain distance to each other, and then calculates the Moran's I and Geary's c for these point pairs within each distance class. Furthermore, all point pairs are counted with a certain distance to each other and with a certain direction towards each other. For these point pairs, semi-variance will be calculated. Then, also, for the direction perpendicular to the specified direction, point pairs are counted and semi-variances calculated.

Both for the omnidirectional or the bidirectional method, linear distance intervals are created where the upper limits of these distance classes are multiples of the user-specified lag spacing.

To calculate semi-variances in a certain direction, you thus have to use the bidirectional method. You will have to supply two parameters: a direction angle and a tolerance angle. When you use a direction angle of 90°, it means that only point pairs for which the points are located in West-East or in East-West direction will be considered (i.e. +90° clockwise from the Y-axis). When using a tolerance of 10°, the direction of every 2 points may differ -10° or +10° from the specified direction (90°). So, in fact, all points that are found in a position within 80° to 100° to one another are valid pairs. Then, for the valid point pairs, the distance class to which they belong will be determined.

Optionally, you can specify a third parameter, the band width (m), to limit the tolerance angle to a certain width.
The parameters for the bidirectional method are schematically presented in Figure 1.

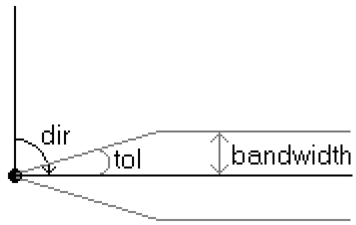


Figure 1: Schematic explanation of the parameters for the bidirectional method with which semi-variances will be calculated for the specified direction as well as for the perpendicular direction. The user has to specify a direction (Dir, the black angle) and a tolerance (Tol, the gray angle), and optionally, also a band width (the gray distance in meters) can be specified. These parameters are used to find valid point pairs. When an input point is located at the origin of this picture, it is calculated if any other input point is within the specified direction, tolerance angle and bandwidth. If this is the case, the 2 points are a valid point pair; otherwise the pair is ignored. For each valid point pair, the distance between the 2 points is calculated, and the point pair is counted in the appropriate distance class.

Finally, from the command line, you can even use another method by which logarithmic distance intervals are used. The lag spacing increases with the distance.

Input map requirements
The input point map should either be a value map itself, or a Class or ID point map which has a linked attribute table with one or more value columns.

Output table
An output table with domain None is created.

When you use the option Omnidirectional, the output table will contain 5 columns:

- Column `Distance` lists the middle values of the distance intervals;
- Column `NrPairs` lists for each distance interval, the number of point pairs found at these distances towards each other;
- Column `I` lists for each distance interval, the spatial autocorrelation of the point pairs in this distance interval;
- Column `c` lists for each distance interval, a statistic for spatial variance of the point pairs in this distance interval;
- Column `Semivar` lists for each distance interval, the semi-variance of the point pairs in this distance interval.

When you use the option Bidirectional, the output table will contain 8 columns:

- Column `Distance` lists the middle values of the distance intervals;
- Column `NrPairs` lists for each distance interval, the number of point pairs found at these distances towards each other;
- Column `I` lists for each distance interval, the spatial autocorrelation of the point pairs in this distance interval;
- Column `c` lists for each distance interval, a statistic for spatial variance of the point pairs in this distance interval;
- Column `NrPairs1` lists for each distance interval, the number of point pairs found in the user-specified direction and at these distances towards each other;
- Column `Semivar1` lists for each distance interval, the semi-variance for the point pairs found in the user-specified direction and at these distances towards each other;
- Column `NrPairs2` lists for each distance interval, the number of point pairs found perpendicular to the user-specified direction and at these distances towards each other;
- Column `Semivar2` lists for each distance interval, the semi-variance for the point pairs found perpendicular to the user-specified direction and at these distances towards each other.

Additional information: Semi-variograms

From the results of this operation, you can make a semi-variogram. A semi-variogram model describes the expected difference in value between pairs of samples with a given relative orientation.

- Display the output table of the Spatial correlation operation in a table window.
- Inspect the output table:
  - it is advised that the distance classes contain at least 30 point pairs, otherwise the calculated values will not be very reliable;
  - usually not more than half of the total sampled distance should be taken into account; the larger the distance between the point pairs, the less point pairs, the less reliable the outcome;
  - determine the variance (σ^2) of your variable in the input table, or in a histogram of the input map. The variance of a column can be calculated by using an expression like `OUT = var(`*`columnname`*`)`
- From the Options menu in the Table window, choose the Show Graph command.
- In the Graph dialog box, choose for the X-axis: the `Distance` column; and choose for the Y-axis: the `SemiVar` column.
  In case you used the bidirectional method, you can draw two graphs:
  - one with column `SemiVar1` as the Y-axis (this is the semi-variance in the direction which you specified) and
  - one with column `SemiVar2` as the Y-axis (this is the semi-variance in a direction perpendicular to the direction which you specified).
- In the Edit Graph dialog box, choose Points to represent the semi-variance values as points.

In literature, the shown graph is called a discrete experimental semi-variogram.

Ideally, a semi-variogram has the shape of Figure 2:
- When the distances between sample points is 0, the differences between sampled values is also expected to be 0. Thus, the semi-variance at distance 0 is 0, and when a line plotted through the points, this line will pass through the origin of the graph.
- Samples that are at a very small distance to each other are expected to have almost the same values; thus, the squared differences between sample values are expected to be small positive values at small distances.
- With increasing distance between point pairs, the expected squared differences between point values will also increase.
- At some distance the points that are compared are so far apart that they are not any more related to each other, i.e. the sample values will become independent of one another. Then, the squared differences of the point values will become equal in magnitude to the variance of the variable. The semi-variance no longer increases and the semi-variogram develops a flat region, called the sill. The distance at which the semi-variance approaches the variance is referred to as the range or the span of the variable.

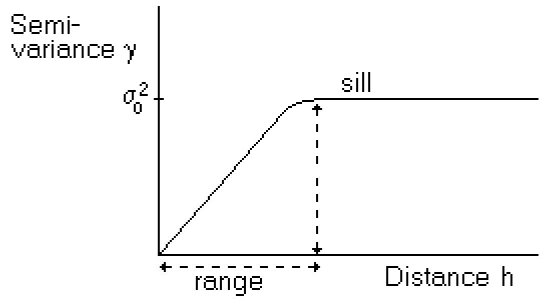


Figure 2: Ideal shape of a semi-variogram.

Remarks on semi-variograms:

- A semi-variogram with a nugget effect is a semi-variogram, which does not go through the origin. The variable is so erratic over very short distances that the semi-variance goes from zero to the level of the nugget effect in a distance less than the sampling distance: i.e. the variable is highly variable over distances less than the specified lag spacing or the sampling interval.
- For semi-variograms, which, after a flat sill level, show an ongoing increase in semi-variance values, probably a trend has to be taken into account for the longer distances. However, the semi-variance values for the distances up to the sill, are probably accurate enough to be used in a model.
- Possible dips in the semi-variogram indicate that at certain distances between points there is less difference between the samples than at other distances; this might indicate periodic trends.

The next step, before Kriging, is to model the discrete values of your experimental semi-variogram by a continuous function, which will give an expected value for any desired distance.

- From the Graph menu, choose the Add Semi-variogram Model command.
- In the Add Graph Semi-variogram dialog box, you can choose a type of semi-variogram model (spherical, exponential, etc.), and you can fill out values for the sill, range and nugget. Next a line will be drawn through your semi-variance values according to the model you selected and the values you selected for sill, range and nugget. You are advised to visually experiment a little with models and sill, range, and nugget values to find the best line through the experimental semi-variance values.

For more information on drawing lines through a semi-variogram, refer to the Graph window: Add Semi-variogram Model dialog box. Once, you have decided which model, and which values for sill, range and nugget fit your data best, you can continue with the Kriging operation.

References

- Clark, I. 1979. Practical geostatistics. Applied Science Publishers, London. 129 pp.
- Davis, J. C. 1973. Statistics and data analysis in geology. Wiley, New York. 646 pp.

- Isaaks, E. H., and R. M. Srivastava. 1989. An introduction to applied geostatistics. Oxford University Press, New York. 561 pp.
- Odland, J. 1988. Spatial autocorrelation. In: G.I. Thrall (Ed.), Sage University Scientific Geography Series no. 9. Sage Publications, Beverly Hills. 87 pp.

Dialog box

Dialog box options

Input point map: Select an input point map. Open the list box and select the desired input map, or drag a point map directly from the Catalog into this box. You can select a point map with a value domain, or a point map with a class or ID domain, which has a linked attribute table with values.

Column: In case you selected an input point map with a class or ID domain, select an attribute column (value domain) from the attribute table that is linked to the map.

Omnidirectional: Choose the option omnidirectional when distances between points should be calculated in all directions. Depending on the user-specified lag spacing, distance classes are determined; the upper limits of the distance classes are multiples of the lag spacing. Based on the found distance between each pair of points, it is determined to which distance class this pair of points belongs. Thus, all point pairs that are found to have a certain distance towards each other, will be counted as a point pair in that distance class. For all point pairs within each distance class, the spatial autocorrelation, spatial variance and semi-variance will be calculated.

Bidirectional: Choose the option bidirectional when distances between points should also be calculated in a certain direction. Like the Omnidirectional option, all points that have a certain distance to each other will be counted as a point pair in a certain distance class. For all point pairs within each distance class, the spatial autocorrelation and spatial variance will be calculated.
Furthermore, for the same distance classes, all points that have a certain distance to each other and that fall within the specified direction, will be counted as a point pair. For all these point pairs, the semi-variance will be calculated. Similarly, point pairs will be counted in the perpendicular direction and semi-variances will also be calculated for this perpendicular direction.

Lag spacing (m): Type a value that will be used for the distance classes; the output table will contain a Distance column with multiples of the specified lag spacing. However, these values represent the middle value of a distance class. For example, when using lag spacing 500, distance classes of 500 m will be used, and the output table will show Distance values of 0, 500, 1000, 1500,

etc. These values represent the intervals 0-250, 250-750, 750-1250, 1250-1750, etc.
- For distance class 0, all point pairs will be counted where the points are less than 250 m apart;
- for distance class 500, all point pairs will be counted where the points are between 250 and 750 m apart;
- for distance class 1000, all point pairs will be counted where the points are between 750 and 1250 m apart, etc.

When a variable was sampled at regular distances, you can enter this distance as the lag spacing. You should not use a value larger than half the size of your map.

Direction (deg): For Bidirectional: type a value for the direction in which point pairs should be found. The direction is specified in degrees as a clockwise angle from the Y-axis.
Direction 0° means finding only point pairs in the direction North-South and South-North to each other.
Direction 90° means finding only point pairs in the direction East-West and West-East to each other.
0° ≤ direction ≤ 90°.

Tolerance (deg): For Bidirectional: type a value in degrees for half of the opening angle with which point pairs should be found in a certain direction. Tolerance 0.01 means a very narrow opening to find point pairs in a certain direction.
Tolerance 45° means an opening of 90°.
≤ 45°.

Band width (m): For Bidirectional: optionally, specify a value for the bandwidth. By specifying a bandwidth, you will limit the opening angle to a certain width; the opening width with which point pairs can be found in a certain direction will never be broader than twice the specified bandwidth.

Output table: Type a name for the output table that will contain the spatial autocorrelation, spatial variance, and semi-variance.

Show: Select this check box if you want the output table to be displayed in a table window when the operation has finished. Clear this check box if you do not want to see this table immediately: you simply define how the output table should be created.

Description: Optionally, type a description for the output table. The description appears in the title bar when the table is displayed.

A dependent output table is created.

Command line

The Spatial correlation operation be directly executed by typing one of the following expressions on the command line of the Main window:

OUTTABLE = `TableSpatCorr`(InputPointMap)

OUTTABLE = `TableSpatCorr`(InputPointMap, *LagSpacing*)
OUTTABLE = `TableSpatCorr`(InputPointMap, *LagSpacing*, *Direction* [, *Tolerance* [, *Bandwidth*]])

where:

OUTTABLE — is the name of your output spatial correlation table.

`TableSpatCorr` — is the command to start the Spatial correlation operation.

InputPointMap — is the name of your input point map (value map). When you want to use a Class or ID point map with an attribute table linked to the map, you can use: *InputPointMap.ColumnName*

LagSpacing — is a parameter to specify the length in meters of the linear distance intervals that should be used. When this parameter is not used, logarithmic distance intervals will be calculated.

Direction — is a parameter to specify an angle in degrees for the direction in which point pairs should be found. Direction 0° is North; direction 90° is East. The direction is a clockwise angle from ≤ direction ≤ 90°. By using a direction, you will obtain bidirectional semi-variance results. When this parameter is not used your semi-variance results will be omnidirectional.

Tolerance — is a parameter to specify half of the opening angle in degrees with which point pairs in a certain direction should be found. 0° < tolerance ≤ 45°. When a Direction is specified but the Tolerance is not specified, a Tolerance of 45° will be used (meaning that all points will be found).

Bandwidth — is a parameter to specify half of the maximum width in meters within which point pairs within a certain angle and in a certain direction should be found. By specifying a bandwidth, you will limit the opening angle to a certain width; the opening width with which point pairs can be found in a certain direction will never be broader than twice the specified bandwidth. When this parameter is not specified, there is no limitation for the width of the opening angle.

When the first formula is used, only spatial autocorrelation and spatial variance are calculated, i.e. no semi-variances. Furthermore, instead of linear distance classes, you will obtain a number of logarithmic distance classes.

When the definition symbol `=` is used, a dependent output table is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output table has been calculated.

Algorithm

First, distances between all points are calculated. Distance classes are created for point pairs that are more or less at the same distance to each other. Distance classes are usually based on a user-specified lag spacing.

Then, for all point pairs within a distance group, the spatial autocorrelation, spatial variance and semi-variance is calculated.

In ILWIS, spatial autocorrelation between points is calculated as Moran's I (Odland):

$$I = \frac{n}{\sum\sum w_{ij}} \frac{\sum\sum w_{ij}(z_i - \bar{z})(z_j - \bar{z})}{\sum(z_i - \bar{z})^2}$$

In ILWIS, the spatial variance is calculated as Geary's c (Odland):

$$c = \frac{n-1}{2 \cdot \sum\sum w_{ij}} \frac{\sum\sum w_{ij}(z_i - z_j)^2}{\sum(z_i - \bar{z})^2}$$

Semi-variance is defined as:

$$g = \frac{\sum\sum w_{ij}(z_i - z_j)^2}{2 \cdot \sum\sum w_{ij}}$$

where:

z	the value of a point
$\bar{z}$	the average value of all available point values
$(z_i - \bar{z})(z_j - \bar{z})$	the product of: the difference of the value of point i and the average value of all points, and the difference of the value of point j and the average value of all points
$(z_i - z_j)^2$	the squared difference of the values of points `i` and `j`; this is calculated for all point pairs within a distance class and summed.
$(z_i - \bar{z})^2$	the squared difference of the value of point i and the average value of all points; this is calculated for all points and then summed; this is a constant value (variance).
n	the total number of points
w_{ij}	weight of a point pair. When using the omnidirectional method, $w_{ij} = 1$ when a point pair belongs to a certain distance class, otherwise $w_{ij} = 0$. When using the bidirectional method, $w_{ij} = 1$ when a point pair belongs to a certain distance class and when within the direction, tolerance and bandwidth as specified by the user (see also Figure 1 in Spatial correlation: functionality); otherwise $w_{ij} = 0$. In the numerators (top of a fraction) of these formulas, the weights assure that only the values of points that have a certain distance towards each other will be taken into account in the calculations for that distance class. In the denominators of these formulas (bottom of a fraction), i.e. in the standardization parts of the formulas, the weights count the number of valid point pairs within a distance class.

All summations are from `i=1` to n and from `j=i+1` to `n`, thus every point pair is counted only once.

References
- Geary, R.C., 1954. The contiguity ratio and statistical mapping.
- Moran, P.A.P., 1948. The interpretation of statistical maps.
- Odland, J. 1988. Spatial autocorrelation. In: G.I. Thrall (Ed.), Sage University Scientific Geography Series no. 9. Sage Publications, Beverly Hills. 87 pp.

7.5 Interpolation

7.5.1 Point interpolation

A point interpolation performs an interpolation on randomly distributed point values and returns regularly distributed point values. This is also known as gridding. In ILWIS, the output values are raster values.

In an ILWIS point interpolation, the input map is a point map in which:

- the points themselves are values (point map with a value domain), for instance concentration values, or
- the points are identifiers (point map with an Identifier domain) and values are stored in a column of an attribute table linked to the point map.

The output of a point interpolation is a raster map. For each pixel in the output map, a value is calculated by an interpolation on input point values.

Several point interpolation methods are available:

- Nearest point: assigns to pixels the value, identifier or class name of the nearest point, according to Euclidean distance. This method is also called Nearest Neighbour or Thiessen. The points in the input point map for the Nearest point operation do not need to be values necessarily; point maps (or attribute columns) with a class, ID or bool domain are also accepted.
- Moving average: assigns to pixels weighted averaged point values. The weight factors for the points are calculated by a user-specified weight function. Weights may for instance approximately equal the inverse distance to an output pixel. The weight function ensures that points close to an output pixel obtain larger weights than points, which are farther away. Furthermore, the weight functions are implemented in such a way that points which are farther away from an output pixel than a user-defined limiting distance obtain weight zero; this speeds up the calculation and prevents artifacts.
- Trend surface: calculates pixel values by fitting one surface through all point values in the map. The surface may be of the first order up to the sixth order. A trend surface may give a general impression of the data. Surface fitting is performed by a least squares fit. It might be a good idea to subtract the outcome of a trend surface from the original data, and calculate the residuals.
- Moving surface: calculates a pixel value by fitting a surface for each output pixel through weighted point values. The weight factors for the points are calculated

by a user-specified weight function. Weights may for instance approximately equal the inverse distance to an output pixel. The weight function ensures that points close to an output pixel obtain larger weights than points, which are farther away. Furthermore, the weight functions are implemented in such a way that points which are farther away from an output pixel than a user-defined limiting distance obtain weight zero; this speeds up the calculation and prevents artifacts. Surface fitting is performed by a least squares fit.

- Kriging: assigns to pixels weighted averaged points values, like the Moving Average operation. The weight factors in Kriging are determined by using a user-specified semi-variogram model (based on the output of the Spatial correlation operation), the distribution of input points, and are calculated in such a way that they minimize the estimation error in each output pixel. Two methods are available: Simple Kriging and Ordinary Kriging. Optionally, an error map can be obtained which contains the standard errors of the estimates. The errors are assumed to have e normal (Gauss) distribution. The technique is derived from the theory of regionalized variables (Krige, Matheron).

Preparations

- Point interpolations assume spatial randomness of the input points. To investigate whether your points are *randomly distributed*, or appear clustered, regular, or paired, etc., you can use the Pattern analysis operation prior to a point interpolation.
  In case your points are regularly distributed, e.g. as a regular grid, it is advised to directly rasterize the points with the Points to Raster operation. Use a georeference, which ensures that each output pixel contains one point and that the points are positioned at the center of the pixels. Further interpolation on the raster map values can be performed using the Densify operation or the Resample operation (bilinear or bicubic interpolation).
- Furthermore, point interpolations assume a certain degree of spatial correlation between the input point values. To investigate whether your point values are spatially correlated and until which distance from any point this correlation occurs, you can use the Spatial correlation operation prior to a point interpolation. The distance over which the data are correlated can be used as the maximum limiting distance in Moving average, Moving surface or Kriging. When there is no correlation between input point values, interpolation is senseless.

☞ When your point map contains values in which all extremes of your measured variable are present (e.g. for height values all mountaintops and valleys are measured), then using a Moving average point interpolation will probably be sufficient.

☞ It is possible that your point map does not contain all extreme values of a measured variable (e.g. you have soil samples and you have measured pH values as part of a large soil survey). In that case the advice is to use the Moving surface operation; the Moving average operation is not suitable. When you find extremes with the Moving surface operation, you might decide to go back to the field and measure the variable at the position of the extreme value; this will improve the results of a subsequent Moving surface operation.

- ☞ When you have relatively few points, it is advised to use a Trend surface operation.
- ☞ When you have rainfall data from a number of rainfall stations, first subtract for known patterns (e.g. height influence), then perform a Trend surface operation, and finally add the known patterns again to the output map.
- ☞ To check whether you have enough points within the limiting distance in a Moving average or a Moving surface point interpolation, you can perform the calculation again with a limiting distance increased by a factor 2. When you find profound differences in outcomes, you have chosen the limiting distance too small in the first calculation.
- ☞ When using a Moving average or a Moving surface point interpolation, it is for time efficiency reasons strongly advised to choose a rather large pixel size for the output map. Further interpolation on the raster map values can be performed using the Densify operation or the Resample operation (bilinear or bicubic interpolation).
- ☞ Instead of using Densify or Resample, you can also use online interpolation. In the Properties dialog box of the output raster map of an interpolation, i.e. of a value raster map, you can select the Interpolation check box. This means that the normal pixel value will only refer to the center of that pixel; elsewhere in any pixel, a value will be directly interpolated based on the values of 4 (bilinear) or 16 (bicubic) neighbouring pixels. The interpolated values are directly available in the raster map, e.g. by using left mouse information or in pixel information. Hence, the creation of an extra raster map with Densify or Resample is not needed. For more information, see Raster Map Properties (dialog box).

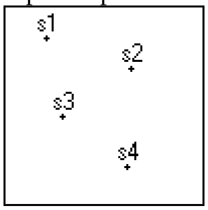
7.5.2 Nearest point

Functionality / Algorithm

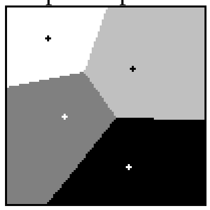
The Nearest point operation requires a point map as input and returns a raster map as output. Each pixel in the output map is assigned the class name, identifier, or value of the nearest point.

Example
Points may represent schools, hospitals, water wells, etc. The output of a nearest point operation on such a point map gives the 'service area' of the schools, hospitals or water wells, based on the shortest distance (as the crow flies) between points and pixels.

Input map:



Output map:



Nearest point operation versus Distance operation
The Nearest point is operation is also known as Nearest neighbour or Thiessen Map. Also the Distance operation has an option to create a Thiessen map.

- When every pixel in the output map is equally accessible, the Nearest point operation offers a quick way to obtain a Thiessen map from point data. Furthermore, as the Nearest point operation uses *Euclidean* distances, the output of the Nearest point operation may be somewhat more precise than the output of the Distance operation, which uses approximated raster distances.
- When you have many points or when you wish to use weights to indicate accessibilities, you can use the Distance operation. For more information, see Distance calculation: Thiessen map.

Input map requirements
No special input map requirements. Input maps may be maps of domain type class, ID, value, or bool. Furthermore, you can use a class or ID map with an attribute table.

Domain and georeference of output map
The output raster map uses the same domain as the input point map or the domain of the attribute column.
The georeference for the output map has to be selected or created; you can usually select an existing georeference corners.

Algorithm
For each output pixel, the Euclidean distances towards all points are determined. The value of the point with the shortest distance towards an output pixel is assigned to this output pixel.

Dialog box

Dialog box options

Input point map:	Select an input point map. Open the list box and select the desired input map, or drag a point map directly from the Catalog into this box. Any map with a class, ID, value or bool domain is accepted.
Attribute:	Select this check box if you want to use an attribute column from the attribute table, which is linked to the input point map (class or ID map). Clear this check box to use the input point map.
Output raster map:	Type a name for the output raster map that will contain, for

	each pixel, the class name, identifier, or value of the nearest point.
Georeference:	Select the name of an existing georeference or create a new georeference (click the create button).
Value range:	In case the output map uses a value domain, accept the default value range, or specify your own range of possible values in the output map.
Precision:	In case the output map uses a value domain, accept the default precision of output values, or specify your own precision.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

7.5.3 Moving average

Functionality

The Moving average operation is a point interpolation, which requires a point map as input and returns a raster map as output. The values for the output pixels are the weighted averages of input point values. Weighted averaging is the calculation of the sum of the products of weights and point values, divided by the sum of weights.

The weight factors for the input points are calculated by a user-specified weight function. There are two methods: inverse distance and linear decrease. Both methods ensure that points close to an output pixel obtain large weights and that points farther away from an output pixel obtain small weights. Values of points close to an output pixel are thus of greater importance to this output pixel value than the values of points farther away.

By specifying a limiting distance, you can influence until what distance from any output pixel, points will be taken into account for the calculation of a new value for that output pixel. For each output pixel, only the values of the points falling within the limiting distance to this output pixel will be used. Values of points that are farther away from an output pixel than the specified limiting distance, obtain weight zero by the weight calculation, and these values will thus not be used in the output pixel value calculation. This speeds up the calculation and prevents artifacts.

Input map requirements
The input point map should be a value map. Furthermore, when a point map uses a class or ID domain and the map is linked to an attribute table, you can also use such a point map and select a column with a value domain from the map's attribute table.

Domain and georeference of output map
The output raster map uses the same value domain as the input point map or the attribute column. The value range and precision can be adjusted for the output map. The georeference for the output map has to be selected or created; you can usually select an existing georeference corners.

- ☞ For time efficiency reasons, it is advisable to choose a rather large pixel size for the output raster map. Further interpolation on the raster values can be performed with the Densify operation or the Resample operation (using bilinear or bicubic interpolation).
- ☞ Prior to interpolation, you can use the Pattern analysis operation to investigate whether your points are randomly distributed, and the Spatial correlation operation to investigate whether your points are spatially correlated and until which distance from any point this correlation occurs. The limiting distance should not be specified larger than the distance until which correlation occurs.
- ☞ Make sure that there are enough points within the limiting distance; in other words, choose a limiting distance, which is large enough. To check whether you have enough points within the limiting distance, you can perform the calculation again with a limiting distance increased by a factor 2. When you find profound differences in outcomes, you have chosen the limiting distance too small in the first calculation.

Dialog box

Dialog box options

Input point map:	Select an input point map. Open the list box and select the desired input map, or drag a point map directly from the Catalog into this box. You can select a point map with a value domain, or a point map with a class or ID domain, which has a linked attribute table with values.
Attribute:	In case you selected an input point map with a class or ID domain, select an attribute column (value domain) from the attribute table.
Weight function:	Select the weight function, which should be used to calculate weight factors for the points. Both weight functions ensure that points close to an output pixel will obtain a larger weight factor than points farther away. The inverse distance method assigns relatively larger weights to points close to an output pixel than the linear decrease method. For more information on weight functions, see Moving average: algorithm.

Inverse distance:	$(1 / d^n) - 1$
Linear decrease:	$1 – d^n$ d = relative distance of points towards pixels n = weight exponent
Weight exponent:	Type a value for weight exponent n to be used in the selected weight function (real value, usually a value around value 1.0).
Limiting distance:	Type a value for the limiting distance. Points that are farther away from any output pixel than the limiting distance are assigned weight zero; the values of these points will thus not be used in the calculation of the output value for that pixel.
Output raster map:	Type a name for the output raster map that will contain the weighted averaged point values.
Georeference:	Select the name of an existing georeference or create a new georeference. For time efficiency reasons, it is advisable to choose a rather large pixel size for the output raster map. Further interpolation on the raster values can be performed with the Densify operation or the Resample operation (using bilinear or bicubic interpolation).
Value range:	Accept the default value range, or specify your own range of possible values in the output map.
Precision:	Accept the default precision of output values, or specify your own precision.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

Command line

The Moving Average operation can be directly executed by typing one of the following expressions on the command line of the Main window:

OUTMAP = `MapMovingAverage`(InputPointMap, Georeference, *WeightFunction*)
OUTMAP = `MapMovingAverage`(PointMap.Column, Georeference, *WeightFunction*)

where:

OUTMAP	is the name of the output raster map.
`MapMovingAverage`	is the command to start the Moving Average operation.

InputPointMap	is the name of the input point map with a value domain.
PointMap.Column	is the name of an input point map with a class or ID domain which has a linked attribute table, and the name of a value column in this attribute table.
Georeference	is the name of an exisiting georeference for the output raster map.
WeightFunction	is an expressing which defines the type of weight function to be used, as: `InvDist`(*Exp*,*LimDist*) `Linear`(*Exp*,*LimDist*)

where:

`InvDist`	is the parameter to indicate the use of the inverse distance method.
`Linear`	is the parameter to indicate the use of the linear decrease method.
Exp	is a value for weight exponent *n* to be used in the specified weight function (real value, usually a value close to 1.0).
LimDist	is a value for the limiting distance: points that are farther away from an output pixel than the limiting distance obtain weight zero.

Example

A full expression for the Moving average operation, using weights according to the inverse distance method, a weight exponent 1, and a limiting distance of 500 m. might thus read:

OUTMAP = `MapMovingAverage(`MapX, GeorefX, `InvDist(1,500))`

When the definition symbol `=` is used, a dependent output map is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output map has been calculated.

Algorithm

Moving average performs a weighted averaging on point values and returns a raster map as output. The user has to specify a weight function and a limiting distance.

Step 1

For each output pixel, the distances of all points towards the output pixel are calculated to determine weight factors for the points:

For each output pixel, weight factors for the points are then calculated according to the weight function specified by the user. Two weight functions are available: inverse distance and linear decrease.

Inverse distance: weight = $(1 / d^n) - 1$
Linear decrease: weight = $1 - d^n$

where:
$d =$ D/D_0 = relative distance of point to output pixel
$D =$ Euclidean distance of point to output pixel
$D_0 =$ limiting distance
$n =$ weight exponent

Figures 1 and 2 below show the manner in which weight values decrease with increasing distance, for different values of n. The X-axes represent d: the distance of a point towards an output pixel divided by the limiting distance. The Y-axes represent the calculated **weight values**.

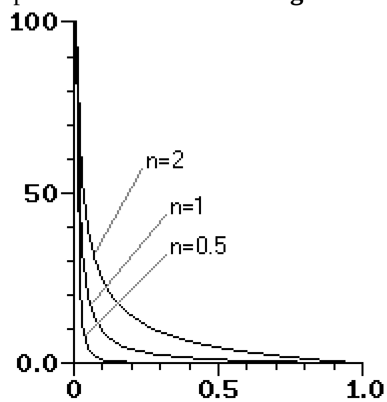


Figure 1: inverse distance
weight = $(1/d^n) - 1$
X-axis: $d = D/D_0$
Y-axis: weight values

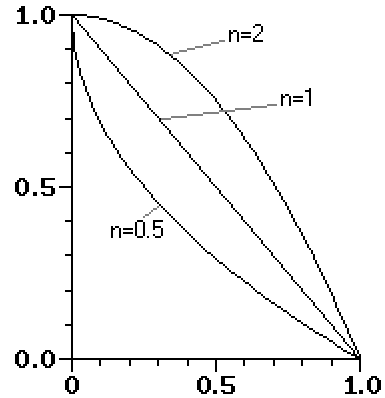


Figure 2: linear decrease
weight = $1 - d^n$
X-axis: $d = D/D_0$
Y-axis: weight values

The weight functions ensure that points close to an output pixel obtain a larger weight value than points, which are farther away from an output pixel.

See that when the distance of a point towards an output pixel equals the limiting distance (value 1.0 at X-axis), or when the distance of a point towards an output pixel is larger than the limiting distance, the calculated weight value will equal 0; the weight functions are thus continuous.

☞ The inverse distance function can be selected when you have very accurately measured point values and when local variation, within a pixel, is small. This function ensures that the computed output values equal the input point values.

☞ The linear decrease function can be selected for point maps in which you know there are measurement errors, and when points lying close to each other have different values. This function will decrease the overall error by correcting erroneous measurements with other close points. The consequence is that the computed output values will not necessarily coincide with the measured point values.

Step 2
Then, for each output pixel, an output value is calculated as the sum of the products of calculated weight values and point values, divided by sum of weights.

output pixel value = $\sum(w_i \times val_i) / \sum w_i$

Where:
w_i = weight value for point i
val_i = point value of point i

☞ For time efficiency reasons, it is strongly advised to choose a rather large pixel size for the output raster map. Further interpolation on the raster values can be performed with the Densify operation or the Resample operation (using bilinear or bicubic interpolation).

7.5.4 Trend surface

Functionality

The Trend Surface operation is a point interpolation, which requires a point map as input and returns a raster map as output. One polynomial surface is calculated by a least squares fit so that the global surface approaches all point values in the map. The calculated surface values are assigned to the output pixels.

The trend surfaces in this operation range from a simple plane to complex polynomial surfaces. You can usually select a first or second order function to calculate the surface, as these are the least sensitive to produce extreme values. For more information on the available functions, which calculate surfaces, see Trend surface: algorithm.

Input map requirements
The input point map should be a value map. Furthermore, when a point map uses a class or ID domain and the map is linked to an attribute table, you can also use such a point map and select a column with a value domain from the map's attribute table.

Domain and georeference of output map
The output raster map uses the same value domain as the input point map or the attribute column. The value range and precision can be adjusted for the output map. The georeference for the output map has to be selected or created; you can usually select an existing georeference corners.

☞ After performing the Trend surface operation, you can press the Additional Info button in the Properties dialog box of the output raster map to display the formula of the calculated surface.

- ☞ You can subtract the output map of the trend surface operation from the original point values, e.g. using the MAPVALUE() and PNTCRD() functions in TabCalc. If the residuals of this subtraction do not have a strong spatial correlation, the trend surface is a representative model of your point values.
- ☞ Prior to interpolation, you can use the Pattern analysis operation to investigate whether your points are randomly distributed, and the Spatial correlation operation to investigate whether your points are spatially correlated.

Dialog box

Dialog box options

Input point map:	Select an input point map. Open the list box and select the desired input map, or drag a point map directly from the Catalog into this box. You can select a point map with a value domain, or a point map with a class or ID domain, which has a linked attribute table with values.
Attribute:	In case you selected an input point map with a class or ID domain, select an attribute column (value domain) from the attribute table.
Surface:	Select one of the 8 functions to calculate a trend surface which approaches all points in your point map.
Output raster map:	Type a name for the output raster map.
Georeference:	Select the name of an existing georeference or create a new georeference.
Value range:	Accept the default value range, or specify your own range of possible values in the output map.
Precision:	Accept the default precision of output values, or specify your own precision.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

Command line

The Trend surface operation can be directly executed by typing one of the following expressions on the command line of the Main window:

OUTMAP = `MapTrendSurface`(InputPointMap, Georeference, *SurfaceType*)
OUTMAP = `MapTrendSurface`(PointMap.Column, Georeference, *SurfaceType*)

where:

OUTMAP	is the name of the output raster map.
`MapTrendSurface`	is the command to start the Trend surface operation.
InputPointMap	is the name of the input point map with a value domain.
PointMap.Column	is the name of an input point map with a class or ID domain which has a *linked* attribute table, and the name of a value column in this attribute table.
Georeference	is the name of an existing georeference that should be used for the output raster map.
SurfaceType	is the parameter which specifies the function with which a surface should be calculated; use one of the following: `Plane`|`Linear2`|`Parabolic2`|`2`|`3`|`4`|`5`|`6`

When the definition symbol `=` is used, a dependent output map is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output map has been calculated.

Algorithm

The Trend Surface operation is a point interpolation, which requires a point map as input and returns a raster map as output. One polynomial surface is calculated by a global least squares fit approaching all point values in the map. The calculated surface values are assigned to the output pixels.

Below the functions and surface types are listed, as well as the absolute minimum number of points that are mathematically required to fit such a surface. You will always need more points than this absolute mathematical minimum to obtain good results.

In general, the use of simple surfaces is preferred, as these will produce the least artificial extreme values.

Plane:	the surface is a plane, formula: $z = a + bx + cy$ Minimum number of points required: 3
2^{nd} degree Linear:	the surface is planar but tilted, i.e. first order plane, formula: $z = a + bx + cy + dxy$ Minimum number of points required: 4
2^{nd} degree Parabolic:	the surface is a second order polynomial surface, formula: $z = a + bx + cy + ex^2 + fy^2$ Minimum number of points required: 5
2^{nd} degree:	the surface is a full second order polynomial surface, formula: $z = a + bx + cy + dxy + ex^2 + fy^2$ Minimum number of points required: 6

3^{rd} degree: the surface is a third order polynomial surface
$z = a + ... + gx^3 + hx^2y + ixy^2 + jy^3$
Minimum number of points required: 10

4^{th} degree: the surface is a fourth order polynomial surface
$z = a + ... + kx + lx^3y + mx^2y^2 + nxy^3 + oy$
Minimum number of points required: 15

5^{th} degree: the surface is a fifth order polynomial surface
$z = a + ... +px + qxy +rx^3y^2 + ... + uy$
Minimum number of points required: 21

6^{th} degree: the surface is a sixth order polynomial surface
$z = a + ... + vx+...$
Minimum number of points required: 28

7.5.5 Moving surface

Functionality

The Moving surface operation is a point interpolation, which requires a point map as input and returns a raster map as output. For each output pixel, a polynomial surface is calculated by a moving least squares fit; for each output pixel, the surface will approach the weighted point values of the points which fall within the specified limiting distance (see below). The calculated surface values are assigned to the output pixels.

The weight factors for the input points are calculated by a user-specified weight function. There are two methods: inverse distance and linear decrease. Both methods ensure that points close to an output pixel obtain large weights and that points farther away from an output pixel obtain small weights. The values of points close to an output pixel are thus of greater importance to the output value for that pixel than points farther away.

By specifying a limiting distance, you can influence until which distance from any output pixel, points will be taken into account for the calculation a value for that output pixel. For each output pixel, only the values of the points falling within the limiting distance to this output pixel will be used to calculate a value for that output pixel. Values of points that are farther away from an output pixel than the specified limiting distance, obtain weight zero by the weight calculation, and these values will thus not be used in the output pixel value calculation. This speeds up the calculation and prevents artifacts.

The surfaces in this operation range from a simple plane to complex polynomial surfaces. You can usually select a first or second order surface, as these are the least sensitive to produce artificial extreme values.

For more information on weight calculation methods and surfaces, see Moving surface: algorithm.

Input map requirements
The input point map should be a value map. Furthermore, when a point map uses a class or ID domain and the map is linked to an attribute table, you can also use such a point map and select a column with a value domain from the map's attribute table.

Domain and georeference of output map
The output raster map uses the same value domain as the input point map or the attribute column. The value range and precision can be adjusted for the output map. The georeference for the output map has to be selected or created; you can usually select an existing georeference corners.

☞ For time efficiency reasons, it is advisable to choose a rather large pixel size for the output raster map. Further interpolation on the raster values can be performed with the Densify operation or the Resample operation (using bicubic interpolation).

☞ Prior to interpolation, you can use the Pattern analysis operation to investigate whether your points are randomly distributed, and the Spatial correlation operation to investigate whether your points are spatially correlated and until which distance from any point this correlation occurs. The limiting distance should not be specified larger than the distance until which correlation occurs.

☞ Make sure that there are enough points within the limiting distance; in other words, choose the limiting distance value large enough.
To check whether you have enough points within the limiting distance, you can perform the calculation again with a limiting distance increased by a factor 2. When you find profound differences in outcomes, you have chosen the limiting distance too small in the first calculation.

☞ To find only a regional trend, use the Trend surface operation, which is a relatively fast operation. When you use the residuals of the Trend surface operation as input for the Moving surface operation, you may be able to split regional and local phenomena.

Dialog box

Dialog box options

Input point map:	Select an input point map. Open the list box and select the desired input map, or drag a point map directly from the Catalog into this box. You can select a point map with a value domain, or a point map with a class or ID domain, which has a linked attribute table with values.
Attribute:	In case you selected an input point map with a class or ID domain, select an attribute column (value domain) from the attribute table.

Weight function:	Select the weight function, which should be used to calculate weight factors for the points. Both weight functions ensure that points close to an output pixel will obtain a larger weight factor than points farther away. The inverse distance method assigns relatively larger weights to points close to an output pixel than the linear decrease method. For more information on weight functions, see Moving surface: algorithm.
Inverse distance:	$(1 / d^n) - 1$
Linear decrease:	$1 - d^n$
	d = relative distance of points towards pixels
	n = weight exponent
Weight exponent:	Type a value for weight exponent *n* to be used in the selected weight function (real value, usually a value close to 1.0).
Limiting distance:	Type a value for the limiting distance. Points that are farther away from any output pixel than the limiting distance, are assigned weight zero; the values of these points will thus not be used in the calculation of the output value for that pixel.
Surface:	Select one of the 8 functions to calculate for each output pixel a surface which approaching all points within the limiting distance towards this output pixel.
Output raster map:	Type the name of the output raster map.
Georeference:	Type the name of an existing georeference or create a new georeference.
Value range:	Accept the default value range, or specify your own range of possible values in the output map.
Precision:	Accept the default precision of output values, or specify your own precision.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

Command line

The Moving surface operation can be directly executed by typing one of the following expressions on the command line of the Main window:

OUTMAP = `MapMovingSurface`(InputPointMap, Georeference, *SurfaceType*, *WeightFunction*)
OUTMAP = `MapMovingSurface`(PointMap.Column, Georeference, *SurfaceType*, *WeightFunction*)

where:

OUTMAP	is the name of the output raster map.
`MapMovingSurface`	is the command to start the Moving surface operation.
InputPointMap	is the name of the input point map with a value domain.
PointMap.Column	is the name of an input point map with a class or ID domain which has a linked attribute table, and the name of a value column in this attribute table.
GeoReference	is the name of an existing georeference that should be used for the output raster map.
SurfaceType	is the parameter which specifies the function with which surfaces should be calculated; use one of the following: `Plane` | `Linear2` | `Parabolic2` | `2` | `3` | `4` | `5` | `6`
WeightFunction	is an expressing which defines the type of weight function to be used, as: `InvDist`(*Exp*,*LimDist*) `Linear`(*Exp*,*LimDist*)

where:

`InvDist`	is the parameter to indicate the use of the inverse distance method.
`Linear`	is the parameter to indicate the use of the linear decrease method.
Exp	is a value for weight exponent *n* to be used in the specified weight function (real value, usually a value close to 1.0).
LimDist	is a value for the limiting distance: points that are farther away from an output pixel than the limiting distance obtain weight zero.

Example
A full expression for the Moving surface operation, using a plane, using weights according to the inverse distance method, a weight exponent 1, and a limiting distance of 500m. might thus read:

OUTMAP = `MapMovingSurface`(MapX, GeoRefX, `Plane`, `InvDist`(1,500))

When the definition symbol `=` is used, a dependent output map is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output map has been calculated.

Algorithm

The Moving surface operation is a point interpolation, which requires a point map as input and returns a raster map as output. For each output pixel, a polynomial surface is calculated by a least square method approaching weighted point values.

Step1
First, for each output pixel, the distances of all points towards the output pixel are calculated to determine weight factors for the points:

For each output pixel, weight factors for the points are then calculated according to the weight function specified by the user. Two weight functions are available: inverse distance and linear decrease.

Inverse distance: weight = $(1 / d^n) - 1$
Linear decrease: weight = $1 - d^n$

where:
d = D/D_0 = relative distance of point to output pixel
D = Euclidean distance of point to output pixel
D_0 = limiting distance
n = weight exponent

Figures 1 and 2 below show the manner in which weight values decrease with increasing distance, for different values of n. The X-axes represent d: the distance of a point towards an output pixel divided by the limiting distance. The Y-axes represent the calculated weight values.

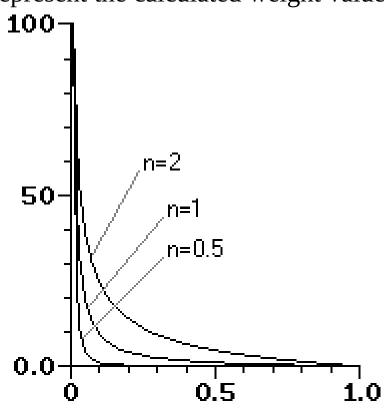


Figure 1: inverse distance
weight = $(1/d^n) - 1$
X-axis: $d = D/D_0$
Y-axis: weight values

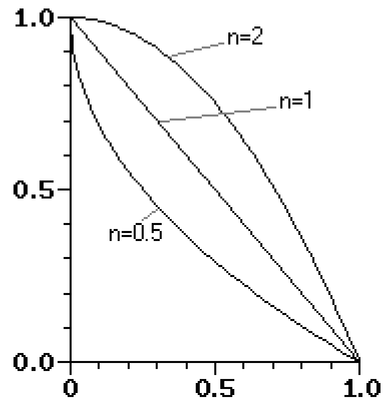


Figure 2: linear decrease
weight = $1 - d^n$
X-axis: $d = D/D_0$
Y-axis: weight values

The weight functions ensure that points close to an output pixel obtain a larger weight value than points farther away from an output pixel.

See that when the distance of a point towards an output pixel equals the limiting distance (value 1.0 at X-axis), or when the distance of a point towards an output pixel is larger than the limiting distance, the calculated weight value will equal 0; the weight functions are thus continuous.

☞ The *inverse distance* function can be selected when you have very accurately measured point values and when local variation, within a pixel, is small. This function ensures that the computed output values equal the input point values.

☞ The *linear decrease* function can be selected for point maps in which you know there are measurement errors, and when points close to each other have different values. This function will decrease the overall error by correcting erroneous measurements with other close points. The consequence is that the computed output values will not necessarily coincide with the measured point values.

Step 2

Next, for each output pixel, an output value is calculated by fitting a polynomial surface through all weighted point values of points that fall within the limiting distance towards this pixel. To each output pixel, the calculated surface value is assigned.

Below the functions and surface types are listed, as well as the absolute minimum number of points that are mathematically required to fit such a surface. To obtain good results, for each output pixel, you need more points within the limiting distance than this absolute mathematical minimum.

In general, the use of simple surfaces is preferred, as these will produce the least articial extreme values.

Plane:	the surface is a plane, formula: $z = a + bx + cy$ Minimum number of points required: 3
2nd degree Linear:	the surface is planar but tilted, i.e. first order plane, formula: $z = a + bx + cy + dxy$ Minimum number of points required: 4
2nd degree Parabolic:	the surface is a second order polynomial surface, formula: $z = a + bx + cy + ex^2 + fy^2$ Minimum number of points required: 5
2nd degree:	the surface is a full second order polynomial surface, formula: $z = a + bx + cy + dxy + ex^2 + fy^2$ Minimum number of points required: 6

3rd degree: the surface is a third order polynomial surface
$z = a + ... + gx^3 + hx^2y + ixy^2 + jy^3$
Minimum number of points required: 10

4th degree: the surface is a fourth order polynomial surface
$z = a + ... + kx + lx^3y + mx^2y^2 + nxy^3 + oy$
Minimum number of points required: 15

5th degree: the surface is a fifth order polynomial surface
$z = a + ... +px + qxy +rx^3y^2 + ... + uy$
Minimum number of points required: 21

6th degree: the surface is a sixth order polynomial surface
$z = a + ... + vx+...$
Minimum number of points required: 28

☞ For time efficiency reasons, it is strongly advised to choose a rather large pixel size for the output raster map. Further interpolation on the raster values can be performed with the Densify operation or the Resample operation (using bicubic interpolation).

7.5.6 Kriging

Functionality

Kriging can be seen as a point interpolation, which requires a point map as input and returns a raster map with estimations and optionally an error map. The estimations are weighted averaged input point values, similar to the Moving Average operation. The weight factors in Kriging are determined by using a user-specified semi-variogram model (based on the output of the Spatial correlation operation), the distribution of input points, and are calculated in such a way that they minimize the estimation error in each output pixel. The estimated or predicted values are thus a linear combination of the input values and have a minimum estimation error. The optional error map contains the standard errors of the estimates.

Kriging is named after D.G. Krige, a South African mining engineer and pioneer in the application of statistical techniques to mine evaluation. The Kriging technique is derived from the theory of regionalized variables (Krige, Matheron). An advantage of Kriging (above other moving averages like inverse distance) is that it provides a measure of the probable error associated with the estimates.

Preparation
Besides an input point map, Kriging requires a semi-variogram model including values for the parameters nugget, sill and range; this can be obtained from the Spatial correlation operation.

- Display the output table of Spatial correlation, and create a graph (i.e. a semi-variogram) in which you plot the semi-variance values against the distance classes.
- Subsequently, model the semi-variogram: select a model like Spherical, Exponential, Gaussian, etc., and choose values for sill, nugget and range. This shows as a line in the graph through the points.
- Superimpose trials of various models with various defining parameters in order to find the best approximation.

For more information, see Spatial correlation : functionality, section on Semi-variograms, or the Graph window : Add semi-variogram model.

☞ If a trend is apparent, it should be removed before Kriging. One could use the Trend surface operation and appropriate TabCalc subtractions.

Methods

Two Kriging methods are available: Simple Kriging and Ordinary Kriging.

- In Simple Kriging, all input points are used to calculate each output pixel value. Only one large matrix needs to be inverted to find the weight factors for all input points.
- In Ordinary Kriging, you can influence the number of points that should be taken into account in the calculation of an output pixel value by specifying a limiting distance and a minimum and maximum number of points:
  - only the points that fall within the limiting distance to an output pixel will be used in the calculation for that output pixel value,
  - within the limiting distance towards each output pixel, at least the specified minimum number of points should be found, otherwise the pixel will be assigned the undefined value.
  - within the limiting distance towards each output pixel, only the specified maximum number of points will be taken into account; when more than the specified maximum number of points are found within the limiting distance, only the points which are nearest to the output pixel will be taken into account.

  For each output pixel, a set of simultaneous equations needs to be solved to find the weight values for those points that contribute to the output value of the pixel.

In general, it can be said that the more points are used, the more reliable the estimation will be.

Removing duplicates or coinciding points

When you have multiple values for the same location or when point locations are very close to each other, i.e. when samples coincide, the Kriging system of equations becomes singular and cannot be solved.

It is therefore advised to use the option Remove Duplicates, which will automatically 'remove' any coinciding points. You can choose to either take the average value of coinciding points, or to take the value of the first (coinciding) point encountered only. By specifying a Tolerance, you can control the distance between points at which points are considered coinciding or not. When the distance between points is less than the specified tolerance, these points are considered coinciding; otherwise the points are considered distinct.

When your input data does contain coinciding points and when the Remove Duplicates option is not used, Simple Kriging will yield an error message, and Ordinary Kriging will assign the undefined value for all pixels to which the coinciding points make a contribution.

Error map
The error map contains the standard error of the estimate, i.e. the square root of the error variance.

The error variance in each estimated output pixel depends on:
- the semi-variogram model including its parameters,
- the spatial distribution of the input sample points,
- the position of an output pixel with respect to the position of the input sample points.

A standard error which is larger than the original sample standard deviation denotes a rather unreliable prediction.

Input map requirements
The input point map should be a value map. Furthermore, when a point map uses a ID domain and the map is linked to an attribute table, you can also use such a point map and select a column with a value domain from the map's attribute table. A current limitation of the operation is that Simple Kriging can only handle point maps with a maximum of 89 valid input points. Ordinary Kriging can only handle point maps that contain a maximum of 2500 input points. However, within each limiting distance (search radius) only a maximum of 89 valid points are allowed.

Domain and georeference of output raster map
The output raster map uses the same value domain as the input point map or the attribute column. The value range and precision can be adjusted for the output map; it is advised to choose a wider value range for the output map than the input value range. The georeference for the output map has to be selected or created; you can usually select an existing georeference corners.

☞ When the output raster map shows undefined pixels, this can be due to several factors:
- There are coinciding points while you did not use the option Remove Duplicates. Remedy: use the option Remove Duplicates, or change the tolerance. You can also remove duplicates yourself, e.g. by editing the point map in the point editor, by editing the point map as a table, or by editing the attribute table of the point map.
- The value range of the output raster map is too narrow: because of extrapolation, certain output values may fall beyond the range limits and ILWIS converts them

to undefined.
Remedy: extend the value range in the dialog box and compare the results.

- The minimum number of points to be taken into account is too high in relation to the limiting distance, i.e. the minimum number of points specified are not found within the specified limiting distance.
Remedy: lower the minimum number of points and/or increase the limiting distance.

☞ The output may also become erratic when:

- The input points are too sparse in certain areas to ensure an estimate with small variance.
Remedy: increase limiting distance, find supplementary point data, investigate possible anisotropy.
- The semi-variogram Range parameter is set incorrectly due to poor interpretation of a correct output of the Spatial correlation operation. This may occur when you chose the lag spacing in Spatial correlation far too large.
Remedy: Try another lag spacing and inspect the Nr of pairs column.
- The semi-variogram model is incorrect: the geometric distribution of the sample points is unbalanced or the user is unaware of an existing anisotropy. There is perhaps an error in the range setting (horizontal scale in graph) or sill and/or nugget (vertical scale).

Dialog box

Dialog box options

Input point map: Select an input point map. Open the list box and select the desired input map, or drag a point map directly from the Catalog into this box. You can select a point map with a value domain, or a point map with a ID domain which has a linked attribute table with values. A current limitation of the operation is that Simple Kriging can only handle point maps with a maximum of 89 valid input points; Ordinary Kriging can only handle point maps that contain a maximum of 2500 input points, however within each limiting distance (search radius) only a maximum of 89 valid points are allowed.

Attribute: In case you selected an input point map with a class or ID domain, select an attribute column (value domain) from the attribute table.

Semi-variogram Model:
Select the model, which should be used to calculate the semi-variogram function $\gamma(h)$. The available models are: Spherical, Exponential, Gaussian, Wave, Rational Quadratic, Circular, and Power. The model/function as well as the variables nugget, sill and range can be found modelling the semi-variogram which is the output of the Spatial correlation operation. For more information, see Spatial correlation : functionality section on semi-variograms, or Graph window : Add Graph Semi-variogram (dialog box).

Nugget: When you found a nugget effect, type a value for the semi-variance γ at distance 0, i.e. the intersection of your semi-variance model and the semi-variance axis (real value ≥ 0).

Sill: Type a value for the sill, i.e. the value where the semi-variance γ approaches the variance of your variable (real value > 0 and sill > nugget).

Range: Type a value for the range, i.e. the distance h at which the semi-variance of your variable is becoming 'stable' (real value > 0).

Slope: When using the Power model, type a value for the linear slope to be used; when a linear model is used, this is the direction coefficient, i.e. $\Delta\gamma/\Delta h$ (real value ≥ 0).

Power: When using the Power model, type a value for the exponent to be used ($0 \leq$ real value ≤ 10). It is advised to use a value between 0 and 2. When you use value 1, the power model will become linear.

Methods: In general, points close to an output pixel will obtain a larger weight value than points farther away.

Simple Kriging: The Simple Kriging method assigns weights to all points of the input map; for the estimation of each output pixel value, all input point values will be used.

Ordinary Kriging: The Ordinary Kriging method uses for the estimation of each output pixel value only the values of points that are within the user-specified limiting distance towards a pixel. This gives the estimation method a more local character.

Limiting distance: For Ordinary Kriging only: type a value for the limiting distance, also called search radius or limiting circle. Points that are farther away from any output pixel than the limiting distance are assigned weight zero; the values of these points will thus not be used in the calculation of the output value for that pixel and they will not influence the Kriging equations either. The distance should be positive and normally less than the range of the semi-variogram.

Minimum nr of points: For Ordinary Kriging only: type a value for the minimum number of points to make sure that the estimation is based at least this many points. When for an output pixel, not enough points are found within the specified limiting distance, then the undefined value will be assigned to the output pixel. It is advised to use at least 4 points.

Maximum nr of points: For Ordinary Kriging only: type a value for the maximum number of points that should be used by the calculation. When for an output pixel, more points are found within the limiting

	distance than specified, then only the points nearest to the output pixel will be used in the calculation. By specifying a rather small maximum, the algorithm may be faster but the estimation quality may be less.
Remove duplicates:	In case of coinciding points, it is advised to always select this check box. Subsequently, select 'Average' or 'First value'. You can clear this check box when you are sure that you have no coinciding points. If the check box is cleared and any coinciding points, i.e. duplicates, are found, Simple Kriging will fail, while Ordinary Kriging will result in undefined values in the neighbourhood of the coinciding points.
Average:	For points that are less than the specified tolerance distance apart, use the mean value of these points.
First value:	For points that are less than the specified tolerance distance apart, only use the value of the first point encountered.
Tolerance (m):	Type a value (meters) for the tolerance distance: points that are found less than this distance apart are considered coinciding points or duplicates.
Output raster map:	Type a name for the output raster map that will contain the Kriging estimates.
Georeference:	Select the name of an existing georeference or create a new georeference.
Value range:	Accept the default value range, or specify your own range of possible values in the output map. It is advisable to make the value range for the output map wider than the input value range; as negative weights may be used, Kriging estimates may be greater or smaller than your original input values. Mind: in case estimates are calculated that fall outside the specified value range, the pixel will be assigned the undefined value.
Precision:	Accept the default precision of output values, or specify your own precision.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.
Error map:	Select this check box when you also want to obtain an error map. The error map will contain the standard error of the estimates, i.e. the square root of the error variance. Subsequently, type a name for the error map; this name should be different from the name specified for the Kriging output map. Pixel values in the output error map will have value 0

when an input point exactly coincides with the center of the pixel.

A dependent output map is created. Optionally, a second output map containing standard errors can be created.

Command line

The Kriging operation can be directly executed by typing one of the following expressions on the command line of the Main window:

OUTMAP =	`MapKrigingSimple`(InputPointMap, Georef, *SemiVarModel*)
OUTMAP =	`MapKrigingSimple`(InputPointMap, Georef, *SemiVarModel*, ErrorMap)
OUTMAP =	`MapKrigingSimple`(InputPointMap, Georef, *SemiVarModel*, ErrorMap, `No` | `Average` | `Firstval`)
OUTMAP =	`MapKrigingSimple`(InputPointMap, Georef, *SemiVarModel*, ErrorMap, `Average` | `Firstval` , *Tolerance*)
OUTMAP =	`MapKrigingSimple`(InputPointMap, Georef, *SemiVarModel*, , `No` | `Average` | `Firstval`)
OUTMAP =	`MapKrigingSimple`(InputPointMap, Georef, *SemiVarModel*, , `Average` | `Firstval` , *Tolerance*)
OUTMAP =	`MapKrigingOrdinary`(InputPointMap, Georef, *SemiVarModel*, *LimDist*)
OUTMAP =	`MapKrigingOrdinary`(InputPointMap, Georef, *SemiVarModel*, *LimDist*, ErrorMap)
OUTMAP =	`MapKrigingOrdinary`(InputPointMap, Georef, *SemiVarModel*, *LimDist*, ErrorMap, *min*, *max*)
OUTMAP =	`MapKrigingOrdinary`(InputPointMap, Georef, *SemiVarModel*, *LimDist*, ErrorMap, *min*, *max*, `No` | `Average` | `Firstval`)
OUTMAP =	`MapKrigingOrdinary`(InputPointMap, Georef, *SemiVarModel*, *LimDist*, ErrorMap, *min*, *max*, `Average` | `Firstval`, *Tolerance*)

where:

OUTMAP	is the name of the output raster map.
`MapKrigingSimple`	is the command to start the Kriging operation, using the Simple Kriging method.
`MapKrigingOrdinary`	is the command to start the Kriging operation, using the Ordinary Kriging method.
InputPointMap	is the name of the input point map with a value domain.
Georef	is the name of an existing georeference for the output raster map.
SemiVarModel	*Model(nugget, sill, range)* | `Power`*(nugget, slope, pow)* This expression defines the semi-variogram model that should be used and the expected parameters.

Model	`Spherical|Exponential |Gaussian|` `Wave|RatQuad|Circular`
nugget	value for the nugget, according to your semi-variogram; real value ≥ 0.
sill	value for the sill, according to your semi-variogram; real value > 0 and sill > nugget.
range	value for the range, according to your semi-variogram; real value > 0.
slope	when using the Power model, specify a value for the linear slope to be used; when a linear model is used, this is the direction coefficient, i.e. $\Delta\gamma/\Delta h$; real value ≥ 0.
pow	when using the Power model, specify an exponent; 0 ≤ real value ≤ 10; it is advised to use a value between 0 and 2. When you use value 1, the power model will become linear.
LimDist	For Ordinary Kriging: a value for the limiting distance: points that are farther away from an output pixel than the limiting distance will not be used in the Kriging equations.
ErrorMap	Optional parameter to calculate an error map which will contain the square root of the Kriging error variance values, i.e. standard deviations per pixel. When this parameter is not specified, no error map will be calculated.
min, max	Optional parameters to specify the minimum and maximum number of points that should be taken into account per Kriging estimate/prediction, i.e. for the calculation of any output pixel value. When the minimum is not specified, value 1 will be used. When, for an output pixel, less points than the specified minimum are found within the specified limiting distance, no Kriging is performed: the output pixel will be assigned the undefined value. When the maximum is not specified, value 16 will be used. When, for an output pixel, more points than the specified maximum are found within the specified limiting distance, only the specified maximum number of points that are nearest to the output pixel will be used.
`No|Average|Firstval`	Choose how to handle possible coinciding points. When no method is specified, Average will be used.
`No`	No removal of duplicates. Mind: when there are coinciding points, the Kriging system may become unstable or unsolvable.
`Average`	Values of duplicates are replaced by the average (arithmetic mean); the coordinates of the first point are taken as new position.

`Firstval`	Values of duplicates (and their coordinates) are replaced by those of the first point.
Tolerance	Optional parameter to specify the distance within which 2 points are considered to coincide. When not specified, a tolerance of 0.1 m will be used.

When the definition symbol `=` is used, a dependent output map is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output map has been calculated.

Example

To perform Ordinary Kriging on input map MPX,

- producing output raster map OUTMAP with georef MYGRF,
- using a Spherical Semi-variogram model with nugget 10 m, sill 70 m and range 0.8 km (found by means of SpatCorr),
- using a limiting distance of 0.2 km,
- producing an error map ERM,
- while each Kriging equation should use at least 5 and at most 16 input points (min, max),
- while considering points that are closer to each other than 10 cm as coinciding (tolerance),
- while for coinciding points only the value of the first point encountered should count,

you can use the following expression:

```
OUTMAP=  MapKrigingOrdinary(MpX,MyGrf,
         Spherical(10,70,800),
         200,ErM,5,16,Firstval,0.10)
```

Algorithm

Kriging can be seen as a point interpolation, which requires a point map as input and returns a raster map with estimations and optionally an error map.

The estimations or predictions are calculated as weighted averages of known input point values, similar to the Moving Average operation.

The estimate to be calculated, i.e. an output pixel value $\hat{Z}$, is a linear combination of weight factors (w_i) and known input point values (Z_i):

$$\hat{Z} = \sum (w_i \times Z_i)$$

In case the value of an output pixel would only depend on 3 input points, this would read:

$$\hat{Z} = w_1 \times Z_1 + w_2 \times Z_2 + w_3 \times Z_3$$

Thus, to calculate one output pixel value $\hat{Z}$, first, three weight factors w_1, w_2, w_3 have to be found (one for each input point value Z_1, Z_2, Z_3), then, these weight factors can be multiplied with the corresponding input point values, and summed.

In Moving average, the weight factors are simply determined by the distances of the input points towards an output pixel. In Kriging, however, the weight factors are calculated by finding the semi-variance values for all distances between input points and by finding semi-variance values for all distances between an output pixel and all input points; then a set of simultaneous equations has to be solved.
All semi-variance values are calculated by using a user-specified semi-variogram model (based on the output of the Spatial correlation operation). The weight factors are calculated in such a way that the estimation error in each output pixel is minimized.

The optional error map contains the standard errors of the estimates.

Process Simple Kriging

1. Find the valid input points:
   - input points which coordinates are undefined are ignored,
   - input points which value is undefined are ignored,
   - handle duplicates or coinciding points as specified by the user (no, average, first value).

2. Determine the distances between all valid input points (n) and find the semi-variance value for these distances:
   - for each combination of 2 input points, i.e. a point pair, the distance between the points is determined,
   - for each combination of 2 input points, the distance value is substituted in the user-selected semi-variance model, using the user-specified nugget, sill, and range parameters; this gives a semi-variance value.
   - the semi-variance values are filled out in matrix C (as in Equation 1 below),
   - matrix C is inverted as a preparation for calculations in step 4.

3. *For the first output pixel*, determine the distances of this pixel towards all input points, and find the semi-variance value for these distances:
   - semi-variances are determined using the selected semi-variance model and its parameters as above,
   - the semi-variance values are filled out in vector D (as in Equation 1).

4. Calculate the weight factors (vector w):
   - by multiplying the inverted matrix C (result of step 2) with vector D (result of step 3).

   The obtained weight factors apply to the current output pixel only.

5. Calculate the estimated or predicted values for this output pixel:
   - as the sum of the products of the weight factors and the input point values (Equation 4).

6. Optionally, calculate the error variance and standard error for this output pixel:
   - error variance: by multiplying vector w (result of step 4) with vector D (result of step 3),

- standard error or standard deviation: as the square root of the error variance, according to Equation 5b.

7. Consider the next output pixel and repeat steps 3-7, for all output pixels.

Ordinary Kriging

1. Find the valid input points:
   - input points which coordinates are undefined are ignored,
   - input points which value is undefined are ignored,
   - handle duplicates or coinciding points as specified by the user (no, average, first value).

2. *For the first output pixel*, determine the input points (n) which will make a contribution to the output value depending on the specified limiting distance and minimum and maximum number of points:
   - input points that are farther away from this output pixel than the specified limiting distance are ignored,
   - if the number of points found within the limiting distance is smaller than the specified minimum nr of points, assign the undefined value to this output pixel,
   - use only the specified maximum number of points within the limiting distance, and, in case more points are found within the limiting distance than the specified maximum number of points, use only the points that are nearest to this output pixel.

3. Determine the distances between all input points that will make a contribution to this output pixel (result of step 2), and find the semi-variance value for these distances.
   - for each combination of 2 contributing input points, the distance between the points is determined,
   - for each combination of 2 contributing input points, the distance value is substituted in the user-selected selected semi-variance function, using the user-specified nugget, sill, and range parameters; this gives a semi-variance value.
   - the semi-variance values are filled out in matrix C below (eq. 1).

4. Determine the distances of this output pixel towards all input points, and find the semi-variance value for these distances:
   - semi-variances are determined using the selected semi-variance function or model and its parameters as above,
   - the semi-variance values are filled out in vector D (eq. 1).

5. Calculate the weight factors (vector w):
   - by first inverting matrix C (result of step 3),
   - by solving the set of simultaneous equations.

   The obtained weight factors apply to the current output pixel only.

6. Calculate the estimated or predicted values for this output pixel:

- as the sum of the products of the weight factors and the input point values (Equation 4).

7. Optionally, calculate the error variance and standard error for this output pixel:
   - error variance: by multiplying vector w (result of step 4) with vector D (result of step 3), according to Equation 5a.
   - standard error or standard deviation: as the square root of the error variance, according to Equation 5b.

7. Consider the next output pixel and repeat steps 2-8, until all output pixels are done.

Formulae to calculate weight factors

The Kriging weight factors of n valid input points p_i (i = 1…n) are found by solving the following matrix equation:

$$(C) = (w) \bullet (D) \qquad (1)$$

or

$$\begin{pmatrix} 0 & \gamma(h_{12}) & \gamma(h_{13}) & \dots & \gamma(h_{1n}) & 1 \\ \gamma(h_{21}) & 0 & \gamma(h_{23}) & \dots & \gamma(h_{2n}) & 1 \\ \gamma(h_{31}) & \gamma(h_{32}) & 0 & \dots & \gamma(h_{3n}) & 1 \\ \dots & \dots & \dots & \dots & \dots & 1 \\ \gamma(h_{n1}) & \gamma(h_{n2}) & \gamma(h_{n3}) & \dots & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \bullet \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \dots \\ w_n \\ \lambda \end{pmatrix} = \begin{pmatrix} \gamma(h_{01}) \\ \gamma(h_{02}) \\ \gamma(h_{03}) \\ \dots \\ \gamma(h_{0n}) \\ 1 \end{pmatrix}$$

This matrix equation can be written as a set of n+1 simultaneous equations:

$$\sum_i \left(w_i \times \gamma(h_{ik})\right) + \lambda = \gamma(h_{0i}) \qquad \text{for } k=1,\dots,n \qquad (2)$$

$$\sum_i w_i = 1 \qquad (3)$$

where:

h_{ik}	is the distance between input point pi and input point p_k
h_{0i}	is the distance between the output pixel and input point p_i
$\gamma(h_{ik})$	is the value of the semi-variogram model for the distance h_{ik}, i.e. the semi-variance value for the distance between input points p_i and input point p_k;
$\gamma(h_{0i})$	is the value of the semi-variogram model for the distance h_{0i} , i.e. the semi-variance value for the distance between the output pixel and input point p_i
w_i	is a weight factor for input point p_i
λ	is a Lagrange multiplier, used to minimize possible estimation error

Matrix C thus contains the semi-variances for all combinations of valid input points that will make a contribution to the output pixel value.
Vector w thus contains the weight factors for all valid input points that will make a contribution to the output pixel value.

Vector D thus contains the semi-variances for an output pixel and all combinations of valid input points.
Equation (3) guarantees unbiasedness of the estimates. The solutions w_i minimize the Kriging error variance σ^2.

Formulae to calculate an estimate or predicted value for an output pixel

$$\hat{Z} = \sum_i \left(w_i + Z_i\right) \qquad (4)$$

where:

$\hat{Z}$	is the estimate or predicted value for one output pixel to be calculated
w_i	is the weight factor for input point p_i
Z_i	is the value of input point p_I

Formulae to calculate error variance and standard error

The error variance is calculated as:

$$s^2 = \sum_i \left(w_i \times g\,(h_{01})\right) + l \qquad (5a)$$

The standard error or standard deviation is the square root of the error variance, thus:

$$s = \sqrt{\left(\sum_i \left(w_i \times g\,(h_{01})\right) + l\right)} \qquad (5b)$$

where:

σ^2	is the error variance for the output pixel estimate
σ	is he standard error or the standard deviation of the output pixel estimate
h_{0i}	is the distance between the output pixel and input point p_i
$\gamma(h_{0i})$	is the value of the semi-variogram model for the distance h_{0i} , i.e. the semi-variance value for the distance between the output pixel and input point p_i
w_i	is a weight factor for input point p_i
λ	is a Lagrange multiplier, used to minimize possible estimation error

Notes:

- The contents of matrix C depends on the semi-variogram model selected by the user, its parameters nugget, sill and range, and the geometric distribution of the points within the limiting circle in the input map.
- The contents of vector D is determined by the location of the estimated pixel value with respect to the surrounding input points (inside the limiting circle) and the semi-variogram.
- The estimates are computed as linear combinations of the n point sample values with the weights w_i as coefficients (the w_i are found from equation (1)). Therefore the estimates are called 'linear predictors'.
- Equation (3) guarantees unbiasedness of the estimates. The solutions w_i minimize the Kriging error variance σ^2.
- Equation (5) does not contain the sample attribute information. This means that the error variances solely depend on the spatial distribution of the samples and not on their measurement values (the attribute values).

- In the case of Simple Kriging, it is assumed that all input points contribute in some way to the estimate in each pixel. The Kriging matrix has thus a constant value for all pixels estimated and needs to be inverted only once; however the right hand-side D keeps changing.
- In Ordinary Kriging the number of points used ($n \leq N$) and hence the size of the Kriging matrix (n+1) will change from pixel to pixel while calculating the output map(s). Hence it is theoretically possible that for each output pixel a new Kriging system of order n+1 has to be solved. The algorithm also takes care that for each new set of surrounding input points, this set is sorted according to distance from the estimated pixel in order to enable to select the closest points satisfying the max nr of points condition.

References:

- Clark, I. 1979. Practical geostatistics. Applied Science Publishers, London. 129 pp.
- Cressie, N.A.C. 1993. Statistics for spatial data. Wiley, New York. 900 pp.
- Davis, J. C. 1973. Statistics and data analysis in geology. Wiley, New York. 646 pp.
- Deutsch, C.V., and A.G. Journel. 1992. Geostatistical software library and user's guide. Oxford University Press, New York. 340 pp.
- Isaaks, E. H., and R. M. Srivastava. 1989. An introduction to applied geostatistics. Oxford University Press, New York. 561 pp.
- Journel, A. G. and Ch. J. Huijbregts. 1978. Mining geostatistics. Academic Press, London, 600 pp.
- Krige, D.G. Two-dimensional weighted moving average trend surfaces for ore-valuation, in Proc. Symposium on Mathematical Statistics and Computer Applications in Ore Valuation: Journ. South African Inst. Mining and Metallurgy, Johannesburg, 1966, Mar. 7-8, pp. 13-38.
- Matheron, G.F. Principles of geostatistics: Economic Geology, 1963, vol. 58, pp.1246-1266.
- Meer, F. D. van der. Introduction to geostatistics. ITC lecture notes. 72 pp.
- Olea, R.A. 1991. Geostatistical glossary and multilingual dictionary. Oxford University Press, New York.
- Stein, A. 1998. Spatial statistics for soils and the environment. ITC lecture notes. 47 pp.

7.6 Vector operations

7.6.1 Transform vector map

Functionality / Algorithm

The transform operation is identical for all vector maps: polygon map, segment map and point map.

The Transform map operation transforms the XY-coordinates in a vector map from the map's current coordinate system to another target coordinate system. For polygon maps this concerns the polygon boundaries, for a segment map the segments, and for a point map the point locations. The Transform operation can only be used when a transformation between the coordinate systems is possible.

With the transform operation, you can:

- transform a vector map with a *coordinate system projection* with a certain projection, ellipsoid and/or datum to *another coordinate system projection* with a different projection, ellipsoid and/or datum;
- transform a vector map with a *coordinate system projection* with a certain projection, ellipsoid and/or datum to a *coordinate system latlon* with a certain ellipsoid and/or datum (and vice versa);
- transform a vector map with a *coordinate system latlon* with a certain ellipsoid and/or datum to *another coordinate system latlon* with a different ellipsoid and/or datum;
- transform a vector map with a *coordinate system formula* to the '*related coordinate system*' of this coordinate system formula (and vice versa);
- transform a vector map with a *coordinate system tiepoints* to the '*related coordinate system*' of this coordinate system tiepoints (and vice versa).

Furthermore:

- in a coordinate system without an ellipsoid specification, a sphere is assumed
- when in only one of the coordinate systems a datum is specified, then this datum is assumed for both.

For more information on coordinate system types, see ILWIS objects: coordinate systems.

In general, the operation can be used:

- to integrate data which are obtained from different sources and when these data are in different projections,
- to present data or results in a more appropriate projection.

Preparations for using a coordinate system projection

- When you want to integrate data of different projections, first think of the most suitable projection in which you can do your analysis: when area calculations are involved, you should use an Equal Area projection.
  You can either:
  - transform your maps to one existing coordinate system; select this existing coordinate system as the target coordinate system during Transform operations.
  - first create a new coordinate system, specify projection information, and then transform all your maps to that new coordinate system.
- When you want to present data in other projections, you can generally first create a number of new coordinate systems (using File, Create, or by clicking the create button in the dialog box of a Transform operation) and then transform your map to these new coordinate systems. During the Transform operations,

specify a different output map name for each selected target coordinate system. For presentation purposes, you can also interactively change the coordinate system as used by maps displayed in a map window. For more information, see the tips below.

Input map requirements
A transformation is only possible between:

- a coordinate system projection with a known projection, ellipsoid, and/or datum, and
  - another coordinate system projection with another projection, ellipsoid, and/or datum, or
  - a coordinate system latlon with a known ellipsoid and/or datum, or
- a coordinate system latlon with a known ellipsoid and/or datum, and
  - another coordinate system latlon with a known ellipsoid and/or datum, or
- a coordinate system formula, and
  - the 'related coordinate system' of this coordinate system formula, or
- a coordinate system tiepoints, and
  - the 'related coordinate system' of this coordinate system tiepoints.

In each of these combinations, one coordinate system is the current coordinate system of the vector map, and the other is the selected target coordinate system.

Domain and coordinate system of output map
The output vector map uses the same domain as the input vector map.
The output vector map uses the selected target coordinate system; the coordinate boundaries of the output map will be the transformed coordinate boundaries of the input map.

☞ To see the effect of using different projections, it is advisable to display the input map in a map window and the output maps in other map windows; then add a graticule to the map windows; open the Layers menu and select the Add Annotation, Graticule command.

☞ The Transform operations permanently change the projection of your map(s), i.e. for analysis and calculation purposes.
To temporarily view map(s) which are displayed in a map window in another projection, i.e. for presentation purposes, you can:

- create a coordinate system in which you already specify some projection information,
- display your map(s) in a map window,
- drag your new coordinate system to the map window, or choose the Coordinate System command from the Options menu in the map window and subsequently select another coordinate system.

The contents of the map window will be displayed in the new projection. Projection information of the coordinate system as currently used by the map window can be refined by choosing the Coordinate System command from the Edit menu in the map window. Thus, you do not need to use a Transform operation.

☞ Polygon Map: During a transformation, straight lines which are simply defined by a start node and an end node will always remain straight as only the coordinates of the begin and end node change. Polygon boundaries however generally consist of a start node and an end node and many intermediate points in between. Then, before using a Transform operation, it is strongly advised to increase the number of intermediate points in all polygon boundaries; in this way, the shape of polygons will be preserved during the transformation.

- First extract segments from your polygons (in the Polygon editor or with the Polygons to segments operation),
- Then, use the Densify segment coordinates operation.

After this, use the Transform segment map operation and repolygonize afterwards, or first repolygonize your segments and then use the Transform polygon map operation.

☞ Segment Map: During a transformation, straight lines which are simply defined by a start node and an end node will always remain straight as only the coordinates of the begin and end node change. Segments however generally consist of a start node and an end node and many intermediate points in between. Then, before using a Transform operation, it is strongly advised to increase the number of intermediate points in all segments with the Densify segment coordinates operation. In this way, the shape of segments will be preserved during the transformation.

Algorithm

The XY-coordinate pairs of the polygon boundaries of the input map are copied, and then transformed.

- When both the input coordinate system and the target coordinate system have the same ellipsoid and datum information, then the transformations are calculated via geographic coordinates; i.e. from XY (input) to φ, λ (LatLon) to XY (target).
- When in the input coordinate system and the target coordinate system, ellipsoid and datum information is different, then Molodensky formulas are used to calculate the ellipsoidal transformations and datum shifts, i.e. from XY (input) to φ, λ (input) to φ, λ (target) to XY (target).

Finally, new polygon areas and perimeters are calculated.

Dialog box

Dialog box options

Input vector map:	Select an input vector map. Open the list box and select the desired input map, or drag a polygon map directly from the Catalog into this box. The coordinate system of the input vector map is the coordinate system you will transform coordinate pairs *from*.
Coordinate system:	Select an existing target coordinate system or create a new coordinate system by using the create button. This is the coordinate system you will transform the coordinate pairs of your input map *to*.
Output vector map:	Type a name for the output vector map that will contain the transformed coordinates.

Show:	Select this box if you want the output map to be displayed in a map window when the operation has finished. Clear this box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

Command line

The Transform vector map operation can be directly executed by typing one of the following expressions on the command line of the Main window:

OUTMAP = `PolygonMapTransform` (InputPolygonMap, CoordinateSystem)
OUTMAP = `SegmentMapTransform` (InputSegmentMap, CoordinateSystem)
OUTMAP = `PointMapTransform` (InputPointMap, CoordinateSystem)

where:

OUTMAP	is the name of your output polygon map.
`PolygonMapTransform`	is the command to start the Transform Polygon Map operation.
`SegmentMapTransform`	is the command to start the Transform Segment Map operation.
`PointMapTransform`	is the command to start the Transform Point Map operation.
InputPolygonMap | InputSegmentMap | InputPointMap	is the name of the input vector map. The coordinate system of the input vector map is the coordinate system *from* which you will transform.
CoordinateSystem	is the name of your target coordinate system; this is the coordinate system *to* which you will transform.

When the definition symbol `=` is used, a dependent output map is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output map has been calculated.

7.6.2 Transform coordinates

The Transform coordinates dialog box allows you to type XY-coordinates or LatLon coordinates, using a certain input coordinate system; the operation will then show the resulting XY-coordinates or LatLon coordinates for another target coordinate system. The Transform Coordinates dialog box can only be used when a transformation between the coordinate systems is possible.

A coordinate transformation can be useful to check whether a transformation is correct. When typing an XY-coordinate of a point of interest to be transformed, you are able to interactively view the output coordinates for the selected point.

This dialog box appears:
- when you choose Points, Transform Coordinates from the Operations, Vector Operations menu in the Main window, or
- when you double-click the Transform Coordinates item in the Operation-list, or
- when you click a coordinate system in the Catalog with the right mouse button and choose Vector Operations, Transform Coordinates from the context-sensitive menu.

The Transform coordinates dialog box can be used to:
- transform coordinates from a *coordinate system projection* with a certain projection, ellipsoid and/or datum to *another coordinate system projection* with a different projection, ellipsoid and/or datum;
- transform coordinates from a *coordinate system projection* with a certain projection, ellipsoid and/or datum to a *coordinate system latlon* with a certain ellipsoid and/or datum (and vice versa);
- transform coordinates from a *coordinate system latlon* with a certain ellipsoid and/or datum to *another coordinate system latlon* with a different ellipsoid and/or datum;
- transform coordinates from a *coordinate system formula* to the *'related coordinate system*' of this coordinate system formula (and vice versa);
- transform coordinates from a *coordinate system tiepoints* to the *'related coordinate system*' of this coordinate system tiepoints (and vice versa).

Furthermore:
- when in a coordinate system an ellipsoid is not specified, then a sphere is assumed, and
- when in either of the 2 coordinate systems a datum is not specified, then the same datum is assumed.

For more information on coordinate system types, see ILWIS objects : coordinate systems.

Dialog box options

Input coordinate system:	Select an input coordinate system. Open the list box and select the desired coordinate system, or drag a coordinate system directly from the Catalog into this box. If a description exists for this coordinate system, this description will appear just below this Input coordinate system list box.
Input coordinate:	Type the X and Y coordinates of a point of interest.
Output coordinate system:	Select a target coordinate system. Open the list box and select the desired coordinate system, or drag a coordinate system directly from the Catalog into this box. If a description exists for this coordinate

	system, this description will appear just below this Output coordinate system list box.
Output coordinate:	The calculated coordinates for the point of interest in the output projection will be displayed.

Note: If a *coordinate system latlon* is used, coordinates are presented as geographic coordinates.

☞ When you wish to compare coordinate transformation results of *multiple* coordinate systems, you can add the different coordinate systems to the pixel information window. Open a map with 'the input coordinate system' and move the mouse pointer or click at a position of interest; when a transformation from the current coordinate system to the other coordinate system(s) is possible, the pixel information window will show the transformation results.

7.7 Rasterize

7.7.1 Polygons to raster

Functionality

The Polygons to raster operation rasterizes a polygon map. The output raster map always uses the same domain as the input polygon map. This means that the class names, IDs, or values used in the polygon map are also used in the raster map.

For the output raster map, an existing georeference has to be selected or a new one can be created. The georeference determines the number of lines and columns of the output map and the pixel size of the map, see also the examples in the Additional information below. It is strongly advised that vector maps of the same area are rasterized on the same georeference: any map calculation or spatial operation performed later on a combination of raster maps will only make sense if the pixels in these maps refer to the same area on the ground.

When a polygon map has an attribute table or when the domain of the polygon map has an attribute table, the Polygons to raster operation automatically links this attribute table to the output raster map.

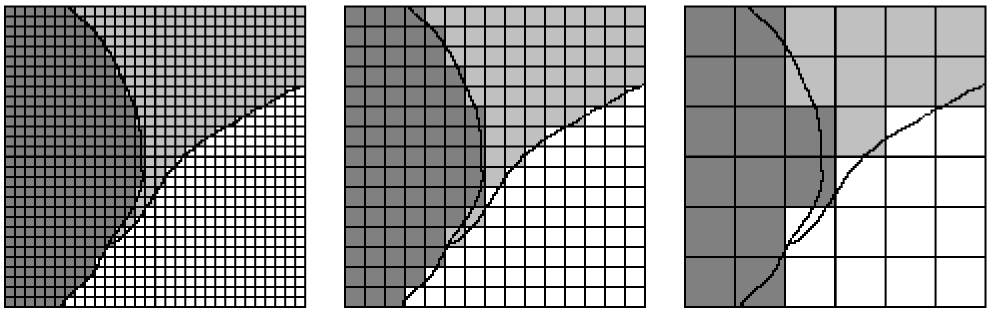
Input map requirements
No special input map requirements. However, at the moment it is not yet possible to rasterize a polygon map with a value domain into a raster map which is stored using 8 bytes per pixel. In that case it is advised to set the value range of the output raster map in such a way that the raster map can be stored with 1, 2, or 4 bytes per pixel.

Domain and georeference of output map
The output raster map uses the same domain as the input polygon map.
The georeference for the output map has to be selected or created; you can usually select an existing georeference corners. The georeference for the raster map must

use the same coordinate system as the polygon map. Georeference `None` cannot be selected for the output map.

☞ To obtain frequency information on polygons, you can calculate the histogram of a polygon map with the Histogram operation.

Additional information
The georeference you use for the output raster map determines the pixel size of the raster map and thus whether shapes of polygons are well retained. Below you find three examples of rasterized polygon maps, each one with a different pixel size.



Dialog box

Dialog box options

Input polygon map:	Select an input polygon map. Open the list box and select the desired input map, or drag a polygon map directly from the Catalog into this box.
Output raster map:	Type a name for the output raster map that will contain the rasterized polygons.
Georeference:	Select an existing georeference for the output raster map; open the list box by clicking it. Or create a new georeference by clicking the little create button. You can usually select an existing georeference corners.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

Command line

The Polygons to raster operation can be directly executed by typing the following expression on the command line of the Main window:

OUTMAP = `MapRasterizePolygon`(InputPolygonMapName, Georeference)

where:

OUTMAP	is the name of your output raster map.
`MapRasterizePolygon`	is the command to start the Polygons to raster operation.
InputPolygonMapName	is the name of your input polygon map.
Georeference	is the name of an existing georeference. At the moment, it is not yet possible to create a new georeference on the command line.

When the definition symbol `=` is used, a dependent output map is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output map has been calculated.

Algorithm

The Polygons to raster operation rasterizes a polygon map. The output raster map always uses the same domain as the input segment map. The user has to specify the georeference for the output raster map.

Process
For each polygon, the corresponding pixels in the raster map are found; these pixels are assigned the class name, ID or value of the polygon.
Other pixels obtain the undefined value.

- The pixel size, indicated in the georeference for the output map, has a large influence on the size (in bytes) of the output raster map. Mind that rasterising on a pixel size of 10 m instead of 50 m. increases the size of the raster map by a factor 25.
- If you want to rasterize only a portion of your polygon map, you can select or create a georeference that does not cover the total area of your polygon map during rasterisation.

For ILWIS 1.4 users

- During rasterisation, an .INF table is not produced anymore. To know the area of polygons, length of segments, nr. of points etc., you can calculate the histogram of a polygon, segment or point map with the Histogram operation.

7.7.2 Segments to raster

Functionality

The Segments to raster operation rasterizes a segment map. The output raster map always uses the same domain as the input segment map. This means that the class names, IDs, or values used in the segment map are also used in the raster map.

For the output raster map, an existing georeference has to be selected or a new one can be created. The georeference determines the number of lines and columns of the output map and the pixel size of the map, see also the examples in the Additional information below. It is strongly advised that vector maps of the same area are rasterized on the same georeference: any map calculation or spatial operation performed later on a combination of raster maps will only make sense if the pixels in these maps refer to the same area on the ground.

When a segment map has an attribute table or when the domain of the segment map has an attribute table, the Segments to raster operation automatically links this attribute table to the output raster map.

☞ When you want to create a Digital Elevation Model, you can directly do the operation Contour interpolation. Contour Interpolation first rasterizes your segment map, then calculates interpolated values for the pixels in between the contour lines.

Input map requirements
No special input map requirements.

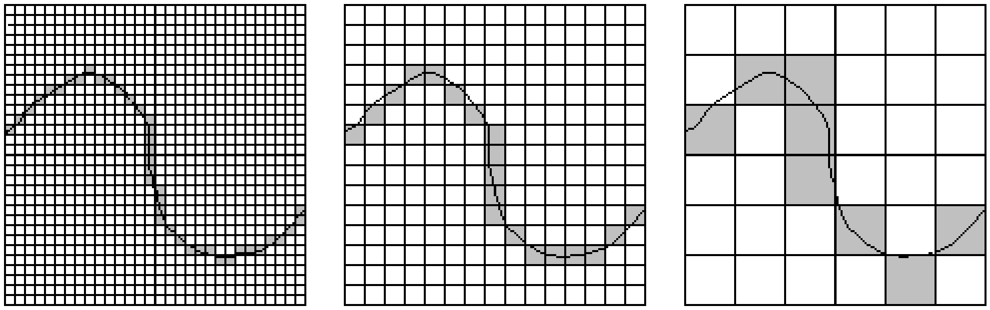
Domain and georeference of output map
The output raster map uses the same domain as the input segment map.
The georeference for the output map has to be selected or created; you can usually select an existing georeference corners. The georeference for the raster map must use the same coordinate system as the segment map. Georeference None cannot be selected for the output map.

☞ To rasterize only segments with a specific code, perform the Mask segments operation first

☞ When you want to print vector maps for annotation purposes on top of raster maps, print quality will improve if you use the vector maps as they are, thus without rasterising them.

☞ To obtain frequency information on segments, you can calculate the histogram of segment map with the Histogram operation.

Additional information
The georeference you use for the output raster map determines the pixel size of the raster map and thus whether shapes of segments are well retained. Below you find three examples of rasterized segment maps, each one with a different pixel size.



Dialog box

Dialog box options

Input segment map:	Select an input segment map. Open the list box and select the desired input map, or drag a segment map directly from the Catalog into this box.
Output raster map:	Type a name for the output raster map that will contain the rasterized segments.
Georeference:	Select an existing georeference for the output raster map; open the list box by clicking it. Or create a new georeference by clicking the little create button. You can usually select an existing georeference corners.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

☞ If you want to create a Digital Elevation Model, you can directly do the operation Contour Interpolation. Contour Interpolation first rasterizes your segment map, then calculates interpolated values for the pixels in between the contour lines.

Command line

The Segments to raster operation can be directly executed by typing the following expression on the command line of the Main window:

OUTMAP = `MapRasterizeSegment`(InputSegmentMapName, Georeference)

where:

OUTMAP	is the name of your output raster map.

`MapRasterizeSegment`	is the command to start the Segments to raster operation.
InputSegmentMapName	is the name of your input segment map.
Georeference	is the name of an existing georeference. At the moment, it is not yet possible to create a new georeference on the command line.

When the definition symbol `=` is used, a dependent output map is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output map has been calculated.

Algorithm

The Segments to raster operation rasterizes a segment map. The output raster map always uses the same domain as the input segment map. The user has to specify the georeference for the output raster map.

Process
For each segment, the corresponding pixels in the raster map are found; these pixels are assigned the class name, ID or value of the segment.
Other pixels obtain the undefined value.

- The specified pixel size in the georeference for the output map, has a large influence on the size (in bytes) of the output raster map. Mind that rasterising on a pixel size of 10 m instead of 50 m. increases the size of the raster map by a factor 25.
- If you want to rasterize only a portion of your segment map, you can either select or create a georeference that does not cover the total area of your segment map during rasterisation, or you can first create a SubMap of the segment map and then rasterize this segment submap.

For ILWIS 1.4 users

- During rasterisation, an .INF table is not produced anymore. To know the area of polygons, length of segments, nr. of points etc., you can calculate the histogram of a polygon, segment or point map with the Histogram operation.

7.8 Vectorize

7.8.1 Raster to polygons

Functionality

The Raster to Polygons operation extracts polygons from units in a raster map. The output polygon map uses the same domain as the input raster map, i.e. the class names or IDs in the input raster map will also be used for the polygons in the output polygon map. No polygons are created for pixels with the undefined value.

The polygons in the output map are derived from areas of pixels in the input raster map:

- where pixels have the same class name or ID or
- where pixels have exactly the same value (only possible via the command line).

You can choose to create polygons:
- from 4-connected pixels: areas of pixels are found where pixels with the same class name, ID or value are horizontally or vertically connected; or
- from 8-connected pixels: areas of pixels are found where pixels with the same class name, ID or value are horizontally, vertically or diagonally connected.

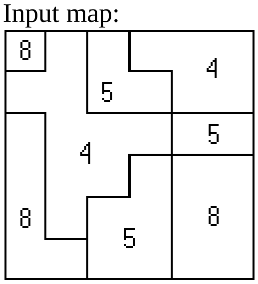
For more information on constructing areas of 4 or 8-connected pixels, see Area numbering : functionality.

Furthermore, you can specify whether or not to smooth the boundaries of the output polygons.

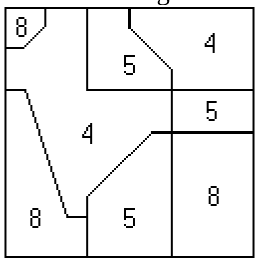
The Raster to Polygons operation attempts to create polygons for neighbouring pixels, which have the same class name or ID or exactly the same value. In an input map which uses a class or ID domain, the areas with a certain class name or ID are clearly distinct, thus, it is rather simple to find the areas. In the dialog box of this operation, you can therefore only select class or ID maps. On the command line, you can also use value maps or images as input. However, in a value map or an image, there are usually no distinct areas with exactly the same value; the values of neighbouring pixels may be similar but are usually not exactly the same. As the operation attempts to find areas where neighbouring pixels have exactly the same value, the resulting output map will usually contain very many areas; these areas may consist of individual pixels or of small groups of a few pixels. To make polygons of a value raster map or an image, it is advised to first use the Slicing operation; then, you can use the output map of the Slicing operation in the Raster to Polygons operation.

Examples
The effect of using 4 or 8-connected pixels with smoothing is illustrated in the figures below.

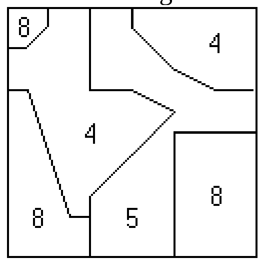
Input map:



Output map 4-connected with smoothing:



Output map 8-connected with smoothing:



The result of this operation depends on the homogeneity of the raster map and the pixel size compared to the size of the mapping units. It is advised to only use this operation on rather homogeneous raster maps that consist of areas with a considerable number of pixels. When you want to extract for instance polygons from a raster map, which is the result of the Classify operation, it may be better to first run the majority filter on that raster map to homogenize the classification results and then perform the Raster to polygons operation.

When the input raster map has an attribute table or when the domain of the input raster map has an attribute table, the Raster to polygons operation automatically links this attribute table to the output polygon map.

Input map requirements

When you use the Raster to Polygons operation through the dialog box, you can use for the input raster map a map with a class, ID or Bool domain. On the command line, you can use any type of input map.
The input raster map must have a georeference, which is not georeference `None`.

Domain and coordinate system of output map

The output polygon map uses the same domain as the input raster map.
The output polygon map uses the same coordinate system as the georeference of the input raster map. The coordinate boundaries for the polygon map are the boundaries of this georeference.

Dialog box

Dialog box options

Input raster map:	Select an input raster map (map with a class, ID, Bool or Group domain). Open the list box and select the desired input map, or drag a raster map directly from the Catalog into this box. The raster map must have a georeference, which is not georeference `None`.
Connect:	Specify to obtain polygons from 4-connected pixels or from 8-connected pixels.
4-connected:	Find areas of pixels with the same value/name, which are *horizontally or vertically connected*, then create polygons

	from these areas.
8-connected:	Find areas of pixels with the same value/name, which are *horizontally, vertically or diagonally connected*, then create polygons from these areas.
Smooth lines:	Select this check box if you want the boundaries of the output polygons to be smoothed. Clear this check box if the polygon boundaries should follow the exact boundaries of the pixel areas.
Output polygon map:	Type a name for the output polygon map that will contain the polygons extracted from units in the raster map.
Show:	Select this box if you want the output map to be displayed in a map window when the operation has finished. Clear this box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

7.8.2 Polygons to points

Functionality

The Polygons to points operation creates a point for each polygon in the polygon map. Each point obtains the class name, ID, or value of the corresponding polygon. In this way, polygon label points are created. Optionally, you can also obtain label points for polygons without class names, ID’s or values, i.e. for undefined polygons.

☞ By creating polygon label points which also contains label points for undefined polygons, you can easily find the polygons which do not yet have a correct class name, ID or value. To check whether there are any undefined polygons, you can open the output point map as a table.

☞ When you want to edit label points of undefined polygons:
- Display the polygon map and in the Display Options dialog box of the polygon map, choose Boundaries Only;
- Display the point map which contains the polygon labels in the same map window (points for undefined polygons do not appear yet);
- In the map window, open the Edit menu, choose Properties and select the point map. In the appearing Point Map Properties dialog box, click the Break Dependency Link button;
- In the map window, open the Edit menu, choose Edit Layer and select the point map;
- The point editor will be started: all points and their class names, IDs or values will be shown including the label points for undefined polygons;

- Now, you can edit the points with the undefined value to a correct class name, ID or value (depending on the domain of the map). All changes are directly stored.
- When finished, you can repolygonize the segment map using the updated label point file; the class names, IDs or values found in the label point file will be used to assign class names, IDs or values to polygons.

☞ To obtain the names, IDs or values of polygons as text within polygons, it is advised to create an annotation text object and base it on your polygon map. You can edit font, color, and position of the texts in the Annotation Text editor. For more information, see ILWIS objects: annotation text.

Input map requirements
No special input map requirements.

Domain and coordinate system of output map
The output point map uses the same domain as the input polygon map.
The output point map uses the same coordinate system and coordinate boundaries as the input polygon map.

Dialog box

Dialog box options

Input polygon map:	Select an input polygon map. Open the list box and select the desired input map, or drag a polygon map directly from the Catalog into this box. No special input map requirements.
Label points:	For each polygon, a point is created in the output map. The points are located inside the corresponding polygons.
Include undefineds:	Select this check box when the label point file should also contain points for polygons without a class name, ID or value, i.e. for undefined polygons. Clear this check box when you only want to obtain label points for polygons that have a class name, ID or value.
Output point map:	Type a name for the output point map that will contain the polygon label points.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

Command line

The Polygons to points operation can be directly executed by typing the following expression on the command line of the Main window:

OUTMAP = `PointMapPolLabels`(InputPolygonMap)
OUTMAP = `PointMapPolLabels`(InputPolygonMap, `AlsoUndefs`)

where:

OUTMAP	is the name of your output point map.
`PointMapPolLabels`	is the command to start the Polygons to points operation.
InputPolygonMap	is the name of the input polygon map.
`AlsoUndefs`	is an optional parameter to obtain label points for undefined polygons as well.

When the definition symbol `=` is used, a dependent output map is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output map has been calculated.

Algorithm

The Polygons to points operation creates a point for each polygon in the polygon map. Each point obtains the class name, ID, or value of the corresponding polygon. In this way, polygon label points are created. Optionally, you can also obtain label points for polygons without class names, ID's or values, i.e. for undefined polygons.

Label points
For each polygon:

- The Y-coordinate of a label point is determined as the mean value of the minimum and maximum Y-coordinate used by the polygon. In other words, the Y-coordinate of the label point will be positioned on the horizontal middle line of the polygon.
- In case the polygon is convex, the X-coordinate of a label point is determined as the mean of the minimum and maximum X-coordinate used by the polygon.
- In case the polygon is concave, it is possible that the horizontal middle line is cut into several pieces by the polygon boundary. Then, the X-coordinate of the label point is positioned at the middle of the longest piece of the horizontal middle line inside the polygon.

This means that:

- for circular polygons, the label point lies more or less at the center of the polygon;
- for vertical half-moon shaped polygons, the label point lies in the middle of the broader part of the polygon.
- for horizontal half-moon shaped polygons, the label point lies in one of the tips of the half-moon.
- for hour-glass shaped polygons, the label point may lie at the narrowest part of the polygon.

- for polygons containing one or more islands, the label point may lie at a narrow part close to an island.

When you are not satisfied with the position of a label point, display the polygon boundaries and the label points in the same map window, break the dependency link of the point map, and use the Point editor to move a label point to another position.

7.9 Table operations

7.9.1 Change domain of table

Functionality

The Change domain of table operation copies the contents of an input table to a new table; the new table will have another domain than the input table.

For the domain of the output table, you can choose:
- domain `None`;
- an existing class or ID domain on disk;
- a class or ID domain of a column in the input table when that column contains unique classes or IDs;
- a class or ID domain of a column in the input table where the column does not contain unique classes or IDs and other column values need to be aggregated (`average`, `minimum`, `maximum`, `sum` and `last` value encountered).

Explanation

1. When you choose to obtain an output table with domain `None`:
   - All records of the input table will be written into the output table.
   - The order of records in the input and in the output table is the same.

2. When you choose to obtain an output table with an existing class or ID domain on disk:
   - The domain items of the input table will be compared with the domain items in the selected domain; when an item (i.e. a class name or an ID) exists in both domains, then the record is written into the output table.
   - The order of records in the output table will be the order of the selected domain.
   - You may use this option: (1) when you have a table containing records for all classes or IDs in a domain, or (2) when you want to obtain a table with only a few records on a few classes or IDs of the input table. For case (2): to create a new domain with only a few classes or IDs (a subset), use the Change domain of table operation and select the domain with the subset.
   - You may also use this option after importing tables and when you know that the imported tables should use the same domain.

3. When you choose to obtain an output table which will use the domain of a column in the table and when each field of that column contains another class name or ID, i.e. when the fields in the selected column are unique:
   - For each unique class name or ID, the values found in other columns in the input table will be written into the output table.
   - The order of records in the output table will be the order of the domain of the selected column.
   - When the class names or IDs in the selected column are not unique and when you do not choose the Aggregate option, an error message will follow.

4. When you choose to obtain an output table which will use the domain of a column in the input table and when the fields of that column do not contain unique class names or IDs, choose the Aggregate option.
   - For each group of class names or IDs found in the selected column, the values found in all value columns will be aggregated by taking either the average, minimum value, maximum value, or the sum; the answers are then written into the output table. You can also choose to simply use the last value encountered per group of class names or IDs.
   - When an aggregation on an input column is not possible, i.e. when an input column does not have a value domain, the last class name, ID, color, etc. encountered per group will be written into the output table.
   - When you use the Average aggregation function, the precision of output value columns will differ from the precision of the input value columns.
   - When you use the Sum aggregation function, the value range of output value columns will differ from the value ranges of the input value columns.
   - The order of records in the output table will be the order of the domain of the selected column.
   - An extra column `Count` will appear in the output table; it contains the number of times that a certain class name or ID was found in the selected column.

Requirements for the input table
No special requirements.

Domain of output table
The domain of the output table will be the domain, which you selected: domain `None`, an existing class or ID domain on disk, or a class or ID domain of a column in the input table.
The output table will contain an extra column Count when an aggregation function is used.

☞ The Change domain of table operation can only use *one* type of aggregation function, which will be applied on *all value columns* in the input table. When you want to use different aggregation functions for various columns or when you want to use weight functions for an aggregation, it is advised to use the Join Column command on the Columns menu of the 'output' table.

Dialog box

Dialog box options

Input table:	Select an input table. Open the list box and select the desired input map, or drag a raster map directly from the Catalog into this box.
Change domain to:	Choose whether you want to change the domain of the table to a class or ID domain of a column in the table or to an existing class or ID domain on disk.
Column:	Select a class or ID column from the table; the domain of this column will be the domain of the output table.
Domain:	Select an existing class or ID domain from disk.
Aggregate:	Select this checkbox when you selected a class or ID column from the table and when the class names or IDs in that column occur more than once, i.e. are not unique. Subsequently, select an aggregation function: `average`, `minimum`, `maximum`, `sum` or `last`. The values of value columns in the input table will appear aggregated in the output table according to the selected aggregation function. Clear this check box when the classes or IDs in the selected column are unique.
Output table:	Type a name for the output table.
Show:	Select this check box if you want the output table to be displayed in a table window when the operation has finished. Clear this check box if you do not want to see this table immediately: you simply define how the output table should be created.
Description:	Optionally, type a description for the output table. The description appears in the title bar when the output table is displayed.

A dependent output table is created.

Command line

The Change domain of table operation can be directly executed by typing one of the following expressions on the command line of the Main window:

OUTTABLE = `TableChangeDomain`(InputTableName, `None`)
OUTTABLE = `TableChangeDomain`(InputTableName, DomainName)
OUTTABLE = `TableChangeDomain`(InputTableName, ColumnName)
OUTTABLE = `TableChangeDomain`(InputTableName, ColumnName, `avg` | `min` | `max` | `sum` | `last` | `no`)

where:

OUTTABLE	is the name of your output table.
`TableChangeDomain`	is the command to start the Change domain of Table operation.

InputTableName	is the name of your input table.
`None`	is a parameter to obtain an output table with domain None.
DomainName	is a parameter to specify an existing class or ID domain on disk; the output table will use this class or ID domain.
ColumnName	is a parameter to specify a class or ID column in the input table; the output table will use this class or ID domain. When you do not perform an aggregation, the specified column must contain unique classes or IDs.
`avg | min | max | sum | last | no`	when the specified column does not contain unique classes or IDs, you can use these parameters to aggregate values of value columns in the table: average, minimum value, maximum value, sum, last value or no aggregation respectively. For non-value columns in the input table, the last class name, ID, or color, etc. encountered will be used.

When the definition symbol `=` is used, a dependent output table is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output table has been calculated.

7.9.2 Table to point map

Functionality

The table to point map operation creates a point map out of a table. The table should have at least two columns, which define the X- and Y-coordinates of the points.

You can choose between the following possibilities:

- the output point map should use the same domain as the table (ID domain); the table will be linked as attribute table to the output point map;
- the output point map will use the domain of a column in the table; the output point map will have no attribute table;
- the output point map should use a new ID domain which is based on the record numbers of the table (domain None) and a user-defined prefix; the table also obtains this new ID domain and the table will be linked as attribute table to the output point map. The new ID domain will automatically obtain the same name of the output point map.

This operation is designed to obtain a point map from data from other packages and which was imported into ILWIS as a table.

Requirements for input table
The input table should have two columns, which contain the X- and Y-coordinates for the points.

Domain and coordinate system of output point map
You can choose whether the output point map should use the ID domain of an ID table, the domain of a column in the table, or whether a new ID domain should be constructed based on the record numbers in a table (domain `None`) and a user-specified prefix.

Dialog box

Dialog box options

Input table:	Select an input table. Open the list box and select the desired input map, or drag a raster map directly from the Catalog into this box.
X column:	Select a column from the table, which contains the X-coordinates of the points.
Y column:	Select a column from the table, which contains the Y-coordinates of the points.
Coordinate system:	Select an existing coordinate system for the output point map or create a new coordinate system (create button) in which the coordinates of the coordinate columns in the table fit.
Domain of output map:	Choose whether the point map should use the domain of the input table (ID domain), the domain of a column in the table, or
Use table domain:	For a table with an ID domain only, select this option when the output point map should use the ID domain of the table. The table will be linked as attribute table to the output point map.
Use record numbers as IDs:	For a table with domain `None` only, select this option when a new ID domain should be constructed from the user-specified Prefix and the record numbers of the table. The new ID domain will be used by the output point map; furthermore, the domain of the table will change from domain `None` to this new ID domain and the table will be linked as attribute table to the output point map.
Use attribute column:	Select this option when the output point map should use the domain of a column in the table. You can select a class, ID, or a value column.
Use column of table:	For a table with domain None only, select this option when the output point map should use the domain of a column in the table. You can select a class, ID, or a value column.
Domain prefix:	When using an input table with domain `None` and when you selected the option Use record numbers as IDs, type the prefix that should be used for the identifiers in a new ID domain. The new ID domain will contain identifiers

	with this prefix followed by the individual record numbers in the table.
Column:	Select a column from the table; the domain of this column will be the domain of the output point map.
Output point map:	Type a name for the output point map.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

Command line

The Table to Point Map operation can be directly executed by typing one of the following expressions on the command line of the Main window:

Obtain an ID point map with linked attribute table
(Create it from a table with an ID domain or domain `None`).

OUTMAP =	`PointMapFromTable`(InputTableName, CoordinateSystem)
OUTMAP =	`PointMapFromTable`(InputTableName, Xcolumn, Ycolumn, CoordinateSystem)

where:

OUTMAP	is the name of the output point map.
`PointMapFromTable`	is the command to start the Table from Point Map operation.
InputTableName	is the name of your input table which has an ID domain or domain `None`. When the table has an ID domain, the output point map will use this ID domain and the table will be linked as attribute table to the output point map. When the table has domain `None`, a new ID domain will be created which contains identifiers with fixed prefix `Pnt` followed by the record numbers of the table. The output point map will use this new ID domain. The domain of the table will change from domain `None` to the new ID domain, and the table will be linked as attribute table to the output point map. The new ID domain will automatically obtain the same name as the output point map.
Xcolumn	is a parameter to specify the name of the column, which contains X-coordinates for the points. When

	the input table already has a column with name X, then you do not have to use the Xcolumn parameter in the expression.
Ycolumn	is a parameter to specify the name of the column, which contains Y-coordinates for the points. When the input table already has a column with name Y, then you do not have to use the Ycolumn parameter in the expression.
CoordinateSystem	is the parameter to specify an existing coordinate system for output point map.

To obtain a point map with the domain of a column in the table

OUTMAP = `PointMapFromTable`(InputTableName, CoordinateSystem, AttributeColumn)

OUTMAP = `PointMapFromTable`(InputTableName, Xcolumn, Ycolumn, CoordinateSystem, AttributeColumn)

where:

InputTableName	is the name of your input table; the table can have any domain.
AttributeColumn	is the name of an attribute column; the output point map will have the domain of this column. The output point map will not have an attribute table.

Obtain an ID point map with linked attribute table

(Create it from a table with domain `None`)

OUTMAP = `PointMapFromTable`(InputTableName, CoordinateSystem, Prefix)

OUTMAP = `PointMapFromTable`(InputTableName, Xcolumn, Ycolumn, CoordinateSystem, Prefix)

where:

InputTableName	is the name of your input table; the table should have domain `None`.
Prefix	is the parameter to specify the prefix for a new output ID domain which will be used both by the output point map and the table. This ID domain is constructed from the user-specified prefix followed by the record numbers of the table. Furthermore, the domain of the input table will change from domain `None` to the newly constructed ID domain, and the input table will be linked as attribute table to the output point map. The new ID domain will automatically obtain the same name as the output point map.

When the definition symbol `=` is used, a dependent output map is created; when the assignment symbol `:=` is used, the dependency link is immediately broken after the output map has been calculated.

7.9.3 Glue tables

Functionality / Algorithm

The Glue tables operation allows you to glue or merge two or more tables together. As input tables, you may use:

- tables with domain `None`,
- tables with class domains,
- tables with ID domains,
- tables with class domains and ID domains.

The Glue tables operation should be regarded as a tool to combine different tables. You can for instance combine or integrate attribute tables of different years. Tables with domain `None` can also be glued vertically, one below the other.

The operation will automatically determine:

- the domain of the output table,
- the domains of the columns in the output table.

Then, fields in the input tables will be copied to the output table.

In the dialog box, you can select up to 4 input tables. On the command line, you can specify as many input tables as you like.

Process

First, the domain for the output table will be determined; the domain for the output table depends on the domains that are used by the input tables:

- when the domain of all input tables is domain None, the output table will also use domain `None`,
  - when the option Vertical is used, the number of records in the output table will be the sum of all records in all input tables,
  - when the option Vertical is not used, the output table will have the same the number of records as the largest input table.
- when all input tables have the same Class or the same ID domain, the output table will also use this class or ID domain;
- when the input tables use different class domains or different ID domains, then a new output class or ID domain will be created which contains all class names or all IDs of all input table domains; the new output domain will obtain the same name as the output table;
- when some of the input tables use class domains and other input tables use ID domains, then a new output ID domain will be created which contains, as IDs, all class names and all IDs of all input table domains; the new output domain will obtain the same name as the output table.

Then, the columns to be copied to the output table will be examined.

For input tables with a class or ID domain and for input tables with domain None (no vertical gluing):

- when in 2 or more input tables, columns are found that have the same name and the same domain, then only the column of the first input table is copied to the output table,
- when in 2 or more input tables, columns are found that have the same name but which are using different domains, then all these columns are copied to the output table; in the output table, the column of the first table will keep its original name, while the second column will obtain a 2 behind its name, etc. (e.g. MyColumn and MyColumn2) so that you can identify from which table each column was copied,
- other input columns which only exist in one table will always be copied to the output table.

For input tables with domain None and when you selected the option vertical gluing:

- when in 2 or more input tables, columns are found that have the same name and the same domain, then all values of all these columns are copied into a single output column (i.e. below each other = vertical);
- other columns are glued as above.

When a new Class or ID domain is created for the output table, this domain will obtain the same name as the output table. On the command line, you can also specify a name for the output domain yourself.

Examples

Combining tables with domain None and using option Vertical

First input table			Second input table			Output table		
	Direction	Length		Direction	Length		Direction	Length
1	0	2223	1	5	6993	1	0	2223
2	1	4737	2	6	1123	2	1	4737
3	2	2048	3	7	4273	3	2	2048
4	3	6000	4	8	1827	4	3	6000
5	4	0	5	9	1265	5	4	0
						6	5	6993
						7	6	1123
						8	7	4273
						9	8	1827
						10	9	1265

- In this example, both input tables use domain None. The output table will also use domain None.
- Because option Vertical is used, the output table contains the total number of records of all input tables.
- In this example, the columns Direction and Length occur in both input tables; in both the input tables, these columns have the same name and use the same domain. The output table will therefore contain one column Direction and one column Length.
- The values of the columns are then copied to the output table.

Combining tables with domain Class

First input table

	Landvalue80
Agriculture	75
Agriculture (irri)	125
Bare rock	10
Bare soils	10
Forest	25
Grassland	25
Lake	?
Riverbed	?
Shrubs	35
Urban center	750
Urban periphery	500

Second input table

	Landvalue90
Agriculture	100
Agriculture (irri)	150
Airport	600
Bare rock	50
Bare soils	50
Forest	75
Grassland	75
Lake	?
Riverbed	?
Shrubs	50
Urban center	1000
Urban periphery	750

Output table

	Landvalue80	Landvalue90
Agriculture	75	100
Agriculture (irri)	125	150
Airport	?	600
Bare rock	10	50
Bare soils	10	50
Forest	25	75
Grassland	25	75
Lake	?	?
Riverbed	?	?
Shrubs	35	50
Urban center	750	1000
Urban periphery	500	750

- In this example, the input tables use different domains. Therefore, for the output table, a new domain is created which contains all items of the input domains.
- In this example, columns 'Landvalue80' and 'Landvalue90' have distinct names and occur only in one input table, therefore both columns will be copied to the output table.
- The values of columns Landvalue80 and Landvalue90 are then copied to the output table.
- If each of the input tables would have contained a column 'Landvalue' (using the same domain in both input tables), then the output table would have contained only one column 'Landvalue' containing the values of column 'Landvalue' of the first input table.
- If each of the input tables would have contained a column 'Landvalue' (using the different domains in both input tables), then the output table would have contained two columns, called 'Landvalue' and 'Landvalue2'.

Domain of output table

- When all input tables use domain None, the output table will also use domain None.
- When all input tables use the same class domain or the same ID domain, then the output table will also use this class domain or this ID domain.
- When the input tables use different class domains or different ID domains, then a new class domain or a new ID domain will be created for the output table; the output domain will contain all class names or all IDs of the input domains.
- When the input tables use class domains and ID domains, then a new ID domain will be created for the output table; the output domain will contain all class names and IDs of the input domains as IDs.

When a new domain is created for the output table, the domain will obtain the same name as the output table. On the command line, you can also specify a name for the output domain yourself.

Dialog box

Dialog box options

Number of input tables:	Select the number of tables that you want to glue or merge together (2, 3, or 4). In the dialog box, the number of input tables is limited to four. On the command line, this limitation is not present.
First input table:	Select the first input table. Open the list box and select the desired input table, or drag a table directly from the Catalog into this box.
Second input table:	Select the second input table.
Third input table:	Optionally, select a third input table.
Fourth input table:	Optionally, select a fourth input table.
Output table:	Type a name for the output table that will contain all input tables.

Show:	Select this check box if you want the output table to be displayed in a table window when the operation has finished. Clear this check box if you do not want to see this table immediately: you simply define how the output table should be created.
Description:	Optionally, type a description for the output table. The description appears in the title bar when the output table is displayed.
Vertical:	In case you are merging tables that use domain None: select this check box to create an output table which will contain as many records as the total number of records of all input tables. If a column exists in more that one input table with the same name and the same domain, then the values of these columns will be glued into one output column (values one below the other = vertically). In case you are merging tables that use domain None: clear this check box when the output table should contain the same number of records as the first input table.

A dependent output table is created. When the input tables use different class or ID domains, a new domain will be created for the output table. This output domain will contain all class names and/or IDs of the input domains and the domain will obtain the same name as the output table.

Command line

The Glue tables operation can be directly executed by typing one of the following expressions on the command line of the Main window:

OUTTABLE =	`TableGlue`(FirstInputTableName, SecondInputTableName [, MoreInputTables])
OUTTABLE =	`TableGlue`(FirstInputTableName, SecondInputTableName [, MoreInputTables] , `Vertical`)
OUTTABLE =	`TableGlue`(NewDomain, FirstInputTableName, SecondInputTableName [, MoreInputTables])

where:

OUTTABLE	is the name of your output table.
`TableGlue`	is the command to start the Glue tables operation.
FirstInputTableName	is the name of the first input table.
SecondInputTableName	is the name of the second input table.
MoreInputTables	optionally, you can specify more input table names, delimited by commas.
`Vertical`	in case the input tables use domain `None`, an optional parameter to specify that the input tables should be
NewDomain	in case of merging two or more Class or ID tables that

do not have the same domain: an optional parameter to specify a name for the new output domain in which all input domain items will be merged.
When the Class or ID domains of input tables are not the same and this parameter is not specified, a new output domain will be automatically created with the same name as the output table.

On the command line, you can specify as many input tables as you like, i.e. you can merge as many tables as you like. When using the dialog box of this operation, only two, three or four input tables can be merged at a time.

When the definition symbol = is used, a dependent output table is created; when the assignment symbol := is used, the dependency link is immediately broken after the output table has been calculated. When tables are merged which use different class or ID domains, a new domain will be created for the output table.

Chapter 8

How to…

Note: This chapter is written as an enhancement to Chapter 8: How to… in the ILWIS 2.1 Reference Guide. It does *not* completely replace the mentioned chapter: instead you are invited to use both as one.

8.1 How to add…

8.1.1 How to add annotation to a map window

Simple annotation can be added to a window in the form of texts, legends, boxes, North arrow, scale bar, grid lines, graticule or bitmaps or metafiles/pictures from disk. Further, any picture or bitmap that was copied to the clipboard from other Windows application programs can be pasted into ILWIS. You can also paste the contents of another ILWIS map window into a current map window; for instance as an inset-map giving an overview of a larger area.
This type of annotation can be edited in the Annotation editor. To store this type of annotation, you need to save the map window as a map view.

Furthermore, you can create an Annotation Text object when you want to store multiple texts at multiple positions. When creating an Annotation text object, you can base the texts that will appear in the text object on an existing point, segment, or polygon map. If you do so, the text object will contain a text (class name, ID or value) for each point, segment or polygon in the selected map.
An Annotation Text object can be edited with the Annotation Text editor or in a table window. In the Annotation Text editor, you can easily insert more text items, change and refine the position of texts (move), make text duplicates, and specify fonts, font sizes, appearance (bold, italics, underline), colors, rotations, etc. for (multiple) selected texts.

Annotation is generally used on top of one or more data layers (displayed maps).

To add an annotation text object to a map window

- From the Layers menu in a map window, choose Add Annotation, and select Text Layer from the cascading menu.
  The Add Annotation Text dialog box will appear in which you can select an existing annotation text object or create a new annotation text object.
  When you select an existing annotation text object, it will be added to the map window.

When you create a new annotation text object, the Annotation Text editor will be opened.

- From the Layers menu in a map window, choose Layer Management.
  In the appearing Layer Management dialog box, click the Add Annotation button.
  In the appearing Add Annotation (from Layer Management) dialog box, select Text Layer.
  The Add Annotation Text dialog box will appear as described above.
- Drag an existing annotation text object from the Catalog to the map window.
  The annotation text object will be added to the map window.

To add a simple annotation to a map window

- From the Layers menu in a map window, choose Add Annotation, and select Single Text, Legend, Grid Lines, Graticule, Box, Scale Bar, North arrow or Bitmap or Picture from the cascading menu.
  Except for bitmaps and pictures, you have to specify some display options for the annotation layer before the annotation is shown. The newly added annotation is displayed at the center of the map window; you are now in the Annotation editor.
- From the Layers menu in a map window, choose Layer Management. In the appearing Layer Management dialog box, click the Add Annotation button.
  In the appearing Add Annotation (from Layer Management) dialog box, select Single Text, Legend, Grid Lines, Graticule, Box, Scale Bar, North arrow or Bitmap or Picture.
  Proceed as in previous paragraph.
- From the Edit menu in a map window, choose Annotation. The Annotation editor is opened immediately. You can add simple annotation in the Annotation editor as described below.

Annotation Text editor

In the Annotation Text editor, you can:

- insert new texts and delete selected existing texts,
- move one or more selected texts to another position,
- make text duplicates, and
- specify fonts, sizes, colors, etc. for (multiple) selected texts.

For detailed descriptions, refer to Annotation Text editor: functionality.

Annotation editor

To add annotation from within the Annotation editor:
When you have already added simple annotation and you are in the Annotation editor, you can use the same possibilities as above but you can also:

- click one of the add annotation buttons in the button bar of the Annotation editor (the status line gives an explanation), or
- click the right mouse button in the map window, and choose Add Annotation from the context-sensitive menu.

Except for bitmaps and pictures, a display options dialog box follows.

To add bitmaps or pictures from disk to a map window

- use one methods described above, choose Bitmap or Picture,
- in the appearing Add BitMap or Picture dialog box, select a Bitmap (.BMP) or Picture/Metafile (.WMF) from disk.

The bitmap or picture is displayed at the left top corner of the map window; you are now in the Annotation editor.

To add an ILWIS map as an inset-map to another map window

- display the map you want to use as inset-map,
- from the Edit menu in this map window, choose Copy,
- display the map in which you want the inset map to appear,
- from the Edit menu in that map window, choose Paste.

The inset-map is displayed at the center of the map window; you are now in the Annotation editor.

To add bitmaps and pictures via Clipboard to a map window

- Open a Windows application program (for example Paintbrush),
- create a bitmap, picture, logo, graph, laid-out text, other graphics, etc.
- copy this to the clipboard,
- go to an ILWIS map window,
- from the Edit menu, choose Paste.

The bitmap, picture, text or symbol is displayed at the left top corner of the map window; you are now in the Annotation editor.
For more information, see Copying and Pasting through the clipboard.

☞ Before adding annotation to a map window, you might want to increase the size of the map window, in order to have space for a title (text), the legend, etc. To achieve this, you can select Extend Window from the Options menu in the map window.

☞ The Annotation editor allows for the positioning and sizing of annotation, and for the insertion of new annotation.

☞ When finished adding and editing simple annotation and when you want to keep this annotation, you need to save the map window as a map view: in the Annotation editor or in the map window, open the File menu and choose the Save View or the Save View As ... command. For more information, see How to save annotation.

8.1.2 How to add coordinates to an image etc.

To add coordinates to a satellite image, to a scanned map, or to a scanned photograph when you do not have a Digital Terrain Model (DTM), create a georeference tiepoints. A georeference stores the relation between locations in the image (row,col) and real world coordinates (X,Y). These locations are called tiepoints or ground control points. A georeference uses a coordinate system.

- In case you have a scanned photograph without fiducial marks and you have a DTM of the area, create a *georef direct linear*.
- In case you have a scanned aerial photograph with fiducial marks and a DTM, create a *georef orthophoto*.

For more information on types of georeferences, see ILWIS objects : georeferences.

The remainder of this topic deals with *georeferences tiepoints*.

The topic contains the following paragraphs:

1. Introduction.

2a. Creating a georef tiepoints while working with an interactive color composite, a single band of a satellite image, a scanned map or a scanned photograph.

2b. Creating a georef tiepoints while working with a permanent color composite.

3. Adding tiepoints.
4. Adding the created georef tiepoints to all bands of a satellite image.
5. Optional resampling to a georef corners.

1. Introduction

By creating and adding tiepoints to a color composite, one band of a satellite image, a scanned map or a scanned photograph, the tiepoint coordinates are **added** to the image, photo or map which was specified as background map.

When you were successful in creating a georef tiepoints for a color composite, a band of a satellite image, a scanned map or a scanned photograph, you can:

- see the coordinates according to the created georeference tiepoints on the status line of the map window;
- display any type of vector data on top of the map;
- create new vector data using the georeferenced image as background;
- update vector data using the georeferenced image as background;
- use the map in pixel info;
- rasterize any vector data on this georeference (for map calculations);
- you can also resample the image which has a georeference tiepoints to a georeference corners, or vice versa, in order to perform raster operations in which raster maps with different georeferences need to be combined;
- screen digitize on satellite images or on scanned photographs which have a georef tiepoints.

2a. Creating a georef tiepoints while working with an interactive color composite, a single band of a satellite image, a scanned map or a scanned photograph

- Display in a map window:
  - an *interactive* color composite (create a map list of *all* bands of an image and double-click the map list in the Catalog or use the Show Map List as Color Composite option), or
  - one of the bands of your satellite image, or
  - a scanned map, or
  - a scanned photograph.
- Choose the Create Georeference command from the File menu in the map window. In the subsequent Create Georeference dialog box, type a name for the georeference, choose for a Georef Tiepoints and select a coordinate system. When you click the OK button, the Georeference Tiepoints editor is automatically started in the map window.

You can now add tiepoints to the georeference tiepoints: continue with step 3.

2b. Creating a georef tiepoints while working with a permanent color composite

When your graphics board cannot be configured to use more than 256 colors (see Display Settings in Windows' Control Panel), you need to create a *permanent* color composite with the Color composite operation. A *permanent* color composite is available in the Catalog and is stored on disk.

- Create a permanent color composite (from the Operations menu in the Main window, choose Image Processing and Color Composite);
- Open the Properties dialog box of the color composite;
- In the Raster Map Properties dialog box, click the Break Dependency Link button (if available);
- Show the map in a map window, choose Create Georeference command from the File menu in the map window, In the subsequent Create Georeference dialog box, type a name for the georeference, choose for a Georef Tiepoints and select a coordinate system. When you click the OK button, the Georeference Tiepoints editor is automatically started in the map window.

You can now add tiepoints to the georeference tiepoints: continue with step 3.

Other methods

When you create a georeference tiepoints through the File menu of the Main window, or through the Edit Properties dialog box of a raster map, you have to open the Georeference Tiepoint editor yourself, for example by double-clicking the newly created georeference tiepoints in the Catalog.

For more information, see How to create a georeference.

3. Adding tiepoints

You have to add tiepoints or ground control points to the image or scanned photograph, which is displayed in the map window of the Georeference Tiepoints editor. Each tiepoint establishes a relation between the row and column value in the map and an XY-coordinate.

In the Georeference Tiepoints editor, you can add tiepoints to a map in several manners:

- first, click at a recognizable point in the map without coordinates,
- the Add TiePoint dialog box appears. In this dialog box, the row and column values at the position of the click are filled out. When you already have some tiepoints, the dialog box will also come up with a suggestion for the XY coordinates.

Then:

- click at the same position in a map which already has correct XY coordinates and which is displayed in another map window (master/slave), or
- digitize the same point in an analog paper map on a digitizer, or
- read the correct XY coordinates for this point from an analog paper map or a table, and type these XY coordinates in the dialog box.

For a more detailed description on inserting tiepoints (for instance Master/Slave), refer to How to use the Georeference Tiepoints editor.

Furthermore, for a georef tiepoints, you have to select a transformation method: conformal, affine, second order, full second order, third order or projective. The number of tiepoints that are required depends on this selected transformation method.

- For satellite imagery, an affine transformation will usually do.
- For scanned photographs, a projective transformation is recommended.

☞ To judge the quality of your tiepoints while entering them in the Tiepoint editor:

- display any existing vector data on top of the background map in the Tiepoint editor (choose Add Data Layer from the Layers menu),
- add tiepoints to the image or photograph in the map window, and
- press the Redraw button .

You can now easily see whether existing roads, rivers, etc. coincide with the background map.

4. Editing properties of satellite bands

You have to make sure, that all maps, which exactly fit on each other also use the same georeference. You can change the georeference of a map in the properties dialog box of the map.

Thus, when you located the tiepoints:

- on only *one* of the bands of an image (as described under 2a), or
- on a *permanent* but a *non-dependent* color composite (as described under 2b), you have to add the created georef tiepoints to *all (other) bands of the image*.

After you have finished adding tiepoints to the background map in the Tiepoint editor:

- Open the Properties dialog boxes of *all* other bands of the image;
- Click the Edit Properties button and select the correct georef tiepoints.

See also How to view and edit properties of an object.

In case you have create a georef tiepoints (as described under 2a) for:

- an interactive color composite, then the created georef tiepoints will be automatically linked to all maps in the map list. You do not need to edit properties of the individual maps in the map list.
- a scanned map or a scanned photograph, then the created georef tiepoints be automatically linked to the scanned map or photo. You do not need to edit properties.

5. Optional resampling to a georef corners

When maps use a georef tiepoints, you can decide to resample these maps to the georeference of another existing north-oriented map to be able to combine the maps with each other in MapCalc, etc. For more information, see Resampling : functionality.

8.2 How to calculate…

8.2.1 How to calculate a confusion matrix

To access the accuracy of an image classification, it is common practice to create a confusion matrix. In a confusion matrix, your classification results are compared to additional ground truth information. The strength of a confusion matrix is that it identifies the nature of classification errors, as well as their quantities.

To obtain a confusion matrix

1. Have the output raster map of your image classification available; check the Properties of the classified image to know which domain and georeference are used by the classified image.
2. Create a raster map, which contains additional ground truth information, such a map is also known as the test set. There are several ways to create a test set, these are shortly described here; at the end of this topic you will find more details of creating a test set.
   - Create a new raster map with the same domain and georeference as the classified image, and add some test pixels to this map yourself with the pixel editor while using an existing map as a background (f.e. a land use map),
   - Use an existing, recent and reliable raster map with the same domain and georeference as the classified image directly as a test set.
   - Use half of your ground truth data for the sample set before classification, and use the other half of your ground truth data for creating a test set.

   Make sure that the ground truth/test set raster map does not contain the same pixels as the sample set raster map from the training phase. Your accuracy assessment will show too optimistic figures when the pixels in the sample set (on which the classification is based) are also used in the test set (with which the classification results are checked).

3. Perform a Cross operation with your ground truth map and the classified image to obtain a cross table.
   To start the Cross operation:
   - from the Operations menu in the Main window, select Raster Operations and then the Cross command, or
   - double-click the Cross item in the Operation-list, or
   - use the right mouse button on your test set raster map in the Catalog and from the context-sensitive menu select Raster Operations and Cross.

   In the Cross dialog box:
   - for First Map, select the test set map,
   - accept the default selection of the Ignore Undefs check box for the first map,
   - for Second Map, select the classified image,
   - clear the Ignore Undefs check box for the second map,
   - type a name for the output cross table, and
   - select the Show check box to directly display the output cross table in a table window.

4. In the table window displaying the cross table, open the Options menu and choose Confusion matrix.
   In the Confusion matrix dialog box:
   - for First Column, select the test set column in the table,
   - for Second Column, select the classified image column in the table,
   - for Frequency, select the NPix column in the table.

When you click OK, the confusion matrix is displayed in a secondary window.

Additional information to create a test set

1. Creating a test set yourself with the pixel editor using a background map:
   - Check the domain and georeference of the classified image in the Properties dialog box of the image. Check the georeference of the background map, which can be for instance an existing land use map. If the background map does not have the same georeference as your classified image, you can use the Resample operation to resample the existing map to the georeference of the classified image.
   - Display the background raster map in a map window. It is advised to select the Legend check box in the Display Options dialog box of this background map.
   - From the File menu in the map window, choose the Create Raster Map command to create an empty ground truth map. In the Create Raster Map dialog box: type a new name for the ground truth map; accept the georeference of the classified image; and select the same domain as used by the classified image,
   - The pixel editor is opened automatically; the background map is displayed. When you zoom in, you can start selecting and giving names to pixels in your ground truth map.

2. Using an existing raster map or polygon map of the area:
   When you have a recent and reliable polygon or raster map of the area, you can directly use this map as the ground truth map. If the map does not have the same domain as the classified image, you can add a column (with the domain of the classified image) to the attribute table of the map, assign a class to each class in the map and create an attribute map. If the map does not have the same georeference as the classified image, you can either rasterize the polygon map on the georeference of the image, or resample the raster map to the georeference of the image. Then, you can directly cross the raster map with the classified image.

3. When you have collected ground truth data in the field, a calculation-wise correct method is to use half of these data for the sample set, and the other half for the test set.

8.2.2 How to calculate a Digital Elevation Model

In ILWIS, you can obtain a Digital Elevation Model (DEM) by interpolation of segments, by an interpolation of points, or by an interpolation of a raster map which contains rasterized value segments and points.

From Segments

Digitize the contour lines of a segment map with a value domain. Make sure that contour lines that consist of more than one segment have the same value. Code consistency of segments can be checked in the Segment editor.

Then perform a Contour interpolation. Contour interpolation is an operation which first rasterizes the contour lines in a segment map with a value domain, and then calculates values by means of a linear interpolation for pixels that are not covered by segments. For the Contour interpolation operation, you can also use a segment map with a class or ID domain as input, then the segment map must have an attribute table with a value column representing height values.

To create a Digital Elevation Model from Segments

- from the Operations menu in the Main window, choose Interpolation, Contour Interpolation, or
- double-click the InterpolSeg operation in the Operation-list, or
- use the right mouse button on a segment map in the Catalog and select Contour Interpolation from the context-sensitive menu.

The Contour Interpolation dialog box appears.

☞ When later on you want to create a georeference orthophoto or a georeference direct linear, which will use the created DEM, it is strongly advised to mark the Interpolation check box in the Raster Map Properties dialog box of the DEM.

From Points

Perform a point interpolation on your point map. A point interpolation performs an interpolation on randomly distributed point values and returns regularly distributed point values. This is also known as gridding. In ILWIS, the output values are raster values.

The input map for a point interpolation is a point map where:

- points are values (point map with a value domain), or
- points are identifiers (point map with an ID domain) and for which elevation values are stored in a column in a linked attribute table.

To create a Digital Elevation Model from Points

- from the Operations menu in the Main window, choose Interpolation, Point Interpolation, Moving Average or Moving Surface, or
- double-click the MovAverage or MovSurface operation in the Operation-list, or
- use the right mouse button on a point map in the Catalog and select Interpolation, Moving Average or Moving Surface.

The Moving Average dialog box or the Moving Surface dialog box appears.

☞ Point interpolation is a time consuming operation, to speed up the process the user is advised to use a relatively large pixel size during the interpolation and perform the Densify operation after the interpolation. Densify is a raster operation which lets you reduce the pixel size of your map. The number of rows and columns is increased, and new values are assigned to the pixels by means of a bilinear or bicubic interpolation.

☞ When later on you want to create a georeference orthophoto or a georeference direct linear, which will use the created DEM, it is strongly advised to mark the Interpolation check box in the Raster Map Properties dialog box of the DEM.

From Segments with additional Point data

1. Digitize the contour lines of a segment map with a value domain. Rasterize the segment map with the Segments to Raster operation:
   - from the Operations menu in the main window, choose Rasterize, Segment to Raster,
   - double-click the SegRas operation in the Operation-list, or
   - use the right mouse button on the segment map in the Catalog and select Rasterize, Segment to Raster from the context-sensitive menu.

2a. If you have a point map with a value domain, rasterize the point map with the Points to Raster operation:
   - from the Operations menu in the main window, choose Rasterize, Point to Raster,
   - double-click the PntRas operation in the Operation-list, or
   - use the right mouse button on the point map in the Catalog and select Rasterize, Point to Raster from the context-sensitive menu.

2b. If you have a point map that contains points with identifiers (point map with an ID domain) for which elevation values are stored in a column in a linked attribute table,
   - rasterize the point map with the Points to Raster operation (see 2a) and in the Rasterize Point Map dialog box, select the correct column in which your elevation data is stored.

   Use the same georeference for both the rasterized segment and rasterized point map.

3. Then, combine the two raster maps by typing the following MapCalc expression on the command line of the Main window:
   pntseg = `iff(isundef(`*pntras*), *segras*, *pntras*)

in which:
pntseg is the output raster map name of the combined point and segment map
pntras is the name of the rasterized point map
segras is the name of the rasterized segment map

4. Finally, interpolate the pntseg raster map, which contains the rasterized segments and the rasterized points. Type on the command line of the Main window:

   DEM = `mapinterpolcontour`(*pntseg*)

Visualize a DEM

To visualize a DEM, you can simply the DEM in a map window or apply the `Shadow` filter on the DEM and display the filtered map in a map window; you can overlay the contour lines in one color. Further, you can use Display 3D to create a three-dimensional view of your study area. For more information and an example, see How to display Digital Elevation Models.

8.2.3 How to calculate the Optimum Index Factor (OIF)

To obtain multivariate statistical information on a data set, you can calculate the Optimum Index Factor (OIF) for any combination of three bands within your map list. The Optimum Index Factor (OIF) is a statistic value that can be used to select the optimum combination of three bands in a satellite image with which you want to create a color composite. The optimum combination of bands out of all possible 3-band combinations is the one with the highest amount of 'information' (= highest sum of standard deviations), with the least amount of duplication (lowest correlation among band pairs).

To calculate and show the Optimum Index Factor

1. Create a map list, which contains the multi-spectral bands of your satellite image. For more information, refer to How to create a map list.
2. Calculate a variance-covariance matrix or a correlation matrix for the maps in the map list:
   - in the Main window, open the Operations menu, choose Statistics, Map List and the Variance-Covariance or Correlation command, and select a map list, or
   - double-click the VarCovMat or the CorrMat operation in the Operation-list, and select a map list, or
   - press the right mouse button on the map list in the Catalog, choose Statistics and Variance-Covariance or Correlation from the context-sensitive menu.

   For more information, refer to Variance-covariance matrix or to Correlation matrix.
3. Open the Properties of the map list:
   - from the Edit menu in the Main window, choose Properties, and select the map list, or
   - press the right mouse button on the map list in the Catalog, and choose Properties from the context-sensitive menu.

The Map List Properties dialog box appears.

4. Click the Additional Info button in the Map List Properties dialog box.

A dialog box appears which displays the ranked OIF values and corresponding band combinations. For more information on the interpretation the OIF values, refer to Optimum Index Factor: functionality / algorithm.

Note: The Additional Info button will not appear in the Map List Properties dialog box when a Variance-covariance matrix or a Correlation matrix has not been calculated yet for the map list.

8.3 How to calculate (Advanced)…

8.3.1 How to calculate a cross-section through the terrain

This topic describes how you can make a cross-section through (a part of) your study area. In general, the procedure is as follows:

1. Choose the location of the cross-section and make raster map which only shows the pixels of the cross-section.
   You can also screen-digitize one segment along which you want to obtain a cross-section.
2. Convert the raster map to a points map, or convert the segment map to a point map.
3. Open the point map as a table and perform a calculation on the point data so that for the coordinates of all points, the height information is retrieved from a DEM.
4. Finally, make a graph to show the cross-section.

Choosing the location of the cross-section
Open some satellite images and raster maps. You can use for instance the TMbands or the Geology or Landuse maps from the Cochabamba area. Furthermore, it is assumed that you already created a DEM from the area.

Column 165 in the Geology map seems to cover many different terrain units and is chosen as the line along which the cross-section should be made. To make a raster map showing the cross-section, use the formula:

```
crosssec = IFF(%C=165, geology,?)
```

In this expression, you specify with `%C` the column number along which you want to obtain the geology class names.
Raster map `crosssec` only shows the geological classes along column 165; all other pixels of this map are undefined.

Convert the raster map to a point map
Drag raster map crosssec from the Catalog to the RasPnt operation in the Operation-list. The Raster to Points dialog box appears:

- Fill out the name of the output point map; you can use map name `crosssec` again as the extension of a point map (.MPP) is different from the extension of a raster map (.MPR).
- Furthermore, you can mark the Show check box.

The output point map will be calculated.

In the Display Options dialog box of the point map, accept the default representation and click on OK. The point map will appear on the screen.

(When you screen-digitized a segment along which you want to obtain the cross-section, use the Segments to Points operation. The operation extracts points at regular distances along the segment; you have to specify the distance between the subsequent point coordinates (e.g. 20m).)

Open the point map as a table and performing calculations in the point table
To open the point map as a table, click point map `crosssec` in the Catalog with the right mouse button, and choose Open as table from the context-sensitive menu.

The table shows all points. Columns X and Y are the XY-coordinates of the points and column Name shows the geology class of each point.

	X	Y	Name
21	798770.00	8090110.00	Shales
22	798770.00	8090090.00	Shales
23	798770.00	8090070.00	Shales
24	798770.00	8090050.00	Shales
25	798770.00	8090030.00	Shales
26	798770.00	8090010.00	Shales

When you would like to retrieve land use information for all points, i.e. to find out about the land use type at each point coordinate, you can type the following formula on the command line of the table window:

```
Landuse = MAPVALUE(landuse,COORD(X,Y))
```

For each XY-coordinate in the point table, the land use class as found in raster map Landuse at that coordinate will be retrieved; this is stored in column Landuse. Accept the defaults in the Column Properties dialog box and click OK.

The new column is calculated and appears in the table.

	X	Y	Name	Landuse
629	798770.00	8077950.00	Lake deposits	Agriculture
630	798770.00	8077930.00	Lake deposits	Agriculture
631	798770.00	8077910.00	Lake deposits	Agriculture
632	798770.00	8077890.00	Lake deposits	Agriculture
633	798770.00	8077870.00	Lake deposits	Urban periphery
634	798770.00	8077850.00	Lake deposits	Urban periphery

In the same way, height information can be retrieved for each point coordinate. You can use the same MAPVALUE function, but this time you will information from raster map DEM:

```
Altitude = MAPVALUE(dem,COORD(X,Y))
```

For each XY-coordinate, column Altitude contains the elevation values as found in the Digital Elevation Model.

	X	Y	Name	Landuse	Altitude
21	798770.00	8090110.00	Shales	Grassland	4260.0
22	798770.00	8090090.00	Shales	Grassland	4257.8
23	798770.00	8090070.00	Shales	Grassland	4256.7
24	798770.00	8090050.00	Shales	Grassland	4254.4
25	798770.00	8090030.00	Shales	Grassland	4253.0
26	798770.00	8090010.00	Shales	Grassland	4252.7

Furthermore, you can see that the difference between each subsequent Y-coordinate is 20m; this is the pixel size of the original raster map from which you extracted the points.
To make a distance axis in kilometers for the graph, you can use the following expression:

```
Distance = %r * 20 / 1000
```

In this expression, `%r` depicts the record numbers; these are multiplied by 20m and divided by 1000. The answers are stored in column Distance.

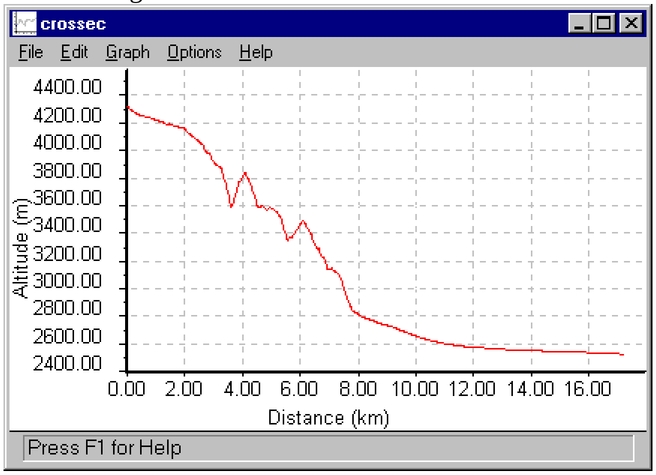
	X	Y	Name	Landuse	Altitude	Distance
21	798770.00	8090110.00	Shales	Grassland	4260.0	0.420
22	798770.00	8090090.00	Shales	Grassland	4257.8	0.440
23	798770.00	8090070.00	Shales	Grassland	4256.7	0.460
24	798770.00	8090050.00	Shales	Grassland	4254.4	0.480
25	798770.00	8090030.00	Shales	Grassland	4253.0	0.500
26	798770.00	8090010.00	Shales	Grassland	4252.7	0.520

(When you screen-digitized a segment from which you extracted points, then multiply the record numbers with the distance value that you specified in the Segments to Points operation, and divide by 1000 if you want to obtain values in kilometers.)

Displaying the cross-section as a graph
Now you can display the cross-section by making a graph, which shows the relief values along the distance values. From the Options menu in the table window, choose Show Graph.
The Edit Graph dialog box appears:
- For the X-axis, select: Distance.
- For the Y-axis, select: Altitude.

Accept other defaults, click OK, and the graph is displayed on the screen.

In this case, you can also choose column Y for the X-axis of the graph, because the points were extracted along a column of a north-oriented raster map; in the point table you can see that X-coordinate values are constant and only Y-coordinate values change.



The graph shows the relief along the cross-section. The first point lies high in the mountains on an altitude of around 4300 m while the end of the cross-section is in a relatively flat area around 2500 m altitude.

8.4 How to change…

8.4.1 How to change the domain of a map

This topic contains five sections to change the domain of a map:

- from system domain `Value` to a user-defined value domain,
- from the Image domain to system domain `Value`,
- from a Class domain to an ID domain or vice versa,
- from a Picture domain to a Class domain,
- from a Class or ID domain to another Class, ID, or to a value domain.

From system domain `Value` to a user-defined value domain

When a map uses system domain `Value`, the map can only be displayed with a *representation gradual* (boundaries in percentages). For intermediate calculation results, this may be fine. For a final map however, you may want to create a *representation value* (boundary values in values) so that you can assign specific ranges of colors to specific ranges of *values* in the map. To be able to create a representation value, the map with system domain `Value` needs to be converted into a map with a user-defined value domain.

The *legend* of a map using system domain `Value` or a user-defined value domain will in both cases show values. Thus it is only necessary to change the domain of a map with system domain `Value` when you want to assign specific colors to specific values in the map, instead of to percentages.

To convert a map (e.g. `Map1`) with system domain `Value` to a map (e.g. `Map2`) with a user-defined value domain:

- Type a simple MapCalc formula on the command line of the Main window, like `Map2 = Map1`
- In the Raster Map Definition dialog box, check the minimum and maximum values as stated under Value Range and Precision which would be used for the output map if you would accept the default system domain `Value` again;
- However, instead of accepting system domain `Value` in the Raster Map Definition dialog box, click the create domain button in the Raster Map Definition dialog box to create a new value domain;
- In the appearing Create Domain dialog box, type a new name for your new value domain and select the Value option button.
  Click the OK button in the Create Domain dialog box.
- Then, in the appearing Edit Domain Value dialog box, type the minimum and maximum values as well as the precision for this new value domain. You can use the value range and precision as found in the Raster Map Definition dialog box, widen the value range and/or increase or decrease the precision. In any case, specify a value range and precision that fit the values in your map.
- Click the OK button in the Edit Domain Value dialog box.
- The Raster Map Definition dialog box will be activated again. The newly created domain appears in the Domain list box.
- In the Raster Map Definition dialog box, type values for the value range and precision for the output map; these can be the same values as used in the Edit Domain Value dialog box.
- Click the OK button in the Raster Map Definition dialog box.

To create a user-defined representation for your map (`Map2`) with the user-defined value domain:

- Display Map2 and in the map's Display Options dialog box, click the create representation button. The Create Representation dialog box will appear: type a name and optionally a description for your new representation value.
  Click the OK button in the Create Representation dialog box.
- The Representation Value editor will be opened and you can see that values of your domain are used for the limits, i.e. the boundary values.
  Insert limits, choose or create colors for each limit, specify whether to stretch colors in between limits or use the color of the upper or lower limit, and in case of stretching colors, specify the number of stretch steps.
- When finished, click the OK button in the Display Options dialog box. You do not have to close the Representation Value editor window.
- You can continue to edit your representation value, in the Representation Value editor and redrawing the map in the map window.
- When you already closed the Representation Value editor, you can open it again by selecting the Representation command from the Edit menu in the map window. Edit the representation value and redraw the map.

- When you want to create another representation value for the map, choose Display Options from the Layers menu. In the appearing Display Options dialog box of the map, click the create representation button.

From domain `Image` to system domain Value

You can use a very simple MapCalc statement; type on the command line something like:

```
mapval = mapimg
```

where `mapimg` is the name of your image and `mapval` is the name of the output map.
In the Raster Map Definition dialog box that follows directly after you typed the statement, select system domain `Value`, and specify minimum `0`, maximum `255`, precision `1`.

From a Class domain to an ID domain or vice versa

In some cases you may want to change from classes to IDs or vice versa.

- In the Catalog, click a map with the right mouse button, and choose Properties. In the map's properties dialog box, check which domain the map is using.
- In the Catalog, click the domain with the right mouse button, and choose Properties.
- In the Domain Properties dialog box, click the button Convert to Classes or the button Convert to Identifiers.

Mind: when converting classes to IDs, you should keep in mind that an ID domain does not use a representation.

From a Picture domain to a class domain

After import, you may find that a raster map has a Picture domain. The colors of the map seem OK but when you want to perform calculations with this map, it is strongly advised to first convert the domain of the map from a Picture domain to a Class domain.

- In the Catalog, click the raster map with the right mouse button, and choose Properties.
- In the Raster Map Properties dialog box, check which Picture domain the map is using.
- In the Catalog, click the Picture domain with the right mouse button, and choose Properties.
- In the Domain Properties dialog box, click the button Convert to Classes.

The domain will contain as many classes as you had colors before. The classes will obtain default class names, like Class1, Class2, etc. In the domain Class/ID editor, you can change these default class names. The colors of the Picture domain will now be available in the representation of the Class domain.

From a Class or ID domain to another Class, ID, or value domain

In short, you have to add a column with the correct domain to an attribute table of the original map and then you have to create an attribute map of the original map.

When your original map does not have an attribute table yet:

- In the Catalog, click a map with the right mouse button, and choose Properties.
- In the map's Properties dialog box, select the Attribute Table check box, and click the create button.
- A table will be created with the same domain as the map and the table will be displayed in a table window.
- Proceed with the section below.

When your original map already has an attribute table:

- Add a column to the table and select for this column the correct domain.
- Fill the column with values, classes or IDs as you like:
  - the domain of the table (gray) shows the current class names or IDs in your map;
  - the class names, IDs or values in your column represent the correct class names, IDs, or values that you want to assign to the map.
- In the Catalog, click the map with the right mouse button and choose Raster Operations, Attribute Map (when you are working on a raster map) or Vector Operations, Attribute Map (when you are working on a polygon, segment or point map).
- In the appearing dialog box Attribute map of Raster/Polygon/Segment/Point map,
  - the name of the map is already filled out,
  - the attribute table currently linked to this map is already filled out,
  - select the attribute column in which you filled out the correct class names, IDs, or values
  - fill out the rest of the dialog box, and click OK.

  The output map will by default use the same domain as the column you selected from the table.

For more information, see the Attribute map of raster map, Attribute map of polygon map, Attribute map of segment map, Attribute map of point map.

8.5 How to create…

8.5.1 How to create a color composite

A color composite is created by combining three raster images (bands/maps). One map is displayed in shades of red, one in shades of green and one in shades of blue. Putting the three bands together in one color composite can give a better visual impression of the reality on the ground, than displaying one band at a time.

There are three types of color composites: natural color composites, pseudo natural color composites and false color composites. Natural color composites are made of the green, blue and red part of the spectrum. This results in an image with realistic colors. A pseudo color composite is created with other parts of the spectrum, but the result has natural looking colors. In false color composites the colors in the image are not the same as in reality: for instance, healthy vegetation is red and urban areas are blue.

In ILWIS, there are two ways in which you can display or create a color composite:

- by showing a map list as a color composite; the resulting color composite is displayed in a map window (interactive color composite), and
- by using the Color composite operation; the resulting color composite is stored as a raster map on disk (permanent color composite).

Interactive color composites

An interactive color composite can be created when your graphics board is configured to use more than 256 colors, for instance High Color 16-bit, or True Color 24-bit (see Display Settings in Windows' Control Panel).

To use an interactive color composite:

- from the Operations menu in the Main Window, choose Visualization, Show Map List as Color Composite, or
- double-click the ColorComp item in the Operation-list.

The Show Map List as Color Composite dialog box appears; select a map list.

Subsequently, the Display Options - Map List as Color Composite dialog box appears; this dialog box will furthermore directly appear:

- when you use the right mouse button on a map list in the Catalog and select Visualization, as Color Composite from the context-sensitive menu,
- when you double-click a map list in the Catalog, and the map list contains raster maps with the `Image` domain or a value domain.

In the Display Options - Map List as Color Composite dialog box:

- from the map list, select three raster maps (domain `Image` or a value domain) which should be displayed in Red, Green and Blue,
- for each map, specify the stretch range.

The interactive color composite is displayed in a map window.

- You can easily change intervals, select other bands, etc. by changing the display options of the map list. Open the Layers menu in the map window, choose Display Options and select the map list, or use the right mouse button in the map window and select the map list from the context-sensitive menu.
- To save an interactive color composite, save the contents of the map window as a map view. Open the File menu in the map window and choose the Save View command.

Permanent color composites

The output of the Color composite operation is stored as a raster map on disk.

To create a permanent color composite:

- from the Operations menu in the Main Window, choose Image Processing, Color Composite, or
- double-click the ColorComp operation in the Operation-list, or
- drag an image from the Catalog to the ColorComp operation in the Operation-list.

The Color Composite dialog box appears:

- select three raster maps (only domain `Image` allowed) which should be displayed in Red, Green and Blue.
- specify the type of color composite which you want to create:
  - standard color composite: choose between linear stretching (values or percentage intervals) and histogram equalization,
  - dynamic color composite: specify the number of output colors,
  - 24-bit color composite (when your graphics board is configured to use more than 256 colors): choose between linear stretching (values or percentage intervals), histogram equalization or use instead of images, Hue, Saturation and Intensity bands as input.

8.5.2 How to create a georeference

A georeference stores the relation between rows and columns of a raster map (row,col) and real world coordinates (X,Y) or (X,Y,Z). A georeference is needed for raster maps. A georeference uses a coordinate system.

It is advised that raster maps of the same area use the same georeference, because raster operations in which raster maps are combined will only make sense if the pixels in the maps refer to the same area on the ground.

There are basically five types of georeferences:

- **georeference corners**: a North-oriented georeference to be used during rasterization of vector data or as the North-oriented georeference to which you want to resample maps;
- **georeference tiepoints**: a non-North-oriented georeference to add coordinates to a satellite image or to a scanned photograph, a scanned map, etc. without using a DTM;
- **georeference direct linear**: to add coordinates to a scanned photograph while using a DTM;
- **georeference orthophoto**: to add coordinates to a scanned photograph while using a DTM and camera parameters;
- **georeference 3D**: to create a three dimensional view of maps.

General ways to create a georeference

To create any kind of georeference, you can open the File menu of the Main window and choose Create Georeference, or double-click the New Georef item in the Operation-list. The Create Georeference dialog box will appear. You can create a georef corners, a georef tiepoints, a georef direct linear, a georef orthophoto or a georef 3D.

When an image, a scanned map or a scanned photograph is displayed in a map window, you can open the File menu in the map window, choose Create Georeference. The Create Georeference (in map window) dialog box will appear. You can create a georef tiepoints, a georef direct linear or a georef orthophoto.

Finally, also through the Edit Properties dialog box of a raster map, you create a new georeference by clicking the create georeference button. The Create Georeference (through Properties) dialog box will appear. You can create a georef corners, a georef tiepoints, a georef direct linear or a georef orthophoto.
For more information, see How to view and edit the properties of an object.

To create a georeference corners during a Rasterize operation
When rasterizing vector maps, you can directly create a georeference corners for the output raster map by clicking the create georeference button in the dialog box of the Rasterize operation. The Create Georeference (during Rasterize) dialog box appears in which you can specify the pixel size and the boundaries of the map.
When rasterizing maps of the same area, it is usually sufficient to only once create a georeference corners; for the other maps to be rasterized, select this same georeference.

To add tiepoints to an image, a scanned photograph or a scanned map
To add tiepoints to a satellite image, a scanned photograph or a scanned map, first display the image, photo or map in a map window. Then, open the File menu in the map window, and choose the Create Georeference. The Create Georeference (in map window) dialog box appears.
Choose whether you want to create a georef tiepoints, a georef direct linear or a georef orthophoto.

In case of creating a georef tiepoints or a georef direct linear, the Tiepoint editor will be opened.
In case of creating a georef orthophoto, first the Locate Fiducial Marks dialog box will be opened in which you can specify the principle distance of the camera and in which you can reference the fiducial marks of the scanned aerial photograph. After that, the Tiepoint editor will be opened.

In the Tiepoint editor, you can insert tiepoints, also called ground control points, which establish relationships between the map pixels (row,col) and XY- or XYZ-coordinates.

☞ When you want to use a created georef tiepoints for *all bands of a satellite image*: after you have finished adding tiepoints to a background map in the Tiepoint editor, you have to add the created georef tiepoints to all other bands. This can be done by opening the Properties dialog boxes of the other bands, click the Edit Properties button, then select the correct georef tiepoints.

For more information and tips, refer to the Create Georeference dialog boxes mentioned above or to How to add coordinates to a satellite image, How to screen digitize, How to monoplot, How to create an orthophoto.

To create a georeference 3D
To obtain a three-dimensional view of your study area, select Visualization, Display 3D on the Operations menu in the Main window or double-click the Display 3D item in the Operation-list. In the Display 3D dialog box, click the create georeference button. The Create Georeference 3D dialog box appears. Then, after the Display Options 3D Grid dialog box in which you can specify whether you want to see your 3D view with grid lines with or without a raster map drape, a map window is opened with an initial georeference 3D.

To modify the 3D parameters, select Georeference from the Edit menu in the map window and the Georeference 3D editor is started. Change the 3D parameters as you wish and exit the editor. The last set of parameters is stored in the georeference 3D.

The 3D view is displayed in a map window. If you like, you can add any other vector map or annotation to this map window. You can save the 3D view in the map window as a map view. To permanently apply a georeference 3D to a raster map, use the Apply 3D operation.

8.5.3 How to create an orthophoto

An orthophoto is a rectified (North-oriented raster map with square pixels) scanned photogrammetric aerial photograph with corrections for tilt and relief displacement. An orthophoto is obtained by resampling a photograph, which has a georef orthophoto to a georef corners.

Requirements

- a scanned photogrammetric aerial photograph on which you can distinguish 2, 3, or 4 fiducial marks;
- known principal distance of the photogrammetric camera;
- Digital Terrain Model (DTM) of the area.

1. Creating a georef orthophoto
You need to create a **georeference orthophoto** for the scanned photograph.
The process is described in topic How to monoplot, steps 1 - 4.

☞ For a correct behaviour of a georef orthophoto, it is essential that you have marked the 'Interpolation' check box in the Properties dialog box of your DTM raster map. For more information, refer to the Raster Map Properties dialog box.

When finished creating and editing a georef orthophoto, leave the Tiepoint editor and close the map window as well.

2. Resampling the scanned photo

Once you have a georeference orthophoto for your scanned aerial photograph, the only step you have to take is to resample the photograph to a North-oriented georef corners.

Usually, you will already have a georeference corners, e.g. the one on which you rasterized your Digital Terrain Model (DTM). In case you do not have a georeference corners yet, create one, for instance by clicking the create button in the Resample dialog box.

To start the Resample operation:

- choose Image Processing, Resample from the Operations menu in the Main window, or
- double-click the Resample operation in the Operation-list, or
- click the scanned photograph in the Catalog with the right mouse button, then choose Image Processing, Resample from the context-sensitive menu.

The Resample dialog box will appear.

In the Resample dialog box:

- select for the input raster map your scanned aerial photograph (which uses a georef orthophoto),
- select for the target georeference a georeference corners,
- choose for nearest neighbour, bilinear interpolation or bicubic interpolation,
- type a name for the output raster map,
- optionally, type a description for the output map.

For each pixel in the output map, a value will be retrieved from the nearest pixel in the input map, or a value will be calculated from 4, or 16 near pixel values in the input map.

Finally, display the output map in a map window.

☞ When you add grid lines to the resampled photo in the map window, the grid lines will be straight and perpendicular to each other.

8.5.4 How to create a representation

A representation defines the manner in which the classes of a map with a class domain, a group domain or a picture domain, or the values of a map with a value domain or the `image` domain should be represented on the screen and on a printer.

A representation stores colors or ranges of colors for the classes in a class domain, or for specific values or ranges of values for the values in a value domain. Furthermore, for polygon classes, colors, hatching and patterns can be stored; for segment classes, colors, line types and line widths, etc. can be stored; and for point classes, colors, symbols, symbol sizes, etc. can be stored.

A representation is a service object for a domain, i.e. a domain uses a certain representation.

Maps with a Bool, the `Bit`, an ID, or the `Color` domain do *not* have a stored representation on disk:

- For maps with a Bool domain or the `Bit` domain, you can interactively select colors in the Display Options dialog box of the map each time you show the map
- Maps with an ID domain can be shown in 7, 15, or 31 colors to inspect the map itself. In most cases however, you will display an ID map by one of its attributes as stored in the map's attribute table; to do so, select the Attribute check box in the Display Options dialog box of the ID map and select an attribute column
- Raster maps with the `Color` domain store colors in each pixel, so no representation is needed

Furthermore, special options for point symbols available in the Display Options dialog box of a point map.

To create a new representation for a class, group or value domain

- in a Display Options dialog box of a map with a class, group or value domain, click the little create button next to the Representation list box, or
- in the Properties dialog box of a domain, click the little create button next to the Representation list box, or
- from the File menu of the Main window, choose Create Representation, or
- in the Operation- list, double-click the New Repr item.

For more information on obtaining a Display Options dialog box or a Properties dialog box, see below.
The Create Representation dialog box will appear. A newly created representation will be directly linked to the specified domain.

Representation types

When you create a new representation for (a map with):

- a class domain: you create a *representation class*;
- a user-defined value domain: you create a *representation value*;
- the system domain Value: you create a *representation gradual*;
- any other system value domain: you create a *representation value*.

For more information on representation types, see ILWIS objects: representations.

Depending on the type of representation you are creating, the Representation Class editor is opened or the Representation Value/Gradual editor is opened.

Representation Class editor

In the Representation Class editor, you can for each class in the domain:

- select or create a color (raster form);
- select or create a color and select a hatching type or a pattern (polygon form);
- select or create a color, select a line type (e.g. single, double, triple, dot, dash dot, blocked), specify line width, have an optional support line, etc., or a symbol font and a symbol which will be used to draw lines (segment form);
- select or create a color, select a simple symbol or a symbol from any installed symbol font, specify symbol size, etc. (point form).

Furthermore, you can assign a single color or a smooth range of colors to multiple selected classes.

Representation Value/Gradual editor
In the Representation Value/Gradual editor, you can:
- insert limits, i.e. boundary values for a representation value, or percentages for a representation gradual,
- choose or create a color for each limit,
- indicate whether to stretch colors between the limits or to use the color of the upper of lower boundary,
- when you stretch in between limits: the number of steps to be used for stretching the colors.

Mind: When a map uses the system domain `Value` and you create a *representation gradual*, then in the Representation editor, boundary values can only be specified in *percentages*.
For value maps which have a specific meaning (e.g. height values) and which need fixed colors, it is therefore advisable to create a user-defined value domain and a user-defined representation value. Then, in the Representation editor, you can insert boundary values as values.
To convert a map, which uses system domain `Value` to a map with a user-defined value domain, see How to change the domain of a map.

☞ To show representations in the Catalog, refer to How to customize the Catalog.

To obtain a Display Options dialog box
- in the Catalog, double-click a map which uses a class, group, value, or picture domain, or
- from the File menu in the Main window, choose Open, and select a map which uses a class, group, value, or picture domain.

A Display Options dialog box will appear.

To obtain a Domain Properties dialog box
- from the Edit menu in the Main window, choose Properties and select a class, group, value, or picture domain, or
- in the Catalog, click a class, group, value, or picture domain with the right mouse button, and select Properties from the context-sensitive menu.

The Domain Properties dialog box will appear.

8.5.5 How to create a sample set for image classification

To perform an image classification, a sample set has to be created. Then, you can sample training pixels: assign class names to groups of pixels with similar spectral values and that are supposed to represent a known feature on the ground.

A sample set consists of:

- a reference to a map list, that is the set of images you want to classify in a later stage; the images in the map list are used to extract sample statistics during sampling and display these in feature spaces,
- a reference to a class domain, that is the collection of class names that you want to assign to your training pixels,
- a reference to a sample set raster map which is automatically created and obtains the same name as the sample set; this map contains the locations of the training pixels and the class names assigned to them.
- when your graphics board is configured to use 256 colors, a reference to a background map, that is the map on which you can locate your training pixels; this background map can for instance be a color composite.

☞ When your graphics board is configured to use more than 256 colors, you will use an interactive color composite; then, a background map on disk is not used. For more information, refer to How to create a color composite.

To create a sample set

- from the File menu in the Main window, choose Create Sample Set, or
- in the Operation-list, double-click the item New SmpleSet, or
- in the Operation-list, double-click the item Sample Map, or
- from the Operations menu in the Main window, choose Image Processing, Sample.

By the first two actions, the Create Sample Set dialog box will appear. In the this dialog box, you can specify the sample set's name, and the domain, map list, and background map that you want to use during sampling.
By the last two actions, the Sampling dialog box appears in which you can select an existing sample set, for instance to add more training pixels, but you can also create a new sample set by clicking the little create button. If the create button is used, the Create Sample Set dialog box will follow.

When everything is OK, the sample set editor is started which allows you to select training pixels, show feature spaces, etc.

8.6 How to display…

8.6.1 How to display a map list as a slide show

With Slide Show, you can display multiple raster maps that are combined in a map list as a slide show in a map window. The maps of the map list are displayed one after the other in the map window at a user-specified rate. A slide show is designed to present multi-temporal changes in maps.

Raster maps that you can include in a map list for a slide show may be:

- satellite images of the same area of for instance different months,
- derived products from satellite images, such as NDVI maps, of for instance different months,

- classified satellite images of for instance different months or years,
- thematic raster maps (e.g. land use maps) of for instance different years.

A slide show is in fact the same as repeatingly replacing raster maps in a map window, e.g. by dragging raster maps from the Catalog one by one into a map window.

To create a map list
The first step of making a slide show is to create a map list of all the raster maps that you want to show. For more information, refer to How to create a map list.

To start a Slide Show
There are several ways in which a slide show can be started:

- From the Operations menu in the Main window, choose Visualization, Show Map List as Slide Show, and select a map list in the Show Map List as Slide Show dialog box, or
- Double-click the Slide Show item in the Operation-list and select a map list, or
- Use the right mouse button on a map list in the Catalog and select Visualization, as Slide Show from the context-sensitive menu, or
- Drag a map list from the Catalog to the Slide Show item in the Operation-list, or
- Double-click a map list in the Catalog; the map list should contain raster maps which do *not* use the `Image` domain nor another value domain (if they do, an interactive color composite will be shown), or
- From the File menu in the Main window, choose Open and select a map list in which the raster maps do *not* use the `Image` domain nor another value domain (if they do, an interactive color composite will be shown), or
- Drag a map list to an existing map window. A small dialog box will appear asking you whether the maps in the map list should be displayed as a Color Composite or as a Slide Show; choose Slide Show.

A Display Options dialog box appears for the first raster map of the map list.

- specify the display options, these settings will be used for all the maps in the list.

The Display Options - Map list as slide show dialog box appears.

- specify the number of maps you want to see per minute as the Refresh Rate.

The Slide Show will start after you press the OK button.

☞ A map list can only be created of raster maps that have the same georeference and the same domain.

☞ If the refresh rate is set to zero, it means that the maps will not be refreshed automatically. To show the next slide, click the Redraw button in the button bar of the map window or press Ctrl+R on the keyboard.

☞ To stop the show, press Esc. To restart the show, press the Redraw button.

☞ When you create a map list of raster maps that are the outcome of the Apply 3D operation (for instance depicting the same area in three dimensions but from different angles), and display this map list as a slide show, you can obtain a kind of rotating DEM with raster drape. For more information, refer to Display 3D and Apply 3D.

8.7 How to edit…

8.7.1 How to edit a georeference

A georeference stores the relation between rows and columns of a raster map (row,col) and real world coordinates (X,Y) or (X,Y,Z). A georeference is needed for raster maps. A georeference uses a coordinate system.

There are basically five types of georeferences:

- **georeference corners**: a North-oriented georeference to be used during rasterization of vector data or as the North-oriented georeference to which you want to resample maps;
- **georeference tiepoints**: a non-North-oriented georeference to add coordinates to a satellite image or to a scanned photograph, a scanned map, etc. without using a DTM;
- **georeference direct linear**: to add coordinates to a scanned photograph while using a DTM;
- **georeference orthophoto**: to add coordinates to a scanned photograph while using a DTM and camera parameters;
- **georeference 3D**: to create a three dimensional view of maps.

Other types of georeferences (non-editable) are obtained when performing an operation on raster maps:

- georeference factor: created by the Aggregate map and the Densify operation;
- georeference mirrorrotate: created by the MirrorRotate operation;
- georeference submap: created by the Sub-map of raster map and the Glue raster maps operation.

Further, you can have georeference `None` for raster maps without a georeference (not advised).

☞ You can see the type of a georeference by viewing the properties of a georeference.
☞ To be able to calculate with raster maps, it is advised that raster maps of the same area use the same georeference corners or the same georeference tiepoints.
☞ To have a raster map use another georeference than its current one, perform the Resample operation.
☞ To show georeferences in the Catalog, refer to How to customize the Catalog.

To edit a georeference

- double-click a georeference in the Catalog, or
- click a georeference in the Catalog with the right mouse button, and choose Open from the context-sensitive menu,
- open the Edit menu in the Main window, choose Edit Object and select any georeference, or
- in the Operation-list, double-click the Edit item, and select any georeference, or

- drag any georeference from the Catalog to the Edit item in the Operation-list, or
- when a raster map is displayed in a map window, you can also choose the Georeference command on the Edit menu in the map window.

Depending on the type of the georeference you selected (or on the type of the georeference of the raster map in the map window), a dialog box will appear or an appropriate editor will be opened:
- for a georef corners: the Edit Georeference Corners dialog box will appear;
- for a georef tiepoints, a georef direct linear, or a georef orthophoto: the Tiepoint editor will be opened. When editing a georef orthophoto, the Locate Fiducial Marks dialog box will appear first;
- for a georef 3D: the Display Options - 3D Grid dialog box appears, in which you can specifying how the 3D view should be displayed in the map window. When the 3D view is displayed in a map window, you can select the Georeference command from the Edit menu of the map window to edit the georeference 3D. The Georeference 3D editor is started with the Georeference 3D editor: Edit dialog box.

8.7.2 How to edit a representation

A representation defines the manner in which the classes of a map with a class domain, a group domain or a picture domain, or the values of a map with a value domain or the image domain should be represented on the screen and on a printer.

A representation stores colors or ranges of colors for the classes in a class domain, or for specific values or ranges of values for the values in a value domain. Furthermore, for polygon classes, colors, hatching and patterns can be stored; for segment classes, colors, line types and line widths, etc. can be stored; and for point classes, colors, symbols, symbol sizes, etc. can be stored.

A representation is a service object for a domain, i.e. a domain uses a certain representation.

Maps with a Bool, the `Bit`, an ID, or the Color domain do not have a stored representation on disk:
- For maps with a Bool domain or the `Bit` domain, you can interactively select colors in the Display Options dialog box of the map each time you display the map;
- Maps with an ID domain can be shown in 7, 15, or 31 colors to inspect the map itself. In most cases however, you will display an ID map by one of its attributes as stored in the map's attribute table; to do so, select the Attribute check box in the Display Options dialog box of the ID map and select an attribute column.
- Raster maps with the Color domain store colors in each pixel, so no representation is needed.

Furthermore, special options for point symbols available in the Display Options dialog box of a point map.

☞ You can see which representation is linked to a certain domain by viewing the properties of a domain.
☞ You can see the type of a representation by viewing the properties of a representation.
☞ To show representations in the Catalog, refer to How to customize the Catalog.

To open and edit a representation

- double-click a representation in the Catalog, or
- click a representation in the Catalog with the right mouse button, and choose Open from the context-sensitive menu,
- open the Edit menu in the Main window, choose the Edit Object command, and select any representation, or
- in the Operation-list, double-click the Edit item, and select any representation, or
- drag any representation from the Catalog to the Edit item in the Operation-list, or
- when a map is displayed in a map window, you can also choose the Representation command from the Edit menu in the map window, or
- open a class domain, and in the domain class editor, choose the Representation command from the Edit menu.

When a map with a class domain is displayed in a map window, you can also set the Double-Click Action in the map window to Edit Representation. For more information, see the section on interactively changing colors in maps below.

Depending on the type of the representation you selected (or the type of the representation of a map in a map window), the Representation Class editor or the Representation Value/Gradual editor is opened.

For a representation class, the Representation Class editor is opened:

- for each class in the class domain to which this representation fits, you can:
  - select or create a color (raster form);
  - select or create a color and select a pattern (polygon form);
  - select or create a color, select a line type (e.g. single, double, triple, dot, dash dot, blocked), specify line width, have an optional support line, or select a symbol font and a symbol which will be used to draw lines, etc. (segment form);
  - select the symbol type, size, pen width, color and fill color (point form);
- furthermore, you can select multiple classes and assign a single color or a smooth range of colors to the selected classes.

For a representation value or gradual, the Representation Value/Gradual editor is opened:

- insert limits, i.e. boundary values for a representation value, or percentages for a representation gradual,
- choose or create a color for each limit,
- indicate whether to stretch colors between the limits or to use the color of the upper of lower boundary (slice),
- when you stretch in between limits: the number of steps to be used for stretching the colors.

☞ For maps, which use system domain `Value`, only a representation gradual can be created which works on percentages. When you want to create a representation value (working on values) for such maps, you need to change the domain of the map from system domain `Value` to a user-defined value domain. For more information, see How to change the domain of a map.

To interactively change colors in a map

1. For maps with a class or group domain:
   - Display the map in a map window and make sure that the Info check box in the map's Display Options dialog box is marked;
   - Open the Layers menu in the map window and choose the Double-Click Action command; in the appearing dialog box, select the Edit Representation option;
   - Then, double-click the pixels, polygons, segments or points in the map: each time an Edit Repr Item dialog box will appear in which you can directly change colors, polygon patterns, segment line types, line widths, point symbols, etc.
   - For a polygon, segment or point map, the map window will directly show the map in the changed representation. For raster maps, click the Redraw button in the map window to apply the changes;
   - Continue double-clicking units in the map and editing the representation of that class until satisfied.

   For more information, see Map window: advanced functionality.

2. For any raster, polygon, segment or point map with a class, group or value domain:
   - Display the map in a map window;
   - Open the Edit menu in the map window and choose the Representation command;
   - Depending on the type of representation that the domain of this map is using, the representation class editor or the representation value/gradual editor is opened;
   - In the appearing representation editor window, edit one or more colors, etc;
   - Press the Redraw button in the map window.
   - Continue editing colors in the representation editor and redrawing the map window until satisfied.

   In this way, you can work with a representation editor and a map window simultaneously.

☞ You cannot edit a representation in this way when you displayed a value map in the map window with a system representation like `Gray`, `Pseudo`, etc., or when a representation is read-only. In those cases the Representation command is grayed out on the Edit menu of the map window.

Instead, you can create a new representation: in the Display Options dialog box of the map, click the create button next to the Representation list box. The Create Representation dialog box will follow.

- When the map uses the `Image` domain or a value domain which is **not** system domain `Value`, you will create a *representation value*(which works on values);
- When the map uses system domain `Value`, you will create a *representation gradual* (which works on percentages).

After the Representation Value/Gradual editor is opened, you can interactively edit the new representation as described above.

Editing a representation class in table form
Advanced users may wish to edit a *representation class* in table form; click a representation class with the right mouse button in the Catalog, and choose Open As Table from the context-sensitive menu.

8.8 How to export data

Exporting data using Export

You can use Export to export ILWIS maps and tables to the following packages and formats:

Raster:	ILWIS 1.4, ILWIS 1.4 ASCII (.ASC) , Arc/Info non-compressed ASCII (.NAS) , IDA (.IMG) , Windows Bitmap (.BMP) , Erdas (.GIS, .LAN) , Geosoft (.GRD) , GIF images (.GIF) , Idrisi maps (.DOC, .IMG) , PaintBrush pictures (.PCX) , Themak2 (.THR) , (Geo)TIFF images (.TIF) .
Vector:	ILWIS 1.4, ILWIS ASCII segments(.SMT) , Arc/Info .E00 files, Arc/Info ASCII .LIN, Arc/Info ASCII .PTS, Arc/View Shape files (.SHP, SHX, DBF) , Atlas segment files (.BNA) , AutoCad .DXF files, Cart/o/graphics (.CV, .VTE) , Erdas (.DIG) , Gina (.GIA) , HPGL (.HPG) , InfoCam (.SEQ) , Intergraph (.SIF) , Themak2 polygons (.THP) , Themak2 segments (.THS) , UseMap (.USE) .
Table:	ILWIS 1.4, dBase (.DBF) , dBase-SDF (.SDF) , ASCII comma delimited (.TXT) .

- Open the File menu in the Main window and choose Export, or
- Double-click the Export item in the Operation-list.

The Export dialog box will follow.

You can also export ILWIS maps and tables, and all other objects that are internally stored as a table, to one of the formats listed above by typing an expression on the command line or by creating a script which contains one or more export commands. For more information, see Appendices: ILWIS script language (syntax).

Exporting data using Convert 1.4

You can use Convert 1.4 to export data to the following packages and formats:

Raster:	ITC's Laboratory of Image Processing files (.LIP), Post Script level 2 (.PS).
Vector:	ILWIS ASCII polygons (.SMT), ILWIS 1.2, ILWIS ASCII ArcVtp.
Table:	Lotus (.DIF).

1. First, export your data to ILWIS 1.4 format:
   - Open the File menu in the Main window and choose Export, or
   - Double-click the Export item in the Operation-list.

   The Export dialog box will follow.

2. Then, double-click the Convert 1.4 item in the Operation-list.
   The Convert program of ILWIS 1.41 will be started in a DOS box; choose whether you want to export Vector Output, Raster Output or Table Output and choose the format to which you want to export. Answer all questions manually. Your data will now be available in the selected format.

8.9 How to import…

8.9.1 How to import ASCII point data

When you have lists of coordinates with or without extra attribute data, e.g. in a spreadsheet, you can choose to:

- *copy* the Windows data to the clipboard and *paste* it into an ILWIS 2 table; then you can convert the table to a point map with an attribute table, or
- save the external data as an *ASCII file* and edit the ASCII file to insert a one-line header (this simulates the ILWIS 1.4 point map file structure); then you can *import* the ASCII point file as ILWIS 2 point map.

Both methods are described below; you can use the method, which suits you best.

This topic describes the tricks to obtain a *point map* with or without additional attribute table from external Windows packages or from an ASCII file. You can also use the descriptions below to obtain an ILWIS table from an external Windows package or ASCII table.

Copying and pasting through the clipboard

1. Start the Windows application with which you normally show your point data; e.g. a spread sheet, a data base or a Word *table*. Open the point data file, and:
   - inspect how many columns you have;
   - inspect what type of columns you have, i.e. character columns, numerical columns (number of decimals), etc.
   - Keep the Windows application open.

2. Start ILWIS and from the File menu in the Main window, choose the Create Table command.
   - In the Create Table dialog box:
     - specify a new name for the table;
     - choose to create a table with domain `None`;
     - specify for the number of records: **0**.
   - The table is displayed in a table window.
   - Add columns to the table of the correct type and make sure that you follow the column order of the other package.

For each column in the other package, open the Columns menu in the ILWIS table window, and choose the Add Column command.
In the Add Column dialog box:
- type a *name* for the column (this is up to you, column names in ILWIS 2 can be as long as you like);
- select a *domain* for the column. It is advised to choose either system domain `String` or system domain `Value` as explained below. When you already have created domains, which fit the column data, you can choose an existing domain.
  When the other package shows:
  - a coordinate column, choose for the ILWIS column the system domain `Value`, and specify the number of decimals;
  - a column with *numbers* or *strings* which represent the *name* by which the points are identified, choose for the ILWIS column the system domain `String`;
  - a column with *numbers* representing the *values* of the points, choose for the ILWIS column the system domain `Value`, and specify the number of decimals;
  - a column with *numbers* which should be considered as other attribute information, choose for the ILWIS column the system domain `Value`, and specify the number of decimals;
  - a column with *characters* which should be considered as other attribute information, choose for the ILWIS column the system domain `String`.
- optionally, type a *description* for the column.
- When finished adding columns, keep the table window open.

3. Return to the other Windows application, and:
   - select the data which you want to copy,
   - choose Edit, Copy.

   The data will be copied to the clipboard. You can inspect the clipboard contents by using the Windows Clipboard Viewer.

4. Return to your ILWIS table, and:
   - make sure that no fields are selected (press the ESC key to clear any selections);
   - open the Edit menu and choose Paste.

   The data is now pasted into your ILWIS table.

5. The next step is to convert any `String` columns in the table into class or ID columns. Only columns containing (long) descriptions of the points, which you do not want to use as point names, can remain columns with domain `String`.
   In the ILWIS table:
   - Double-click the column names of the columns with a `String` domain.
   - In the appearing Column Properties dialog box, click the Create Domain button.
   - In the Create Domain for String Column dialog box:

- type a name for a new domain;
- create an ID domain when the point names in the column are unique; if the table contains a column with strings by which points are identified, you should create an ID domain for that column;
- create a Class domain when the strings in the column are not unique;
- apply the new domain to the selected column.

6. You can now convert the table to a point map. Depending on whether:
   - you do not mind to obtain new IDs for your points, continue with step 6a;
   - your table only contains coordinate columns X, Y and one value column, e.g. Z, continue with step 6b;
   - you want to keep original IDs of the points, continue with step 6c.

6a. Although the table may have an *ID column* representing *point names*, you do not mind to obtain *new ID*s for the points. The table may also contain additional attribute information.
Use the Table to Point map operation:
- From the Operations menu in the Main window, choose the Table Operations, Table to PointMap command, or in the Operation-list, double-click the TblPnt item.
- In the Table to Point Map dialog box:
  - select the table;
  - select the column which contains X-coordinates;
  - select the column which contains Y-coordinates;
  - select or create a coordinate system for the point map (for more information, see the Create Coordinate System dialog box);
  - choose to use the *record numbers* in the table to automatically create a *new domain* for the output point map;
  - optionally, type a prefix for the new ID domain: every ID in the new domain will be preceded by this prefix;
  - type a new name for the output point map;
  - optionally, type a description for the output point map.

Result: a point map with linked attribute table; the domain of the point map and the attribute table contains new IDs; the point map will have a coordinate system.

6b. Besides the coordinate columns, the table only has one *value* column with point values (e.g. columns X, Y, Z).
Use the Table to point map operation:
- From the Operations menu in the Main window, choose the Table Operations, Table to PointMap command, or in the Operation-list, double-click the TblPnt item.
- In the Table to Point Map dialog box:
  - select the table;
  - select the column which contains X-coordinates;
  - select the column which contains Y-coordinates;

  - select or create a coordinate system for the point map (for more information, see the Create Coordinate System dialog box);
  - choose to use the *value column* in the table as the domain for the output point map;
  - type a new name for the output point map;
  - optionally, type a description for the output point map.

Result: a point map without attribute table; the domain of the point map is a value domain; the point map will have a coordinate system.

6c. The table has an *ID column* representing *point names* and you want to keep these *original IDs* in your point map. The table may also contain additional attribute information.
First use the Change domain of table operation, then use the Table to Point map operation.

Using the Change domain of table operation
The ID column with point names will become the domain of your table.

- From the Operations menu in the Main window, choose the Table Operations, Change Domain command, or in the Operation-list, double-click the TblChDom item.
- In the Change Domain of Table dialog box:
  - select the table;
  - select the *ID column* representing the names of the points;
  - do not aggregate;
  - type a new name for the output table.
- Open the output table to check the result: you will see that the point names or point IDs are now displayed on the gray record buttons, i.e. the table now uses a class or ID domain; the record order will be the order of the ID domain.

Using the Table to point map operation
The table with the ID domain will be converted to a point map and an additional attribute table.

- From the Operations menu in the Main window, choose the Table Operations, Table to PointMap command, or in the Operation-list, double-click the TblPnt item.
- In the Table to Point Map dialog box:
  - select the table;
  - select the column which contains X-coordinates;
  - select the column which contains Y-coordinates;
  - select or create a coordinate system for the point map (for more information, see the Create Coordinate System dialog box);
  - choose to use the *domain of the table* as the domain for the output point map;
  - type a new name for the output point map;
  - optionally, type a description for the output point map.

Result: a point map with linked attribute table; the domain of the point map and the attribute table contains the original IDs; the point map will have a coordinate system.

Simulating the file structure of an ILWIS 1.4 point map (ASCII)

You will open the ASCII file and edit it (adding a one-line header which contains column names and types of columns) to simulate the file structure of an ILWIS 1.4 point map and then import the ILWIS 1.4 point map into ILWIS 2.

Standard ILWIS 1.4 point map

An ILWIS 1.4 point map is an ASCII file with extension .PNT; an ILWIS 1.4 point map may contain the following:

X!	Y!	Name$	Attrib1#	Attrib2%	Attrib3&
5000102	560205	Point1	253	8356	63.9
5001102	561205	Point2	132	572	84.5
5002102	562205	Point3	96	1830	47.2
5003102	563205	Point4	58	6293	64.5

Columns `X` and `Y` are the columns containing the coordinates; column `Name` contains the name with which points are represented and columns `Attrib1`, `Attrib2` and `Attrib3` are optional attribute columns.
Column names in ILWIS 1.4 may not be longer than **20** characters.

As you can see, each column name is directly followed by a special sign, which defines the *type* of the ILWIS 1.4 column. You can use:

colname$	to store any strings or characters
colname#	to store byte values (whole positive numbers between 0 and 255)
colname%	to store integer values (whole numbers between -32766 and 32767)
colname!	to store long integer values (whole numbers between -2 to +2 billion)
colname&	to store real values (values with decimals)

Furthermore, an ILWIS 1.4 point map complies to the following:

- It is obligatory that there are one or more spaces between each column name and also between the strings or values in the columns themselves; i.e. the file is space delimited. This implies that empty fields are not allowed! You should fill empty fields with a question mark or any other character you like.
- For a point map, it is obligatory that columns X and Y are available. Other columns are optional.
- The column type of the X and Y columns must be either ! (long integers) or & (reals).
- When you use the Name$ column, this must be a string column, thus use column type $.
- Strings in a string column cannot contain spaces as spaces are used as delimiters and indicate the start of a next column. When you have spaces in strings, you can replace these with underscores.
- The first line of the file containing the column names, is called the header of the file.
- The file has extension .PNT.

When you want to import table-like data from other packages which are not point data, you can use the same file structure as described above, only coordinate columns X and Y are not necessary and the file must have extension .TBL. You will then simulate an ASCII *ILWIS 1.4 table*.

Procedure to simulate an ILWIS 1.4 ASCII point map

1. Make sure that:
   - the file is saved in ASCII format;
   - the file has extension .PNT;
   - any empty fields in the original table have been replaced by a question mark.

   This means: when you use the `Type` *filename*`.pnt` command in a DOS window or when you open the file in NotePad, the file should be readable and should not have odd looking characters; the contents of columns should be shown next to each other but not necessarily aligned; each record should start at a new line.

2. Open the ASCII file in an editor; you can use for instance `Edit` in a DOS window or NotePad, whichever is available (or can be installed) as one of the Windows Accessories.
   - At the top of the file, add one line which represents column names and add to each column name one of the special signs to define the column type as explained above.
     - a column which contains X-coordinates must be named `X`
     - a column which contains Y-coordinates must be named `Y`
     - for these coordinate columns, use column type `!` (long) or `&` (real)
     - only when the file contains a column with *strings* representing the names identifying the points, give this column the name and type `Name$`; otherwise do not use a `Name$` column at all;

   Make sure that there is at least one space or tab between each column name.
   - Save the file as an ASCII file with extension .PNT. The file name should not be longer than 8 characters.

   Note: Very large ASCII files which cannot be handled by the DOS Editor nor by NotePad can always be opened and edited with other word processing packages. However, when saving the file in such word processing packages, you must take special care that the file is saved in ASCII format. Usually, you will have to save the file as 'Text Only' or 'MS-DOS Text' which will generate a file with extension .TXT. Then, in a DOS window, you will have to copy or rename the file to extension .PNT yourself. Avoid working with files which seem to have double extensions!

   To check whether everything is correct:
   - Open a DOS window (in Windows 3.1 by double-clicking the DOS icon in the Program Manager; in Windows '95 by choosing Start, Programs, MS-DOS Prompt);
   - In the DOS window, type: `Type` *filename*`.pnt`
     The file should be readable; the file should look like the standard ILWIS 1.4

point map as described above; the fields should be shown next to each other; each record should start at a new line.

3. Start ILWIS 2.
   - From the File menu in the Main window, choose the Import command, or double-click the Import item in the Operation-list.
   - In the Import dialog box:
     - for the Import format, select ILWIS 1.4 point map .PNT;
     - your file will now appear in the left hand list box; select it;
     - type a name for the output ILWIS 2 point map.
     - click the OK button.

Depending on whether or not the .PNT file contains a `Name$` column, either the Import Point Map from version 1.4 dialog or the Import other point maps dialog box will follow.

4a. When your file contains a `Name$` column, the Import Point Map from version 1.4 dialog box will follow:
   - choose for the Domain of the point map:

Class:	when the point names in the `Name$` are not unique (not advised);
Identifier:	when the point names in the `Name$` column are unique (advised);
Value:	when the `Name$` column contains *values* and when you do not care for an attribute table.

   - type a name for the domain of the point map and optional attribute table;
   - when the file contains additional attribute columns: type a name for the attribute table (you can use the same name as for the output point map);
   - type a name for the output ILWIS 2 point map.
   - click the OK button.

Result: ILWIS 2 point map with or without linked attribute table. When you selected to create an ID domain, the strings of the `Name$` column are used as the IDs, both in the point map and in the optional attribute table. When you selected to use system domain `Value` for the points, the points will have the value as found in the `Name$` column and an attribute table will not be available; the point map will have coordinate system `Unknown`.

4b. When your file does *not* contain a `Name$` column, in the Import Other Point Maps dialog box:
   - choose for the Domain of the point map:

Identifier:	a new ID domain will be created; the points will automatically obtain identifiers like `pnt 1`, `pnt 2`, etc.

   - type a name for the domain of the point map and optional attribute table;
   - when the file contains additional attribute columns: type a name for the attribute table (you can use the same name as for the output point map);
   - type a name for the output ILWIS 2 point map.
   - click the OK button.

Result: ILWIS 2 point map with or without linked attribute table; the domain of the point map and the optional attribute table will contain new IDs like `pnt 1`, `pnt 2`, etc.; the point map will have coordinate system `Unknown`.

5. The Catalog will now show your ILWIS 2 point map with or without attribute table. You can double-click them to open them.

6. To create a correct coordinate system for the point map, you can open the File menu in the Main window and choose the Create Coordinate System command.
   - In the Create Coordinate System dialog box:
     - choose the correct type of coordinate system (all *coordsys types* are available);
     - in any following dialog box, specify the coordinate boundaries of the map, optional projection, ellipsoid and/or datum, etc.

   - Then, in the Catalog, click the point map with the right mouse button and choose Properties from the context-sensitive menu.
   - In the Point Map Properties dialog box, click the Edit Properties button.
   - In the Edit Properties dialog box, select the correct coordinate system.

   You can also create a coordinate system for the map, by first displaying the point map in a map window. Then, open the File menu in the map window and choose the Create Coordinate System command.
   In the Create Coordinate System (in map window) dialog box:
   - choose the correct type of coordinate system (only types coordsys formula and coordsys tiepoints available);
   - in any following dialog box, specify the coordinate boundaries of the map, optional projection, ellipsoid and/or datum, etc.

   The created coordinate system will be directly linked to the map.

8.9.2 How to import data

Importing data from ILWIS 1.4

To import *single* maps or tables from ILWIS 1.4:
- open the File menu from the Main window and choose Import, or
- double-click the Import item in the Operation-list.

The Import dialog box will follow.

To import multiple maps and tables from ILWIS 1.4:
- open the File menu from the Main window and choose Batch import from 1.4, or
- double-click the Import 1.4 item in the Operation-list.

The Import from ILWIS 1.4 dialog box will follow.

For more information, see also: How to import data from ILWIS 1.4.

Importing data using Import

You can use Import to import data from the following packages and formats:

Raster:	ILWIS 1.4, ILWIS ASCII raster (.ASC) , General Raster Data, Arc/Info non-compressed ASCII files (.NAS) , Arc/Info .E00 files (.E00) , IDA images (.IMG) , Windows Bitmaps (.BMP) , Erdas (.GIS, .LAN) , GIF images (.GIF) , Idrisi maps (.DOC) , Paintbrush pictures (.PCX) , (Geo)TIFF images (.TIF) ,
Vector:	ILWIS 1.4, ILWIS ASCII segments (.SMT) , Arc/Info .E00 files, Arc/Info ASCII .LIN, Arc/Info ASCII .PTS, Arc/View Shape files (.SHP, .SHX, .DBF) , Atlas segment files (.BNA) , AutoCad .DXF files, InfoCam sequential files (.SEQ) .
Table:	ILWIS 1.4, dBase tables (.DBF) .

- Open the File menu from the Main window and choose Import, or
- Double-click the Import item in the Operation-list.

The Import dialog box will follow.

Except for General Raster Data and dBase DBF, the data types listed above can also be imported by typing an expression on the command line or by creating a script, which contains one or more import commands. For more information, see Appendices: ILWIS script language (syntax).

Importing data using Convert 1.4

You can use Convert 1.4 to import data from the following packages and formats:

Raster:	ITC's Laboratory of Image Processing files (.LIP), Geosoft (GRD),
Vector:	ILWIS ASCII polygons (.SMT), Intergraph (.SIF), HPGL (.HPG), ILWIS 1.2, Atlas polygons (.BNA), GINA (.GIA), UseMap, SiCad (.GDB), SysScan (.GPI),
Table:	Comma Delimited files (.TXT), Lotus (.DIF), DBase (.SDF).

1. First, double-click the Convert 1.4 item in the Operation-list.
   The Convert program of ILWIS 1.41 will be started in a DOS box; choose whether you want to import Vector Input, Raster Input or Table Input and choose the format from which you want to import. Answer all questions manually. Your data will be imported into ILWIS 1.4 format.

2. Now that you have imported your data into ILWIS 1.4 format, you can import it into ILWIS 2.
   - Open the File menu in the Main window and choose Import, or
   - Double-click the Import item in the Operation-list, or
   - Open the File menu in the Main window and choose Batch import from ILWIS 1.4, or
   - Double-click the Import 1.4 item in the Operation-list.

   The Import dialog box or the Batch import from ILWIS 1.4 dialog box will follow.

8.10 How to monoplot

Monoplotting is the process of creating and/or editing a segment or point map while a scanned photogrammetric aerial photograph is displayed as a background in a map window. On the photograph, you should be able to distinguish 2, 3, or 4 fiducial marks and you should know the camera's principal distance. Furthermore, a Digital Terrain Model (DTM) of the area should be available.

By creating a georeference orthophoto (differential rectification) for the photograph, you will obtain a georeference with corrections for tilt and relief displacement and you can, by using the mouse, directly digitize elements of interest on the scanned aerial photograph.

When your scanned photograph was taken with a normal camera, i.e. no fiducial marks, but you do have a DTM, you should create a georef direct linear (direct linear transformation; also corrections for tilt and relief displacement, but more tiepoints required).
When your scanned photograph was taken with a normal camera, i.e. no fiducial marks, and you have no DTM, you should create a georef tiepoints (select transformation projective, no corrections for relief displacement).

1. Display the scanned aerial photograph
Display the scanned photogrammetric photograph, which you want to use as background in a map window. When the map already has a correct georeference orthophoto, i.e. check the properties of the map, continue with step 5.

2. Creating a georeference
When the map is using georef `None`, i.e. when the status line of the map window only shows Row Col values, open the File menu in the map window, and choose the Create Georeference command.

The Create Georeference (in map window) dialog box appears:
- choose to create a **georef orthophoto**;
- specify the coordinate system and the DTM you want to use.

☞ For a correct behaviour of a georef orthophoto and subsequent monoplotting, it is essential that you have marked the 'Interpolation' check box in the Properties dialog box of your DTM raster map. For more information, refer to the Raster Map Properties dialog box.

The Locate Fiducial Marks dialog box will appear.

When your scanned photograph was taken with a normal camera, i.e. no fiducial marks, but you do have a DTM, you should create a **georef direct linear**.
When your scanned photograph was taken with a normal camera, i.e. no fiducial marks, and you have no DTM, you should create a **georef tiepoints**.
To digitize on photographs, which have no fiducial marks or on satellite imagery, see topic How to screen digitize.

The remainder of the current topic deals with georefs orthophoto.

3. Registering fiducial marks
When in the photograph, fiducial marks are available both in the *corners* of the photo and in the *middle along the edges* of the photo, then choose which type of fiducial marks you want to use. You may not mix corner fiducials with mid-edge fiducials.

In the Locate Fiducial Marks dialog box,
- specify the principal distance of the camera,
- then, zoom in on a fiducial mark in the scanned photo, position the mouse pointer carefully in the middle of a fiducial mark, and press the left mouse button (under heading RowCol (pixels), the row and column number of this fiducial mark will appear),
- type under heading Photo (mm), the location in photo-coordinates in mm of the fiducial mark as precisely as you can (decimals allowed).
- register other fiducial marks in the same way.

For more information and examples, see the Locate Fiducial Marks dialog box.

When finished, the Tiepoint editor will be opened.

4. Using the Tiepoint editor
In the Tiepoint editor, you have to insert tiepoints, also called ground control points, which establish relationships between the scanned photo pixels (row,col) and real world XYZ-coordinates.

To add tiepoints:
- first, click at a recognizable point in the photograph displayed in the map window,
- the Add Tiepoint dialog box appears. In this dialog box, the row and column values at the position of the click are filled out.
  When you already have some tiepoints, the dialog box will also come up with a suggestion for the XY-coordinates. This suggestion is the result of the calculation with the existing tiepoints and using the current transformation method. This suggestion is a measure of the quality of the current tiepoints. The suggestion is merely a suggestion, it is advised to enter your own XY-coordinates to prevent false accuracy.

Then:
- click at the same position in a map which already has correct XY-coordinates and which is displayed in another map window (Master/Slave), or
- digitize the same point in an analog paper map on a digitizer, or
- read the correct XY-coordinates for this point from a table or an analog paper map, and type these XY-coordinates in the dialog box.
- you can optionally specify a Z-value for the inserted tiepoint; otherwise the height value as found in the DTM is used.

The inserted tiepoint appears in the map window and in the tiepoint table.

A georef orthophoto, which performs a differential rectification, has a mathematical minimum of 3 tiepoints.

☞ You should always insert more tiepoints than is mathematically required.
☞ Tiepoints should be well spread over the map (XY).

For more information on the Tiepoint editor, see Georeference Tiepoint editor: functionality.

When finished editing the georeference, leave the Tiepoint editor by choosing Exit from the File menu or by clicking the Exit Editor button in the button bar. The photograph map will now use the coordinates as stored in the georeference you just created. The photograph is still displayed in a map window; the status line now shows Row Col values and XY-coordinates.

5. Creating and/or editing a segment or point map on the background map
Have the raster map, which now uses correct XY-coordinates and which you want to use as background displayed in a map window.

To create a new segment or point map, open the File menu in the map window, and choose Create Segment Map or Create Point Map. The Create Segment Map or the Create Point Map dialog box appears. Type a map name, specify a domain and optionally type a description. Accept the coordinate system and map boundaries, which are already filled out.

To edit an existing segment or point map, first add the map to the map window by selecting the Add Data Layer command from the Layers menu in the map window. When the segment or point map is displayed, choose the Edit Layer command from the Edit menu of the map window: select the segment or point map which you want to edit.

The Segment editor or the Point editor is opened. You can now add, edit and delete segments or points by using the mouse as usual.

When your background map uses a georef orthophoto, you may see that an edited segment or point does not exactly appear at the position of the pencil pointer: this is a measure of the quality of your georeference in the area. Errors will usually be larger along the edges of a photograph.

- When the deviation is small, your georeference is good enough;
- When the deviation is large, you can check the number, even distribution and quality of the tiepoints in your georeference, the quality of your DTM, etc.

☞ When you add grid lines to the photo in the map window, the grid lines will not be straight nor perpendicular to each other; the grid lines indicate the corrections for tilt and relief displacement.

6. Resampling the scanned photograph, yes or no
The crux of this topic is that you do **not** need to use the resample operation in order to monoplot on scanned photogrammetric aerial photographs, which use a georef orthophoto. An advantage of using the original image or photograph instead of a resampled one is that the original image is not disturbed by the resampling; in general, features of interest will be easier to recognize using the original data.

Performing a resample operation is required to obtain a North-oriented and rectified orthophoto. In this case, resample the photograph with a georef orthophoto to a georef corners. For more information, see How to create an orthophoto.

Furthermore, performing a resample operation is required before using MapCalc when you want to combine raster maps with for instance a georeference corners with a scanned photograph using a georef orthophoto. In this case, before using MapCalc, resample for instance your photograph with the georef orthophoto to a georef corners.

8.11 How to save…

8.11.1 How to save annotation

Simple annotations such as single texts, legends, boxes, scale bar, North arrow, grid lines, graticule, bitmaps or metafiles/pictures which are displayed in a map window can only be stored by saving the map window as a map view.

When you want to use such simple annotations as displayed in one map window in another map window as well, it is advised:
- to open the Annotation editor (map window Edit, Annotation) and
- to copy an annotation layer into the clipboard (Ctrl+C or Edit, Copy)
- you can then paste this layer into the other map window (Ctrl+V or Edit, Paste).

For more information, see also How to add annotation to a map window.

Annotation Text objects however, which can store multiple texts for multiple positions, are genuine ILWIS objects: when creating or editing an annotation text object, all text strings, positions, fonts, font sizes, colors, etc. as defined in the object will be stored in the Annotation Text object itself. For annotation text objects, there is thus no need to save a map window as a map view.

What is the use of a map view
When you want to save everything that is currently displayed in a map window, you can save the map window as a map view: all data and/or annotation layers that are currently available in the map window are stored including the current display options for all these layers.

A map view contains the names of data layers and/or annotation layers to be displayed in one map window. Also, the display settings of the layers are stored; so the system knows the colors, widths etc. for the display of each layer. Further, the georeference is stored, meaning that when you save a map view when zoomed in on a part of the map, this zoomed area will be displayed when opening the map view later.

By opening a map view later on, all data and annotation layers which are stored in the map view are directly displayed in a map window, the map window will have the same dimensions as when it was stored and there is no need to specify display options for any layer anymore.

To save annotation as a map view
- from the map window or from the Annotation editor:
- open the File menu,
- choose the Save View or the Save View As ... command.

If this is a new map view, the Save View As ... dialog box appears, in which you can specify a name and a title for the map view.

8.11.2 How to save a map view

You can save any set of data and annotation layers as currently displayed in a map window, this means including the current display options of these layers, as a map view.

A map view contains the names of data layers and/or annotation layers to be displayed in one map window. Also, the display settings of the layers are stored; so the system knows the colors, widths etc. for the display of each layer. Further, the georeference is stored, meaning that when you save a map view when zoomed in on a part of the map, this zoomed area will be displayed when opening the map view later.

By opening a map view later on, all data and annotation layers, which are stored in the map view, are directly displayed in a map window. The map window will have the same dimensions as when it was stored and there is no need to specify display options for any layer anymore.

To save the contents of a map window
- from the map window or from the Annotation editor:
- open the File menu,
- choose the Save View or the Save View As ... command.

If this is a new map view, the Save View As ... dialog box appears, in which you can specify a name and a title for the map view.

8.12 How to screen digitize

Screen digitizing is the process of creating and/or editing a segment or point map while an existing raster map is displayed as a background in a map window. The raster map can be for instance a band of a satellite image, a color composite, a scanned map, or a scanned photograph. By using the mouse, you can directly digitize elements of interest on the background map.

1. Display the background map
Display the map, which you want to use as background in a map window. When the map already has a correct georeference, i.e. when the status line of the map window shows Row Col values and XY-coordinates, continue with step 4.

2. Creating a georeference
When the map is using georef None, i.e. when the status line of the map window only shows Row Col values, open the File menu in the map window, and choose the Create Georeference command.

The Create Georeference (in map window) dialog box appears:

- For a satellite image, for a scanned map, or for a scanned photograph: when a Digital Terrain Model (DTM) of the area is not available, create a **georef tiepoints**.
  Specify the coordinate system you want to use.
- For scanned photograph, which was taken with a normal camera (i.e. no fiducial marks on the photo) and when a DTM of the area is available, create a **georef direct linear**.
  Specify the coordinate system and the DTM you want to use.

☞ For a correct behaviour of a georef direct linear and subsequent screen digitizing, it is essential that you have marked the 'Interpolation' check box in the Properties dialog box of your DTM raster map. For more information, refer to the Raster Map Properties dialog box.

The Tiepoint editor will be opened.

For a scanned photograph which was taken with a photogrammetric camera (i.e. fiducial marks on the photograph and known principal distance of the camera) and a DTM of the area is available, create a georef orthophoto. Digitizing on such an aerial photograph is called monoplotting; continue with topic How to monoplot.

The remainder of the current topic deals with georefs tiepoints and georefs direct linear.

3. Using the Tiepoint editor
In the Tiepoint editor, you have to insert tiepoints, also called ground control points, which establish relationships between the map pixels (row,col) and real world XY- or XYZ-coordinates.

To add tiepoints:

- first, click at a recognizable point in the map without coordinates which is displayed in the map window,
- the Add Tiepoint dialog box appears. In this dialog box, the row and column values at the position of the click are filled out.
  When you already have some tiepoints, the dialog box will also come up with a suggestion for the XY-coordinates. This suggestion is the result of the calculation with the existing tiepoints and using the current transformation method. This suggestion is a measure of the quality of the current tiepoints. The

suggestion is merely a suggestion, it is advised to enter your own XY-coordinates to prevent false accuracy.

Then:

- click at the same position in a map which already has correct XY-coordinates and which is displayed in another map window (Master/Slave), or
- digitize the same point in an analog paper map which is stuck on a digitizer, or
- read the correct XY-coordinates for this point from a table or an analog paper map, and type these XY-coordinates in the dialog box.

When editing a georef direct linear or a georef orthophoto, you can optionally specify a Z-value for the inserted tiepoint; otherwise the height value as found in the DTM is used.

The inserted tiepoint appears in the map window and in the tiepoint table.

For a georef tiepoints, you need to select a transformation method:

- For satellite images, an *affine transformation* will usually do (mathematical minimum of 3 tiepoints).
- For a scanned photograph (without DTM), a *projective transformation* is recommended (mathematical minimum of 4 tiepoints).

A georef direct linear, which performs a *direct linear transformation* (DLT) has a mathematical minimum of 6 tiepoints.

☞ You should always insert more tiepoints than is mathematically required.
☞ Tiepoints should be well spread over the map (XY).
☞ For a georef direct linear, the tiepoints should also be well spread in Z-direction and they should not be co-planar, i.e. in Z-direction, the tiepoints should not be on a (tilted) plane.

For more information on the Tiepoint editor, see Georeference Tiepoint editor: functionality.

When finished editing the georeference, leave the Tiepoint editor by choosing Exit from the File menu or by clicking the Exit Editor button in the button bar. The raster map will now use the coordinates as stored in the georeference you just created. The raster map is still displayed in a map window; the status line now shows Row Col values and XY-coordinates.

4. Creating and/or editing a segment or point map on the background map

Have the raster map, which now uses correct XY-coordinates and which you want to use as background displayed in a map window.

To create a new segment or point map, open the File menu in the map window, and choose Create Segment Map or Create Point Map. The Create Segment Map or the Create Point Map dialog box appears. Type a map name, specify a domain and optionally type a description. Accept the coordinate system and map boundaries, which are already filled out.

To edit an existing segment or point map, choose the Edit Other Map command from the Edit menu of the map window, then select the desired editor and select a map.

The Segment editor or the Point editor is opened. You can now add, edit and delete segments or points by using the mouse as usual.

When your background map uses a georef direct linear, you may see that an edited segment or point does not exactly appear at the position of the pencil pointer: this is a measure of the quality of your georeference in the area. Errors will usually be larger along the edges of a photograph.

- When the deviation is small, your georeference is good enough;
- When the deviation is large, you can check the number, even distribution and quality of the tiepoints in your georeference, the quality of your DTM, etc.

5. Resampling the background image, yes or no

The crux of this topic is that you do not need to use the resample operation in order to screen digitize on images which use a georef tiepoints or on scanned photographs which use a georef direct linear. An advantage of using the original image or photograph instead of a resampled one is that the original image is not disturbed by the resampling; in general, features of interest will be easier to recognize using the original data.

Performing a resample operation is only required before using MapCalc when you want to combine raster maps with for instance a georeference corners with raster maps using a georef tiepoints or a georef direct linear. In this case, before using MapCalc, resample for instance your image with the georef tiepoints to a georef corners.

8.13 How to use…

8.13.1 How to use the Georeference Tiepoints editor

This topic gives information on different methods to insert tiepoints in the Tiepoint editor for a georef tiepoints, a georef direct linear or a georef orthophoto. For more information on georeference types, see ILWIS objects: georeferences.

To insert tiepoints by using an existing map with coordinates (Master/Slave)

- Display any map with correct XY-coordinates in a map window (Master).
- Display the image that you want to give tiepoints (Slave) in another map window; open the Edit menu and choose Georeference; the Georeference Tiepoints editor will be started.
- Make sure that you can identify future tiepoints both in the slave image (row, col) and in the other master map with the correct georeference and/or coordinate system (XY-coordinates).

Then:

- click in the slave image;
- the Add Tiepoint dialog box appears, the Row and Column values are filled out; and
- click on the corresponding position in the master raster map in the other map window;
- in the Add Tiepoint dialog box, the correct X and Y coordinates are filled out;
- click OK.

Repeat for other tiepoints.

To insert tiepoints by using a map on a digitizer
Stick an analog paper map on the digitizer, setup the digitizer and reference the map. For more information, see How to setup a digitizer and reference a map.
Start the Georeference Tiepoints editor with the image that you want to give coordinates. Make sure that you can identify future tiepoints both in the image and in the map on the digitizer.
Then:

- click in the image;
- the Add Tiepoint dialog box appears, the Row and Column values are filled out; and
- click with the digitizer cursor on the corresponding position in the map on the digitizer;
- in the Add Tiepoint dialog box, the correct X and Y coordinates are filled out;
- click OK.

Repeat for other tiepoints.

To insert tiepoints by reading coordinates from paper
Start the tiepoint editor with the map that you want to give coordinates.
Then:

- click with the mouse pointer in the tiepoint editor window on a recognizable point;
- the Add Tiepoint dialog box appears, the Row and Column values are filled out;
- when you already have some tiepoints, the dialog box will also come up with a suggestion for the XY-coordinates. Do **not** accept these defaults, but always read the corresponding XY-coordinates for this point from the paper map.

Repeat for other tiepoints.

To insert tiepoints entirely by keyboard
When you have a complete relation between row and column numbers and X and Y coordinates *on paper*, you can also:

- click on a gray record button in the tiepoint table, or
- in the map window, press the right mouse button, select the Add Tiepoint command from the context-sensitive menu;

then:
- the Add TiePoint dialog box appears;
- type Row and Column values and X and Y coordinates for a tiepoint in the dialog box.

Repeat for other tiepoints.

8.13.2 How to use parameters in scripts

Parameters in a script can replace (parts of) object names, operations, etc. Parameters in scripts work as DOS replaceable parameters in DOS batch files, and must be written in the script as `%1`, `%2`, `%3`, up to `%9`.
The parameters of a script have to be filled out on the command line when you run the script. The first text string found after the script name will replace `%1` in the script, etc.

Example 1
Suppose you have a segment map 'Contour' and an existing georeference 'Geo' with a certain pixel size.

To obtain a slope map in degrees and a slope map in percentages, you can create a script (e.g. Slopes) and type in the script text box:

```
%2 = MapInterpolContour(%1,geo)
%3 = MapFilter(%2, dfdx)
%4 = MapFilter(%2, dfdy)
%5 = 100 * HYP(%3,%4)/PIXSIZE(%2)
%6 = RADDEG(ATAN(%5/100))
```

To run this script, type on the command line of the Main window:

```
run Slopes Contour.mps DEM.mpr DX DY SlopePct SlopeDeg
```

In script 'Slopes', 'contour.mps' replaces `%1`, 'dem.mpr' replaces `%2`, 'DX' replaces `%3`, 'DY' replaces `%4`, 'SlopePct' replaces `%5`, and 'SlopeDeg' replaces `%6`.
This script is more elaborately explained in MapCalc & TabCalc: creating and running scripts.

Of course, you could also specify the objects names and pixel size in the script itself, thus creating a script without parameters.

Example 2
Suppose you have three landuse maps (LU70, LU80, and LU90) of different years (1970, 1980, 1990). Each map has an attribute table linked to it (LUTBL70, LUTBL80, LUTBL90), and each table contains a column YieldMainCrop.

To obtain attribute maps of all landuse maps, you can create a script (e.g. Yields) and type in the script text box:

```
Yield%1=MapAttribute(LU%1, YieldMainCrop)
Yield%2=MapAttribute(LU%2, YieldMainCrop)
Yield%3=MapAttribute(LU%3, YieldMainCrop)
```

To run the script (e.g. called Yields), type on the command line of the Main window:
`run Yields 70 80 90`

In script Yields, '70' replaces `%1`, '80' replaces `%2` and '90' replaces `%3`. You thus obtain three attribute maps (Yield70, Yield80, Yield90) from the three input maps (LU70, LU80, LU90).

Of course, you could also specify the years in the script itself, thus creating a script without parameters.

Chapter 9

Exercises

9.1 Georeference Orthophoto

By:
Ir. J.A.M. de Brouwer
Ing. R.J.J. Dost

Data provided by:
Dr. C.J. van Westen

In the following exercises you will georeference a scanned aerial photograph using four different methods:

- an affine transformation,
- a bilinear (2nd order) transformation,
- a tilt displacement correction, and
- a projective transformation *and* a relief displacement correction.

A certain transformation method is usually selected according to the availability of a Digital Elevation Model and distortions to be expected in the image. This exercise however shows the effect of the different georeference methods especially on the accuracy obtained. The geometry of the scanned photograph is transformed to the geometry of another existing map. When the geometries of both match, an orthophoto has been obtained.

9.1.1 Materials

- The digital aerial photograph: ALPAGO
- Digital topographic maps: INFRASTR, DRAINAGE and HOUSE
- A coordinate system: ITALEAST
- A digital contour map: CONTOUR
- A digital elevation map: DEM

You can find these materials in directory C:\ILWIS22\DATA\ALPAGO.

A first look at the files

Before starting the georeferencing process, inspect the contents of the digital files. Find out the domain of the individual maps and the domain of the scanned aerial photograph and check the meaning of the segments in the segment files. Take a close look at the contour map in order to get an impression of the relief (variations) within the area. Calculate the approximate pixel size of the scanned aerial photograph.

9.1.2 An affine transformation

As a first attempt, a georeference tiepoints will be created using an affine transformation. Combine the topographic maps in one map window and display the aerial photograph in another. From the File menu of the map window of the aerial photograph choose *Create, Georeference*. Select Tiepoints. Use coordinate system ITALEAST.

Add ground control points by first clicking locations in the photograph, and then selecting the same position in the digital topographic map. Take at least 10 control points well spread over the entire area.

- What is the minimum number of control points needed for an affine transformation?
- Why should you take more than the minimum number of control points required?
- Perform a check on the accuracy of the transformation by evaluating the residuals DROW, DCOL.
- Can an accuracy check be best performed on the active or on the non-active control points?
- What is the sigma (in meters)?

Display the topographic map on top of the georeferenced photo and make a visual accuracy check. Indicate in which part of the area the distortions are large and in which part they are small. Use the Distance tool.

- How large are the distortions in meters?

9.1.3 A bilinear transformation

As a second attempt, the georeference tiepoints will use a bilinear transformation. In the Georeference Tiepoints editor choose *Edit, Transformation*. Select Second Order Bilinear.

- What is the minimum number of control points needed for a bilinear transformation?

Perform a check on the accuracy of the transformation by evaluating the residuals DROW, DCOL.

- What is the sigma (in meters)?

Display the topographic map on top of the georeferenced photo and make a visual accuracy check. Indicate in which part of the area the distortions are large and in which part they are small. Use the Distance tool.

- How large are the distortions in meters?
- Is there any improvement comparing the results with the affine transformation?

9.1.4 A projective transformation

As a third attempt, the georeference tiepoints will use a projective transformation. In the Georeference Tiepoints editor choose *Edit, Transformation*. Select Projective.

- What is the minimum number of control points needed for a projective transformation?

Perform a check on the accuracy of the transformation by evaluating the residuals DROW, DCOL.

- What is the sigma (in meters)?

Display the topographic map on top of the georeferenced photo and make a visual accuracy check. Indicate in which part of the area the distortions are large and in which part they are small. Use the Distance tool.

- How large are the distortions in meters?
- Is there any improvement comparing the result with the affine and the bilinear transformations?

9.1.5 A projective transformation and relief displacement correction

By creating a georeference Orthophoto and creating an orthophoto, both the nature of the photo (being in a central projection on a plane tilted with respect to the horizontal terrain plane) and the displacement due to relief, will be taken into account. Therefore, information about the photo itself (focal length and the position of the fiducial marks) and altitude information is required.

Entering photo information
From the menu of the map window displaying the aerial photograph, select *File, Create*, *Georeference*. Select Orthophoto.

For the creation of an orthophoto, the positions of the fiducial marks must be entered. In this exercise the positions of the fiducial marks on the photo are (in mm):

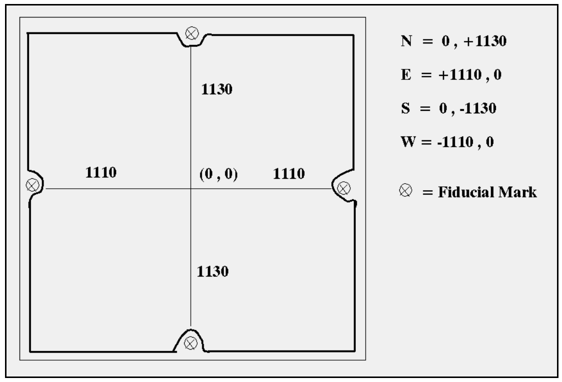
In this exercise the positions of the fiducial marks on the photo are (in mm):

Fiducial mark	X	Y
1	-106.0	-106.0
2	-106.0	106.0
3	106.0	106.0
4	106.0	-106.0

The Principal Distance is 15.2 cm.

If the fiducial marks are not given, they can be obtained from a hardcopy of an aerial photograph. The distance between the four fiducial marks (usually crosses) at the borders or corners of the photo have to be measured with an accuracy of at least

1/10th of a millimeter. Such accuracy is necessary to digitize features on the photograph. The imaginary origin is the center of the photo.
Example: northern fiducial mark is, say, x=0, y=1130 (11,30 cm in the photo) and western mark x=1110 (11,10 cm), y=0.



Entering ground control points
Enter at least 10 ground control points well spread over the entire area. Use only points, which are recognized with certainty.

Accuracy check
Perform a check on the accuracy of the transformation by evaluating the sigma values.

- What is the overall accuracy (in meters)?

Display the topographic map on top of the georeferenced photo and make a visual accuracy check. Indicate in which part of the area the distortions are large and in which part they are small.

- Are there any improvements comparing the results of the affine, the bilinear and the projective transformations?

Display a grid on top of the georeferenced aerial photo; use a grid distance of 500 and choose yellow as display color.

- Explain the shape of the gridlines. In which direction (North, East, South or West) do they converge and why?

Zoom in on a hilly part of the georeferenced aerial photo.

- Explain the shape of one grid line.

In the properties dialog box of the DEM select *Interpolation* and *BiLinear*, click OK and ‘refresh’ the georeferenced aerial photo (do not enlarge it, but keep it zoomed in).

- Explain the change of shape of the gridline.

Study the *Additional Info* in the properties dialog box of the Georeference Orthophoto and find the flying height and nadir point of the airplane at the moment of exposure.

Open the Pixel Info window and find the height and steepness of a wood covered hill.

Resample the aerial photograph to create an orthophoto (i.e. resample the photograph which uses the georeference orthophoto to the correct North-oriented georeference).

9.2 Geostatistics in ILWIS 2.2

By:
J. Hendrikse

Data provide by:
A. Gieske
University of Botswana
Free University Amsterdam

This is an introductory exercise using a case study on Carbon14 (C^{14}) water measurements in Botswana, Southern Africa.

9.2.1 Point maps with a value domain

1. Start ILWIS and go to directory C:\ILWIS22\DATA\CARBON14.

2. A point map with C^{14} percentages is given and named C14; you can find it in the Catalog. Open point map C14. In the Display Options dialog box check *Attributes* ON, accept column C14 and all other defaults. Then click OK.

3. Move with your mouse pointer over the map window and keep the left mouse button pressed. Notice that the pixel values are displayed for each position of the mouse pointer:
   - next to the mouse pointer and
   - at the status line.

4. Select *Layers, Add Annotation, Grid Lines* from the menu. Accept the defaults Display Options.

   - Estimate the size of the map (X-extent and Y-extent) and verify this using the Distance tool: Press the right-most button in the button bar (the pair of compasses) and measure some distances.
   - Measure the length of both the shortest and the longest point-pair vector (zoom in if needed). Find also the angle and the C^{14} percentage difference for each point pair.

5. Select point map C14 in the Catalog. To view the properties of the map, use the right mouse button on the map in the Catalog, and select Properties from the context-sensitive menu. Then, use the right mouse button again on the point map in the Catalog and open the point map as a table.

   - How can you verify that the map is from an area in Botswana, Africa?
   - How many different point pairs can be made?

6. Open the table named C14. In the C14 column you will find the percentages of C^{14} content (p.m.c = percents ‘modern carbon’), measured in well water at all locations appearing in the point map. Percentages are given with respect to a stabilized C^{14}/ C^{12} ratio.

7. The other columns show the computed standard deviation and two outcomes of variance computations.

   - Can you explain the difference in the columns VAR and VAR52_51 ?

9.2.2 Spatial correlation and empirical semi-variogram

1. Select the point map in the Catalog, press the right mouse button and select Spatial Correlation, or double-click the SpatCorr operation in the Operation-list. Specify a lag spacing of 10 km. Call the output table SPCORR10 and check Show ON. Type in the description box: ‘Point pair statistics of C14 with lag-interval of 10km’ (This text is optional and can always be changed later). Accept all other default values and click OK.

2. Inspect the different columns and their properties. Double-click the column names to view details, i.e. the Column Properties.

   - What does 37 in the first field of the NRPAIRS column mean precisely?
   - How many point pair “vectors” are longer than 55 km?
     Do the six longest vectors show the expected semi-variance

3. Select *Options, Show Graph* and choose DISTANCE for the X-axis and SEMIVAR for the Y-axis.

   - In which units are these coordinates X and Y expressed respectively?

4. Repeat steps 1, 2, and 3 for a lag-interval of 5 km (you can skip the questions). Use SPCORR5 for the output table name.

   - Explain the differences in the NRPAIRS column, even in records with equal lag.

5. Make a bi-directional SPATCORR table with 10km lag interval. For the primary direction take an angle of 15° (clockwise from the map north), a tolerance of 30°, and no band limitation. Call this table BI1530.

   - Inspect the columns of the table BI1530 and explain the main differences between columns NRPAIRS1 and NRPAIRS2.
   - Why don’t they sum up to $52 \times 51 / 2$ point pairs? Where are the remaining ones?
   - Compare columns SEMIVAR1 and SEMIVAR2. Do you conclude a trend or anisotropy from these figures? Why?

Close all map windows and tables that are still open.

9.2.3 Modeling the semi-variogram

1. Open the table SPCORR10.
   Select *Option, Show Graph* and display a *point* graph of SEMIVAR against DISTANCE. This is the empirical (= physically measured) semi-variogram.

2. The ('theoretical' or 'estimated') Semi-variogram Model is now made as follows:
   In the graph window select *Graph, Add SemiVariogram Model.*
   Choose *Spherical*, 200, 650 and 40 km for the model parameters.

   - At which distance (X-coordinate) do you find the greatest discrepancy between the semi-variogram model and measured values?
   - To which PMC (percent modern Carbon) does this discrepancy correspond?
   - Make a list of all available semi-variogram models or functions (Spherical, Circular…), their numeric parameters and the units in which these parameters are measured.

3. Select *Graph, Add Semivariogram Model* and add a circular semi-variogram model with the same parameter values as above. Use a different color.

9.2.4 Point map interpolation

When you already familiar with point interpolation operations you can skip this part.

1. Select georeference C14 from the Catalog. View its properties by using the right mouse button.

   - Explain the term Georeference Corners.
   - How many pixels (RowCols) will make up a raster map based on this georeference?
   - Write down its coordinate boundaries and the pixel size. Do they fit?

2. Select the point map C14.
   Use the right mouse button and select *Interpolation, Nearest Point*.
   Check the *attribute box* ON, select column C14 and use C14 georeference C14.
   Call the output map C14NP. Let it show immediately (check the S*how box* ON).
   Accept all other default and click OK.

   - Do you know the name of the type of raster map you obtain?

3. In the catalog click the point map with the right mouse button and select *Interpolation, Moving Average*.
   Check the A*ttribute box* ON, select column C14 and check the S*how box* ON.
   Call the output C14MA, use georeference C14 and accept other defaults.
   Click OK to execute the gridding operation by Moving Average.

9.2.5 Kriging estimation and error map

1. Select the point map C14.
   Use the right mouse button and select *Interpolation, Kriging*.

2. Accept the default Spherical Model and fill in Nugget, Sill and Range values as in 9.2.3.
   Specify a limiting distance of 25 km.
   For *Nr of points* fill in 4 for *Min* and 14 for *Max*.
   Call the output raster map KO25 and call the Error map FO25. Check *Show* ON.
   Accept all other defaults and click OK.
   The calculation can take a few minutes.

   - Which parameters influence the speed of calculation?
   - Explain the numbers in the *Tranquilizer*, i.e. the numbers above the increasing progress bar.

3. Use the right mouse button to view the properties of raster maps KO25 and FO25.

   - Find the range of output values in both maps. Which units?
   - Was the Kriging estimation ‘simple’ or ‘ordinary’? What’s the difference?

4. Perform another Kriging estimation from the C14 point map with a search radius (limiting distance) of 40 km, copy all other parameter values from KO25.
   Call the output maps KO40 and FO40 respectively.

5. Open both error maps FO25 and FO40 and close all other maps and tables.
   Drag the input point map C14 on top of each of the error maps. Use crosses as point- symbols.

   - Find the point with the best Kriging estimate and the point with the worst one in both estimation results.

6. Make a simple Kriging interpolation similar to the previous tasks.

   - List the parameters that don’t enter into the process anymore compared to ordinary Kriging). Call the output raster maps KS and FS (estimate map and error variance map respectively).

7. Drag the input point map over raster map FS.

   - Find the point with minimum and the point with maximum error variance (‘uncertainty’).

Appendix A

ILWIS objects

Note: This appendix replaces Appendix B.1 in the ILWIS 2.1 Reference Guide.

Data objects

Raster maps, polygon maps, segment maps, point maps, tables and columns are called data objects.

Raster maps
A raster map consists of **pixels** (picture elements) of a certain size, e.g. 20m × 20m. Pixels are either codified by IDs, class names, values, or colors; this is determined by the domain of the map. A raster map should have coordinates, that is a georeference. In ILWIS, most spatial operations are performed on raster maps.

To obtain a raster map:
- rasterize an existing point, segment or polygon map, or
- create a new raster map and edit it with the pixel editor,
- use a satellite image which is already a raster map, or
- scan a map or photograph and import it into ILWIS.

Polygon maps
A polygon map is a vector map containing closed **areas** and the boundaries making up the areas. Polygon maps can for example contain uniquely codified areas such as cadastral plots, or mapping units such as land use classes, geological units or soil units. The areas are either codified by IDs, class names or values; this is determined by the domain of the map. Further, a polygon map uses a certain coordinate system. Polygon maps are generally used as a stepping stone to raster maps.

To obtain a polygon map, you first have to create a new segment map and edit it with the segment editor (with or without digitizer); then polygonize these segments within the segment editor or with the Segments to Polygons operation.

Segment maps
A segment map is a vector map containing **lines** (for example roads, rivers or contour lines). Segments are either codified by IDs, class names or values (height map); this is determined by the domain of the map. Further, a segment map uses a certain coordinate system.

To obtain a segment map, you should create one and edit it with the segment editor (with or without digitizer).

Point maps

A point map contains **points**, for example water wells or sample points. Points are either identified by IDs, class names or values; this is determined by the domain of the map. Further, a point map uses a certain coordinate system.

To obtain a point map you should create one and edit it with the point editor (with or without digitizer).

Map lists

A map list is a set of raster maps, for example the bands of a satellite image. All raster maps in the map list must have the same georeference and the same domain.

A map list is used for:

- sampling and classification
- creating a color composite
- a principal components analysis, or
- present multi-temporal changes in maps as a slide show.

A map list can be created from the File menu in the Main window or while starting an operation which requires a map list as input. You can include as many maps in a map list as you like.

(Attribute) Table

A table consists of records and columns. A table is an attribute table when the table stores additional information on elements in a map; i.e. extra tabular data which relates to mapping units, points, segments, or polygons in maps.

Raster, polygon, segment and point maps of the domain type Class or ID can have attribute tables. The domain of the attribute table should be the same as the domain of the map to which it relates. An attribute table can be linked to a map or to a domain through the Properties of a map or a domain.

An attribute table can be edited when the table is displayed in a table window. When the table is linked to a map or to a domain, and the map is displayed in a map window, you can also double-click the units in the map.

Columns

A table consists of columns. You can perform calculations with columns using TabCalc.

Each column has a domain. A column with a value domain contains values, a column with a class domain contains class names, a column with domain ID contains IDs, etc. Columns can also have domain String; you can use this to type for instance descriptions.
If in an attribute table you have columns with class domains, or with user-defined value domains, you may consider to prepare a representation for these columns as well. When you open the map to which the attribute table is linked, you can directly display the map by one of its attributes.

Service objects

Domains, representations, georeferences, and coordinate systems are called service objects.

Domains

A domain is a set of possible values of a variable. In ILWIS, a domain is the set of possible IDs, class names, or values that can be used for instance in a map. All maps, tables and columns (data objects) have a domain; a domain is a service object for maps, tables and columns. One domain can be used for several data objects.

The four main types of domains are:

- ID for data objects which contain *unique identifiers* (e.g. plot 104, plot 105)
- Class for data objects that contain *classes* (e.g. soil units: clay, sandy loam)
- Value for data objects that contain measurable*, calculated or interpolated values* (e.g. height, concentration)
- Image for *satellite images* containing values between 0 and 255.

Maps using a class, value or the image domain can have a user-defined representation.

Representations

A representation generally defines the manner in which the classes of a map with a class domain or the values of a map with a value domain or the image domain should be represented on the screen and on a printer. A representation is a service object for a domain, i.e. a domain uses a certain representation. For maps, which have a specific meaning (e.g. land use classes or height values) and which need fixed colors, it is advisable to create a user-defined domain and a user-defined representation. Maps, which use the same domain, are by default displayed in the same colors.

Representation types:

- *Representation class*: for maps that have a class domain and for raster maps with a group domain or a picture domain. For each class in the domain, a representation class contains: colors for mapping units in raster maps; colors, hatching or patterns for polygons; colors, line types, line widths, etc. or equally spaced symbols for segments; colors, symbol type, symbol sizes, etc. for points.
- *Representation value* or *representation gradual*: for maps that have a value domain or the image domain. Such representations contain colors assigned to ranges of values in raster, polygon, or segment maps.

Georeferences

A georeference is a service object which stores the relation between rows and columns of a raster map (row,col) and coordinates (X,Y). A georeference is needed for raster maps. A georeference uses a coordinate system. It is advised that raster maps of the same area use the same georeference.

- For rasterized vector maps, which are usually North-oriented, you can use a *georef corners*.

- For other raster maps, for example satellite images, which are not North-oriented and in which the pixels do not represent exactly square areas on the ground, you can use a *georeference tiepoints*.
- To add coordinates to a scanned photograph while using a Digital Terrain Model (DTM), create a *georef direct linear*.
- To add coordinates to a scanned aerial photograph with fiducial marks for which you have DTM, create a *georef orthophoto*.
- To create three dimensional views of maps while using a Digital Elevation Model, create a *georef 3D*.

To combine raster maps with different georeferences, for instance in MapCalc or Cross, first use the Resample operation, so that all maps will use one georeference.

Coordinate systems
A coordinate system defines the possible XY- or LatLon-coordinates that can be used in your maps and thus stores information on the kind of coordinates you are using in your maps. You may for instance use user-defined coordinates, coordinates defined by a national standard or coordinates of a certain UTM zone. Point, segment and polygon maps always have a coordinate system. Raster maps have a georeference, which uses a coordinate system. A coordinate system is a service object for point, segment and polygon maps, and for georeferences of raster maps.

There are five main types of coordinate systems:
- *coordsys boundary only*: to define XY-coordinates for maps by only specifying the boundaries of your study area.
- *coordsys projection*: to define XY-coordinates for maps by specifying the boundaries of your study area and optionally projection information, ellipsoid information and/or datum information.
- *coordsys latlon*: to define LatLon-coordinates for maps by specifying the boundaries of your study area in Latitudes and Longitudes and optionally ellipsoid information and/or datum information.
- *coordsys formula*: when you obtained data which is using different XY-coordinates than the coordinate system of your project, and when you know the relation between the two coordinate systems.
- *coordsys tiepoints*: when you obtained data which is using different XY-coordinates than the coordinate system of your project, and when you do not know the relation between the two coordinate systems.

☞ It is advised to use one coordinate system for all your maps. In case you have data of different sources in different projections, it is advised to *transform* all data to one common coordinate system.

Special objects

Map views, histograms, sample sets, two-dimensional tables, matrices, filters, functions, scripts are called special objects.

Map views

A map view is a saved map window. When a map view is opened, the set of data and/or annotation layers that it contains is directly displayed.

A map view contains the names of data layers and/or annotation layers to be displayed in one map window. Also, the display options of the layers are stored; so the system knows the colors, widths etc. for the display of each layer. Further, the georeference is stored, meaning that when you save a map view when zoomed in on a part of the map, this zoomed area will be displayed when opening the map view later.

Histograms

A histogram is a special object, which lists frequency information on values, classes or IDs in a raster, polygon, segment or point map. A histogram is automatically calculated when displaying a value map with stretching; you can also use the Histogram operation. The values in a histogram are presented as a table; optionally a graph can be shown.
Summary information of a value histogram (mean, standard deviation, and intervals) can be viewed through the Properties dialog box of the histogram.

A **raster histogram** lists the number of pixels, the percentages and the areas per value, class or ID. If the input raster map uses a domain Value, also cumulative number of pixels and cumulative percentages are calculated.

A **polygon histogram** lists the number of polygons and the perimeter and area of polygons per class, ID, or value. If the input polygon map uses a Value domain, also the cumulative number of polygons, cumulative perimeters and cumulative areas are calculated.

A **segment histogram** lists the number of segments and their length per class, ID or value. If the input segment map uses a Value domain, also the cumulative number of segments and cumulative lengths are calculated.

A **point histogram** lists the number of points per class, ID or value. If the input point map uses a Value domain, also the cumulative number of points are calculated.

Sample sets

Prior to an image classification, sample pixels or training pixels have to be selected in a sample set. To create a sample set, first a map list and a domain have to be specified. Then, with sampling, assign **class names** to groups of pixels that are supposed to represent a known feature on the ground and that have similar spectral values in the maps in the map list.

A sample set contains:

- a reference to a map list, that is the set of images you want to classify in a later stage. The spectral values of the images in the map list, at the position of the training pixels provide the basis on which decisions are made in the classification. During sampling, these values can be inspected in the sample statistics of a certain class of training pixels, and can be visualized in feature spaces;

- a reference to a class domain, that is the collection of class names that you want to assign to your training pixels and that are the classes that you want to obtain from the classification. The representation of this domain determines in which colors the training pixels are displayed during sampling;
- a reference to a raster map which is automatically created and obtains the same name as the sample set. This so-called sample map contains the locations of the training pixels and the class names assigned to them.

When your graphics board is configured to use 256 colors, you can locate your training pixels on a background map, for instance a color composite. In case your graphics board is configured to use more than 256 colors, you will use an interactive color composite; then, a background map is not used.

Two-dimensional tables

A two-dimensional table is used to combine two raster maps with class or ID domain. It defines a new value for each possible combination of input classes or IDs.

A two-dimensional table view consists of rows, which represent the domain of one map and of columns, which represent the domain of another map. You have to assign a new value, class name or ID to the fields, which represent the combination of the domains. Then you have to apply the two-dimensional table on the command line of the Main window. The output raster contains the values, classes or IDs as entered in the two-dimensional table.

Matrices

A matrix is a 2-dimensional array of values. Matrices are calculated by the Principal Components operation and by the Factor Analysis operation. The Principal Components operation calculates a.o. a variance-covariance matrix and the Factor Analysis operation calculates a.o. a correlation matrix. The matrices can be shown.

In the properties dialog box of a matrix, some additional information of the matrix can be viewed namely the total variances in the output bands.

Filters

Filters are used in the Filter operation.

ILWIS offers you:

- a choice of six filter types: Linear filters, Rank order filters, Majority filters, Binary filters, a Pattern filter and a Standard deviation filter. Each filter type calculates results by a different method;
- a choice of about 30 standard filters;
- the possibility to create, edit and store your own Linear filters, for instance through the New Filter option in the Operation-list;
- the possibility to modify the Average filter, Majority filters, Median filters, the Pattern filter, Rank Order filters, and the Standard Deviation filter according to your wishes in the Filter : dialog box.

Functions fn

Functions can be used in all calculators in ILWIS: MapCalc, TabCalc, scripts and the pocket line calculator. Some 50 internal functions are available (see also MapCalc and TabCalc), but also user-defined functions can be created.

A user-defined function may contain any combination of operators and functions, and may use parameters representing maps and columns. Parameters in a user-defined function can have any name.

Scripts

A script is a sequence of ILWIS expressions. By creating a script, you can build a complete GIS or Remote Sensing analysis for your own research discipline. Scripts are equivalent to batch files in ILWIS version 1.4.

For more information about script syntax, see Appendices: Operators and functions in MapCalc and TabCalc, Appendices: ILWIS expressions and Appendices: ILWIS script language (syntax).

Annotation Text Txt

An Annotation Text object, also called an annotation text layer, is designed to display and store multiple texts at *multiple positions*. While simple annotation types can only be stored by saving a map window as a map view, an Annotation Text object can be stored by itself. An Annotation Text object can be edited with the Annotation Text editor or in a table window.
When creating an Annotation text object, you can base the texts that will appear in the text object on an existing point, segment, or polygon map. If you do so, the text object will contain a text string (class name, ID or value) for each point, segment or polygon in the selected map.

In the Annotation Text editor, you can then easily insert more text items, change and refine the position of texts (move), make text duplicates, and specify fonts, font sizes, appearance (bold, italics, underline), colors, rotations, etc. for (multiple) selected texts.

Appendix B

ILWIS operations

Note: This appendix replaces Appendix C in the ILWIS 2.1 Reference Guide.

VISUALIZATION

Show Map
Show a map in a new map window.

Show Table
Show a table in a table window.

Show Map List as Color Composite
Display three images or other raster maps with a value domain, present in a map list as a color composite. You will show an Interactive color composite.

By using an *interactive* color composite, you can easily change intervals, select other bands, etc. The resulting color composite is displayed in a map window, which can be saved as a map view. Interactive color composites are very suitable to be used as a background during sampling or during screen digitizing.
Your graphics board needs to be configured to use more than 256 colors, for instance High Color 16-bit, or True Color 24-bit (see Display Settings in Windows' Control Panel).

A *permanent* color composite can always be created with the Color Composite operation on the Operations, Image Processing menu.

Show Map List as Slide Show
Show multiple raster maps, which are combined in a map list, one after the other in a map window at a user-specified rate. You will show a Slide Show. This visualization technique is designed to present multi-temporal changes in maps. All maps in the map list must use the same domain and the same georeference.

You can use a slide show:

- to present the temporal changes of maps with the same theme but of different years,
- to display a number of classified satellite images of different months or years,
- to display derived products from satellite images such as NDVI maps of different months,
- to display a number of 3D views one after the other.

Display 3D
With Display 3D, you can create and edit a georeference 3D in order to obtain a three dimensional view of one or more maps. A Digital Elevation Model (DEM) is required to create a georeference 3D. The DEM can then be displayed as 3D grid lines with or without a drape of a raster map. The georeference 3D can be edited with the Georeference 3D editor.
When finished editing the 3D view, you can add point, segment and/or polygon maps and/or annotation to improve the 3D view.

Apply 3D
The Apply 3D operation resamples an input raster map according to a georeference 3D. This enables you to permanently and quickly display the output raster map in three dimensions, i.e. as a 3D view.
You can create a georeference 3D with Display 3D or Apply 3D; a Digital Elevation Model is required and you have to specify 3D view parameters using the Georeference 3D editor.

RASTER OPERATIONS

Map Calculation
Map Calculation can be used to perform calculations with raster maps. Map calculation is used for the execution of most spatial analysis functions and modelling operations. It integrates spatial and tabular data. The program enables the user to perform overlay, retrieval operations, and queries. Type map calculation formulae on the command line of the Main Window.

The following operations can be executed:
- manipulation of one or more raster maps by performing arithmetical, relational, logical, conditional, exponential, logarithmic and other operations,
- creation of attribute maps from map-related tables with attribute data,
- classification of domain Value maps according to a domain Group,
- application of user-defined functions.

Attribute map of raster map
By creating an attribute map of a raster map, the class name or ID of each pixel in the original map is replaced by the value, class or ID found in a certain column in an attribute table.

A raster map using a Class or ID domain, can have extra attribute information on the classes or identifiers in the map. These attributes are stored in columns in an attribute table. The attribute table can be linked to the map to which it refers, or to the domain of the map. You can check whether an attribute table is linked to the raster map or to its domain through the Properties dialog box of the map or the domain.

Cross
The Cross operation performs an overlay of two raster maps. Pixels on the same positions in both maps are compared; the occurring combinations of class names,

identifiers or values of pixels in the first input map and those of pixels in the second input map are stored. These combinations give an output cross map and a cross table. The cross table also includes the number of pixels that occur for each combination.

Aggregate map
The Aggregate map operation aggregates blocks of input pixels by applying an aggregation function: Average, Count, Maximum, Median, Minimum, Predominant, Standard Deviation or Sum. The Aggregate Map operation either creates a new georeference in which each block of input pixels corresponds to one output pixel (group) or the output raster map uses the same georeference as the input map (no group).

Distance calculation
Distance calculation assigns to each pixel the smallest distance in meters towards user-specified source pixels, for example to schools, to roads etc. The output is called a distance map.
The input map for a distance calculation is called a source map: all pixels with a class name, ID, or value are regarded as source pixels, and distance values will be calculated for all pixels that are undefined. In the Distance calculation dialog box, a source map can be any raster map with a class domain or an ID domain. On the command line, you can also use raster maps with a value domain.

Inaccessible or less accessible areas can be indicated in a weight map. The weight factors in such a map represent the relative difficulty, a 'resistance', to surpass pixels. By using weight factors that are inversely proportional to the possible speed that can be obtained in different mapping units, a so-called travel time map can be calculated. Through a distance calculation, also a Thiessen map can be calculated. A weight map can be used, but is not obligatory.

Iteration
Iterations are a special type of map calculations. They are a successive repetition of a mathematical operation, using the result of one calculation as input for the next. These calculations are performed line by line, pixel by pixel and take place in all directions. When a calculation in one direction is finished (for instance from top to bottom), a rotation takes place for the calculation in the next direction. The calculation stops when there are no more differences between an output map compared to the previous output map, or when a certain number of iterations is reached as defined before. Iterations are often used in combination with neighbourhood operations.

Area numbering
Area numbering is a raster operation, which assigns unique identifiers to pixels with the same class names or values that are horizontally, vertically or diagonally connected. The output of the Area numbering operation is a map in which these connected areas are codified as Area 1, Area 2, etc. Further, an attribute table is created with the map, which contains the new Area IDs and the original class names, IDs or values.

Area numbering can be used to make a decision based on the area of individual groups of pixels (uniquely identified areas) instead of on the total area of all pixels with the same class name or value.

Sub-map of raster map
The Sub-map of raster map operation copies a rectangular part of a raster map into a new raster map. The user has to specify row and column numbers of the input map to indicate the part of the input map that should be copied into the new raster map.

Glue raster maps
The Glue raster maps operation glues or merges two or more georeferenced input raster maps into one output raster map. The output map then comprises the total area of all input maps. The domains of the input maps are merged when needed. A resampling is performed when needed.

With the Glue raster maps operation, you can thus merge two or more adjacent or partly overlapping raster maps (i.e. make a mosaic), or glue smaller raster maps onto a larger one.

Mirror Rotate
The Mirror/Rotate operation allows you to reflect a raster map in a horizontal, vertical, or diagonal line, to transpose the map's rows and columns, or to rotate a raster map 90°, 180°, 270° (clock-wise).

IMAGE PROCESSING

Filter
Filtering is a raster operation in which each pixel value in a raster map is replaced with a new value.

The new value is obtained by applying a certain function to each input pixel and its direct neighbours. These neighbours are usually the 8 adjacent pixels (in a 3×3 filter) or the 24 surrounding pixels (in a 5×5 filter). When you create your own filters, any odd sized matrix is allowed (5×1, 11×23, 25×25); the maximum user-defined filter size is 8000.
Filtering is for instance used to sharpen a satellite image, to detect line features, etc.

Stretch
The Stretch operation re-distributes values of an input map over a wider or narrower range of values in an output map. Stretching can for instance be used to enhance the contrast in your map when it is displayed. Two stretch methods are available: linear stretching and histogram equalization.

Slicing
The Slicing operation classifies ranges of values of an input raster map into classes of an output map. A domain Group should be created beforehand; it lists the upper value boundaries of the groups and the output class names.

To perform an interactive slicing, you can create a representation value for the input map and change value boundaries and colors of the representation value.

Color separation
The Color separation operation allows you to extract different 'bands' for instance from a scanned or digital color photo as if using color filters when taking the picture. After color extraction, you can perform the normal Image Processing operations like Filtering, Classification, etc. on these bands.

Maps that have a Picture domain or the (24 bit) `Color` domain store for each pixel three values: Red, Green and Blue. The Color separation operation allows you to retrieve for each pixel either the Red, Green or Blue value and store these in a separate map. You can also retrieve Yellow, Magenta, Cyan, combined Gray values, or Hue, Saturation or Intensity values for each pixel.

Color composite
A color composite is a combination of three raster bands. One band is displayed in shades of red, one in shades of green and one in shades of blue. Putting three bands together in one color composite map can give a better visual impression of the reality on the ground, than by displaying one band at a time. Examples of color composites are false color (or IR) images and 'natural color' images.
The Color Composite operation on the Operations, Image Processing menu creates a *permanent* color composite raster map.

To *interactively* create a color composite, choose Show MapList as Color Composite from the Operations, Visualization menu. Interactively created color composites can be stored by saving the map window as a map view.

Cluster
Clustering, or unsupervised classification, is a rather quick process in which image data is grouped into spectral clusters based on the statistical properties of all pixel values. It is an automated classification approach with a maximum of 4 input bands.

Sample
Sample is an interactive process of selecting training pixels in a sample set prior to an image classification.

In the sample set editor, select pixels that are characteristic for a certain type of a certain natural resource on the ground and that have similar spectral values in the maps in the map list, and assign a class name to them. The spectral values of these sampled pixels or training pixels provide the basis on which decisions are made during classification. These values can be inspected in the sample statistics of a certain class of training pixels and can be visualized in feature spaces. The result of Sampling is a filled sample set.

Classify
The Classify operation performs a multi-spectral image classification according to training pixels in a sample set (supervised classification).

The following classification methods can be used:
- Box classifier
- Minimum Distance to Mean classifier
- Minimum Mahalanobis Distance classifier
- Maximum Likelihood classifier

Resample
The Resample operation resamples a raster map from the map's current georeference to another target georeference. The coordinate of each output pixel is used to calculate a new value from close-by pixel values in the input map. Three resampling methods are available: nearest neighbour, bilinear interpolation, and bicubic interpolation.

In raster operations (e.g. MapCalc, Cross), all input raster maps must have the same georeference. Thus, prior to such operations, use Resample:
- *to combine raster maps from various sources*, when maps use different coordinate systems (projections) or different georeferences (pixel size): resample the maps to one common georeference;
- *to combine satellite imagery of different dates or resolutions*: create a georef tiepoints for each set of images, then resample the images preferably to a georef corners;
- *to combine satellite images with rasterized vector maps*: rasterize the vector data on the georef tiepoints of the satellite images, or, in case you prefer North-oriented raster maps, rasterize the vector maps with a georef corners, and resample the images with the georef tiepoints to this georef corners;
- *to combine scanned photographs with rasterized vector data*, or *to rectify scanned aerial photographs*: create a georef tiepoints, a georef direct linear or a georef orthophoto for the photo, then resample the photo to a georef corners.

STATISTICS

Histogram
The Histogram operation calculates the histogram of a raster, polygon, segment or point map. Histograms list frequency information on the values, classes, or IDs in your map. Results are presented in a table and optionally in a graph. Summary information of a histogram of a Value raster map can be viewed in the properties of the histogram (mean, standard deviation, and percentage intervals).

A **raster histogram** lists the number of pixels, the percentages and the areas per value, class or ID. If the input raster map uses a Value domain, also cumulative number of pixels and cumulative percentages are calculated.
A **polygon histogram** lists the number of polygons and the perimeter and area of polygons per class, ID, or value. If the input polygon map uses a Value domain, also

the cumulative number of polygons, cumulative perimeters and cumulative areas are calculated.
A **segment histogram** lists the number of segments and their length per class, ID or value. If the input segment map uses a Value domain, also the cumulative number of segments and cumulative lengths are calculated.
A **point histogram** lists the number of points per class, ID or value. If the input point map uses a Value domain, also the cumulative number of points are calculated.

Raster

Autocorrelation - Semivariance
The Autocorrelation **-** Semivariance operation calculates the autocorrelation and semivariance of a raster map. The autocorrelation of a raster map is generated by calculating the correlation between pixel values of a raster map and pixel values of the same raster map for different shifts (lags) in horizontal and vertical directions. The semivariance, a measure for the spatial variability of a raster map, is calculated for the same shifts.

Map list

Principal Components
The Principal Components analysis calculates the variance-covariance matrix for a map list. New output bands are constructed in such a way that the largest variation is written to a new band 1, the second largest perpendicular variation to band 2, etc.

Factor Analysis
The Factor analysis operation calculates the correlation matrix for a map list. New output bands are constructed in such a way that the largest correlation is written to new band 1, the second largest perpendicular correlation in the second band, etc.

Variance-covariance matrix
The Variance**-**Covariance matrix operation calculates variances and covariances of raster maps in a map list. The variance is a means to express the variation of pixel values within a single raster map, i.e. a measure of the variation to the mean of the DN (Digital Number) values in a raster map. The covariance is a measure to express the variation of pixel values in two raster maps. It denotes the joint variation to the common mean of pixel values of the maps. Furthermore, the mean and standard deviation of each individual raster map is calculated.

Correlation matrix
The Correlation matrix operation calculates correlation coefficients of input raster maps of a map list. Correlation coefficients characterize the distribution of pixel values in two raster maps. Furthermore, the mean and standard deviation of each individual raster map is calculated.

Polygons

Neighbour polygons
The Neighbour polygons operation finds adjacent (or neighbouring) polygons in a polygon map and calculates the length of the boundaries of adjacent polygons.

Segments

Segment direction histogram
The Segment direction histogram operation calculates directions and lengths within segments, i.e. between all stored coordinate pairs of the segments. The output is a table with directions from 0 to 179° and the length and number of the segment parts in that direction. The results can be shown in a rose diagram.

Points

Spatial correlation
Spatial correlation calculates some point statistics: spatial autocorrelation (as Moran's I), spatial variance (as Geary's c) and semi-variance. Semi-variance is either calculculated in all directions (omnidirectional) or in a certain direction and the perpendicular direction (bidirectional). This may help you to get an impression of the nature of your point data, for instance prior to a point interpolation, and to find necessary input parameters for a Kriging operation.

As input for this operation, you can use a point map with a value domain, or a point map with a class or ID domain with an attribute table that contains one or more value columns (the attribute table must be linked to the map). The output of this operation is a table from which you can create graphs such as a semi-variogram.

Pattern Analysis
The Pattern analysis operation is a tool to obtain information on the spatial distribution of points in a point map. The output table contains six columns with the probabilities of finding 1 point (`Prob1Pnt`) within a certain distance from any point in your input map, then 2 points (`Prob2Pnt`), 3 points (`Prob3Pnt`), etc. Another column (`ProbAllPnt`) contains the sum of `Prob1Pnt`, `Prob2Pnt`,…, `Prob(n-1)`, in which `n` is the number of points in the input map.

By inspecting the graphs of distances against probabilities, you may recognize distribution patterns of your points like random, clustered, regular, paired etc.

INTERPOLATION

Raster

Densify raster map
The Densify raster map operation reduces the pixel size of your map. The number of rows and columns is increased and the new pixels in between the existing ones are assigned a value by means of a bilinear or bicubic interpolation.

You should use densify after a point interpolation. Further, densify can be used to improve the quality of printed raster maps.

Segments

Contour interpolation
Contour interpolation is an operation, which first rasterizes segments of a domain Value segment map, and then calculates values for pixels that are not covered by segments by means of a linear interpolation.

When using Contour interpolation on a segment map containing height (contour) information, the resulting raster map is a Digital Elevation Model.

Points

Point interpolation
In a point interpolation, the input map is a point map, and the output map is a raster map. The pixel values in the raster output map are interpolated from the point values.

There are four point interpolations, Nearest point, Moving average, Trend surface, and Moving surface:

Nearest point
The Nearest point operation requires a point map as input and returns a raster map as output. Each pixel in the output map is assigned the class name, identifier, or value of the nearest point, according to Euclidean distance. This method is also called Nearest Neighbour or Thiessen. The points in the input point map for the Nearest point operation do not need to be values necessarily; point maps (or attribute columns) with a class, ID or bool domain are also accepted.

For example, schools, hospitals, water wells, etc. can be represented by points. The output map of a nearest point operation on such a point map gives the 'service area' of the schools, hospitals or water wells, based on the shortest distance (as the crow flies) between points and pixels.
When you have many points or when you wish to use weights to indicate accessibilities, it is advised to rasterize the points, and then use Distance calculation to make a Thiessen map.

Moving average
The Moving average operation is a point interpolation, which requires a point map as input and returns a raster map as output. To the output pixels, weighted averaged point values are assigned.

The weight factors for the points are calculated by a user-specified weight function. The weight function ensures that points close to an output pixel obtain larger weights than points, which are farther away. Furthermore, the weight functions are implemented in such a way that points which are farther away from an output pixel than a user-defined limiting distance obtain weight zero.

When interpolating point values, it is for time efficiency reasons, strongly advised to choose a rather large pixel size for the output map. Further interpolation on the raster map values can be performed using the Densify operation or the Resample operation.

Trend surface
The Trend surface operation is a point interpolation which requires a point map as input and returns a raster map as output. One polynomial surface is calculated by a least squares fit so that the surface approaches all point values in the map. The calculated surface values are assigned to the output pixels.

Moving surface
The Moving surface operation is a point interpolation, which requires a point map as input and returns a raster map as output. For each output pixel, a polynomial surface is calculated by a least squares fit; for each output pixel, the surface will approach the weighted point values of the points which fall within the specified limiting distance.

Weight factors for the input points are calculated by a user-specified weight function. The weight function ensures that points close to an output pixel obtain larger weights than points, which are farther away. Furthermore, the weight functions are implemented in such a way that points which are farther away from an output pixel than a user-defined limiting distance obtain weight zero.

When interpolating point values, it is for time efficiency reasons, strongly advised to choose a rather large pixel size for the output map. Further interpolation on the raster map values can be performed using the Densify operation or the Resample operation.

Kriging
Kriging can be seen as a point interpolation, which requires a point map as input and returns a raster map with estimations and optionally an error map. The estimations are weighted averaged input point values, similar to the Moving Average operation. The weight factors in Kriging are determined by using a user-specified semi-variogram model (based on the output of the Spatial correlation operation), the distribution of input points, and are calculated in such a way that they minimize the estimation error in each output pixel. The estimated or predicted values are thus a linear combination of the input values and have a minimum estimation error. Two methods are available: Simple Kriging and Ordinary Kriging. The optional error map contains the standard errors of the estimates.

VECTOR OPERATIONS

Unique ID
The Unique ID operation can be used to assign a unique ID to all elements in a segment, polygon or point map. The result is an ID map that contains the same geographic information as the input map but now each point, segment or polygon has a unique ID. Further an attribute table is created which uses the same ID domain

as the output map; the table contains one column with the original classes, IDs or values of the input map. The domain of the table together with the column establishes the relation between the original classes, IDs or values in the input map and the output IDs.

Polygons

Attribute map of polygon map
By creating an attribute map of a polygon map, the class name or ID of each polygon in the original map is replaced by the value, class or ID found in a certain column in an attribute table.

A polygon map using a Class or ID domain, can have extra attribute information on the classes or identifiers in the map. These attributes are stored in columns in an attribute table. The attribute table can be linked to the map to which it refers, or to the domain of the map. You can check whether an attribute table is linked to the polygon map or to its domain through the Properties dialog box of the map or the domain.

Mask polygons
The Mask polygons operation allows you to selectively copy polygons of an input polygon map into a new output polygon map. The user has to specify a mask to select and retrieve the class names, IDs or values of the polygons that are to be copied.

Assign labels to polygons
The Assign labels to polygons operation can be used to recode polygons in a polygon map according to label points in a point map. For each label point, the surrounding polygon is determined; then the class name, ID, or value of the label point is assigned to that polygon.

Transform polygon map
The Transform polygon map operation transforms the XY-coordinate pairs of polygon boundaries in a polygon map from the map's current coordinate system to another target coordinate system. The Transform operation can only be used when a transformation between the coordinate systems is possible.

Segments

Attribute map of segment map
By creating an attribute map of a segment map, the class name or ID of each segment in the original map is replaced by the value, class or ID found in a certain column in an attribute table.

A segment map using a Class or ID domain can have extra attribute information on the classes or identifiers in the map. These attributes are stored in columns in an attribute table. The attribute table can be linked to the map to which it refers, or to the domain of the map. You can check whether an attribute table is linked to the segment map or to its domain through the Properties dialog box of the map or the domain.

Mask segments
The Mask segments operation allows you to selectively copy segments of an input segment map into a new output segment map. The user has to specify a mask to select and retrieve the class names, identifiers or values of the segments that are to be copied.

Assign labels to segments
The Assign labels to segments operation can be used to recode segments in a segment map according to label points in a point map. For each label point, the closest segment is determined; then the class name, ID or value of the label point is assigned to that segment.

Sub-map of segment map
The Sub-map of segment map operation copies a rectangular part of a segment map into a new segment map. The user has to specify minimum and maximum XY-coordinates for the new segment map.

Glue segment maps
The Glue segment maps operation glues or merges two or more segment maps into one output map. By default, the output map then comprises the total area of all input maps. The domains of the input maps are merged when needed.

For each input map, the user can specify a mask to select and retrieve the class names, IDs or values of the segments that are to be copied into the output map. The user can also specify a clip boundary, to copy only those segments to the output map which fall within the specified coordinate boundaries of the output map.

Densify segment coordinates
The Densify segment coordinates operation allows you to obtain more intermediate coordinates within segments. The segments of an input map are copied, and extra intermediate coordinates are added to the segments in the output map at a user-specified distance.
It is advised to use this operation before a Transform segments operation is performed.

Tunnel segments
The Tunnel segments operation reduces the amount of intermediate points within segments in a segment map. The segments of the input map are copied into a new segment map. However, for every three consecutive intermediate points within a segment, the middle one is omitted if it falls within a user-defined tunnel-width. Redundant nodes can also be removed.

Transform segment map
The Transform segment map operation transforms the XY-coordinate pairs of segments in a segment map from the map's current coordinate system to another target coordinate system. The Transform operation can only be used when a transformation between the coordinate systems is possible.

Points

Attribute map of point map
By creating an attribute map of a point map, the class name or ID of each point in the original map is replaced by the value, class or ID found in a certain column in an attribute table.

A point map using a Class or ID domain can have extra attribute information on the classes or identifiers in the map. These attributes are stored in columns in an attribute table. The attribute table can be linked to the map to which it refers, or to the domain of the map. You can check whether an attribute table is linked to the point map or to its domain through the Properties dialog box of the map or the domain.

Mask points
The Mask points operation allows you to selectively copy points of an input point map into a new output point map. The user has to specify a mask to select and retrieve the class names, IDs or values of the points that are to be copied.

Sub-map of point map
The Sub-map of point map operation copies all points within a user-specified rectangle into a new point map. The user has to specify minimum and maximum XY-coordinates for the new point map.

Glue point maps
The Glue point maps operation glues or merges two or more point maps into one output map. By default, the output map then comprises the total area of all input maps. The domains of the input maps are merged when needed.

For each input map, the user can specify a mask to select and retrieve the class names, IDs or values of the points that are to be copied into the output map. The user can also specify a clip boundary, to copy only those points to the output map which fall within the specified coordinate boundaries of the output map.

Transform point map
The Transform point map operation transforms the XY-coordinate pairs of points in a point map from the map's current coordinate system to another target coordinate system. The Transform operation can only be used when a transformation between the coordinate systems is possible.

Transform coordinates
The Transform coordinates dialog box allows you to type in XY-coordinates or LatLon coordinates, using a certain input coordinate system; the operation will then show the resulting XY-coordinates or LatLon coordinates for another target coordinate system. The Transform coordinates dialog box can only be used when a transformation between the coordinate systems is possible.

RASTERIZE

Polygons to raster
The Polygons to raster operation rasterizes a polygon map. The class names, IDs, or values in the polygon map are also used in the raster map, i.e. the domain of the polygon map is also the domain of the raster map. The user has to select or create a georeference for the output raster map.

Segments to raster
The Segments to raster operation rasterizes a segment map. The class names, IDs, or values in the segment map are also used in the raster map, i.e. the domain of the segment map is also the domain of the raster map. The user has to select or create a georeference for the output raster map.

Segment density
The Segment density operation rasterizes a segment map. For each output pixel, the total length of segment parts within the boundaries the output pixel is summed: this is the output value for the pixel. By using a mask you can specify the elements of the input map that are to be used in the calculation.

Points to raster
The Points to raster operation rasterizes a point map. The class names, IDs, or values in the point map are also used in the raster map; i.e. the domain of the point map is also the domain of the raster map. The user has to select or create a georeference for the output raster map.

Point density
The Point density operation rasterizes a point map. For each output pixel, the number of points found in the pixel is counted: this is the output value for the pixel. This operation can be used to examine the regional distribution of points.

VECTORIZE

Raster to Polygons
The Raster to Polygons operation extracts polygons from units in a raster map. The output polygon map uses the same domain as the input raster map, i.e. the class names or IDs in the input raster map will also be used for the polygons in the output polygon map. No polygons are created for pixels with the undefined value.

Raster to Segments
The Raster to segments operation extracts segments from the boundaries of mapping units in a raster map. The segments in the output map either obtain the code Segments or a special code which is a combination of the class names or IDs of the two mapping units found on either side of the segment.

Raster to Points
The Raster to Points operation extracts a point from each pixel in the raster map. Each point gets the value, class name or ID of the corresponding pixel.

Polygons to Segments
The Polygons to Segments operation extracts polygon boundaries and creates a segment map out of them. A mask can be specified to extract segments of specific polygons.

Polygons to Points
The Polygon to Points operation creates a point for each polygon in the polygon map. Each point obtains the class name, ID, or value of the corresponding polygon. In this way, polygon label points are created. Optionally, you can also obtain label points for polygons, which have no class name, no ID or no value, i.e. for undefined polygons.

Segments to Polygons
The Segments to Polygons operation polygonizes a segment map. All segments in the segment map must be closed, i.e. connected to other segments or to themselves (islands) by nodes; dead ends are not allowed. A mask can be specified to polygonize specific segments.
Mind: to interactively polygonize segments, use the Polygonize option in the Segment editor.

Segments to Points
The Segments to Points operation creates a point map from a segment map. The output point map can contain either:

- a point for each node in the segment map, or
- points at a regular distance along the segments in the segment map, or
- points for all stored coordinate pairs in the segment map.

TABLE OPERATIONS

Transpose table
The Transpose table operation interchanges the rows and columns of a table. Each row of the input table becomes a column in the output table; while column names of the input table become output domain records.

Change domain of table
The Change domain of table operation copies the contents of an input table to a new table; the new table will have another domain than the input table.

For the domain of the output table, you can choose:

- domain `None`;
- an existing class or ID domain on disk;
- a class or ID domain of a column in the input table when that column contains unique classes or IDs;
- a class or ID domain of a column in the input table where the column does not contain unique classes or IDs and other column values need to be aggregated (average, minimum, maximum, sum, last value encountered).

Table to point map
The Table to point map operation creates a point map out of a table. The table should have at least two columns, which define the X- and Y-coordinates of the points.

You can choose between the following possibilities:

- the output point map should use the same domain as the table (ID domain); the table will be linked as attribute table to the output point map;
- the output point map will use the domain of a column in the table; the output point map will have no attribute table;
- the output point map should use a new ID domain which is based on the record numbers of the table (domain None) and a user-defined prefix; the table also obtains this new ID domain and the table will be linked as attribute table to the output point map. The new ID domain will automatically obtain the same name of the output point map.

Glue tables
The Glue tables operation allows you to glue or merge two or more tables together. As input tables, you may use:

- tables with domain `None`,
- tables with class domains,
- tables with ID domains,
- tables with class domains and ID domains.

The Glue tables operation should be regarded as a tool to combine different tables. You can for instance combine or integrate attribute tables of different years. Tables with domain `None` can also be glued vertically one below the other.

Appendix C

ILWIS expressions

Note: This appendix replaces Appendix E.2.1 in the ILWIS 2.1 Reference Guide.

Any ILWIS operation like Filter, Cross and Distance calculation, can be performed by typing an ILWIS expression on the command line of the Main window. You can also use these expressions in scripts.
In this topic, the syntax of expressions of operations is described.

For an overview of MapCalc and TabCalc expressions, see Appendices: ILWIS operators and functions (MapCalc/TabCalc).
For details on creating expressions on the command line and in scripts, see Appendices: construction of expressions.

For special script commands, see Appendices: ILWIS script language (syntax).

Introduction

The general syntax for expressions is:

```
OUTMAP = expression
OUTMAP := expression
```

The definition symbol (`=`) is used to create a dependent output object; the assignment symbol (`:=`) is used to create an editable object.

In the overview below:

- **Bold**	is used for expression names, the expression name is followed by parameters (between brackets, separated by commas);
- `Courier`	is used for obliged parts in expressions or in parameters;
- *Italics*	is used for parameters with special requirements, usually a short explanation follows;

- parameters:

'rasmap'	an input raster map;
'map list'	an input map list (set of raster maps with same domain and same georeference);
'pol map'	an input polygon map;
'segmap'	an input segment map;
'pntmap'	an input point map;
'table'	an input table;

'column'	an input column;
'domain'	an existing domain; the domain will be used for the output object;
'georef'	an existing georeference except georef `None`; the georeference will be used for the output raster map;
'coordsys'	any existing coordinate system;
'sample set'	an input sample set;
'newdomain'	the output domain that will be created by the expression;
'newgeoref'	the output georeference that will be created by the expression;

- a vertical bar | represents a choice;
- a paramater in square brackets [ ] represents an optional parameter;
- the phrase 'value map' or 'map with a value domain' means that the map should have domain of type Value.
- Any operation name in the list below starting with:

`Map`	creates an output raster map
`PolygonMap`	creates an output polygon map
`SegmentMap`	creates an output segment map
`PointMap`	creates an output point map
`Table`	creates an output table
`Matrix`	creates an output matrix

☞ Some operations need a value input map. When your raster map is of domain type Class, ID or Group, and an attribute table is linked to the map with one or more suitable value columns, then you may type 'map.column' on the command line instead of parameter 'map' listed below. The operation then directly uses the values of the attribute column.

List of ILWIS expressions

The list below follows the order of the Operations menu in the Main window. For more information about the individual operations, click the hyper-links of the operation names.

VISUALIZATION

Apply 3D
`MapApply3D`(*rasmap*, *georef3D*)
rasmap input raster map cannot have georef `None`
georef3D a georeference 3D

RASTER OPERATIONS

Map Calculate
expression see MapCalc and TabCalc

Attribute Map

`MapAttribute`(*rasmap*, *column*) | *Rasmap.column*

rasmap	raster map with a Class, ID, or Group domain
column	column with Value, Class, ID, Group, Picture, or Color domain; by default a column from the attribute table of the raster map.

Cross

`MapCross`(rasmap1, *rasmap2*, output cross table)
`TableCross`(rasmap1, *rasmap2*)
`TableCross`(rasmap1, *rasmap2*, output cross rasmap)
`TableCross`(rasmap1, *rasmap2* [, output cross rasmap] [, `IgnoreUndefs` | `IgnoreUndef1` | `IgnoreUndef2`])

rasmap2 same georeference as input raster map1.

Aggregate Map

`MapAggregate`*AggFunc*(*rasmap*, *groupfactor*, *group* [,rowoffset, coloffset] [, newgeoref])
`MapAggregate`*AggFunc*(*rasmap*, *groupfactor*, *nogroup* [,rowoffset, coloffset])

AggFunc	`avg` | `cnt` | `max` | `med` | `min` | `prd` | `std` | `sum` When no aggregation function is specified, the upper left pixel of each block is used.
rasmap	raster map with a value domain for aggregate functions `avg`, `max`, `min`, `std`, `sum`; raster map with a class, ID, or value domain for aggregate function `med` ; raster map with any domain for aggregate function `prd`
groupfactor	a value (>= 1) to define the size of the blocks of input pixels to be aggregated; a value of 4 means that each block of 4 x 4 input pixels will be aggregated.
`group`	each block of input pixels is aggregated to 1 output pixel; a georeference factor is created with the same name as the output map.
`nogroup`	each block of input pixels is aggregated and the output value is stored in all pixels of the block. that correspond to the considered block of input pixels; the output map uses the same georeference as the input map.
rowoffset	optional parameter to skip the specified number of rows.
coloffset	optional parameter to skip the specified number of columns.
newgeoref	when using option `group`, optional parameter to specify the name of the output georeference; if not specified, the output georeference obtains the same name as the output map.

Distance

`MapDistance`(*source rasmap* [, *weight rasmap*| `1`])
`MapDistance`(*source rasmap*, [*weight rasmap*| `1`], output rasmap Thiessen)
`MapThiessen`(*source rasmap*[, *weight rasmap*| `1`], output rasmap Distance)

source rasmap input raster map of any domain type; for all pixels with the undefined value, a distance value is calculated

[weight rasmap| 1]
weight map is an optional parameter to specify a map with weight factors; raster map of domain type Value. When a 1 is specified or when the parameter is not used, weight factor 1 is used for all pixels.

output rasmap Thiessen
name of output Thiessen raster map

output rasmap Distance
name of output Distance raster map

Iteration

`MapIter`(*StartMap*, *IterExpr* [, *nr of iterations*])

`MapIterProp`(*StartMap*, *IterExpr* [, *nr of iterations*])

StartMap raster map that is used in the IterExpr.

IterExpr an expression for neighbourhood operations.

nr of iterations optional parameter to specify the maximum number of iterations; if not specified, the operation continues until no more changes occur.

Area Numbering

`MapAreaNumbering`(rasmap, `8|4` [, *newdomain*])

`8|4` distinguish 8-connected or 4-connected areas; default is 8

newdomain optional parameter to specify a name for the output ID domain; if not specified, the output domain will be stored by the output raster map (internal domain).

Sub Map

`MapSubMap`(rasmap, first row, first col [, nr rows, nr cols] [, *newgeoref*])

newgeoref optional parameter to specify a name for the output georeference; if not specified, then the output georeference obtains the same name as the output map.

Glue Maps

`MapGlue`(*rasmap1*, *rasmap2* [, *rasmaps*] [, *newdomain*] [, `replace`])

rasmap1 input raster map which georeference will be used for the output raster map.

rasmap2 input raster map that will be resampled if needed.

rasmaps optional parameter to specify more input raster maps. You can specify as many input raster maps as you like; comma delimited.

newdomain optional parameter, when merging class or two ID maps, to specify the name of the output domain into which all items of the input domains are merged; if not specified, the output domain will be stored by the output raster map (internal domain).

`replace` optional parameter to use for overlapping pixels the values of the last input map; if not specified, the values of the first input map are used for overlapping pixels.

Mirror Rotate

`MapMirrorRotate`(rasmap, *rotate type*)

rotate type `mirrhor| mirrvert| mirrdiag|transpose| rotate90| rotate180| rotate270|normal`

IMAGE PROCESSING

Filter

`MapFilter`(*rasmap*, *filter*| *filter expression*)

rasmap all filters use input raster maps with a value domain; the Majority and the UndefMajority filters also work on other domain types.

filter `avg3x3` | `binmajor` | `conn8to4` | `d2fdx2` | `d2fdxdy` | `d2fdy2` | `dfddn` | `dfdup` | `dfdx` | `dfdy` | `dilate4` | `dilate8` | `edgesenh` | `inbnd4` | `inbnd8` | `laplace` | `lifegame` | `majority` | `majundef` | `majzero` | `med3x3` | `med5x5` | `outbnd4` | `outbnd8` | `peppsalt` | `shadow` | `shrink4` | `shrink8` | *user-defined filter on disk*

filter expression

`FilterLinear`(*rows,cols,expression*) |
`Average`(*rows,cols*) |
`RankOrder`(*rows,cols,rank*[,*threshold*]) |
`Median`(*rows,cols*[,*threshold*]) |
`Majority`(*rows,cols*) |
`ZeroMajority`(*rows,cols*) |
`UndefMajority`(*rows,cols*) |
`Pattern`(*threshold*) |
`FilterStandardDev`(*rows,cols*)

rows,cols size of filter in rows and columns; value >= 1; maximum size of filter (rows×cols) <= 8000

threshold if the difference between the resulting value and the original pixel value is smaller than or equal to the threshold, the calculated value is used. If the difference between the resulting value and the original pixel value is larger than the threshold, the original pixel value is retained.

rank the rank number of which the pixel value is assigned to the central pixel.

expression fill the values in the filter by an expression in which you can use the parameters `x`, `y`, and `r`.

Stretch

`MapStretch[Linear]`(*rasmap*, *range from*, *domain*)
`MapStretch[Linear]`(*rasmap*, *range from*, *domain*, *range to*)
`MapStretchHistEq`(*rasmap*, *range from*, *intervals*)

rasmap	input raster map using a value domain
range from	*min:max* | *perc*
perc	real value > 0
intervals	number of output intervals
domain	output value domain
range to	value range of output map as *min:max* | *min:max:prec*| ::*prec*

Slicing

`MapSlicing`(*rasmap*, domain group)

rasmap	input raster map using a value domain

Color Separation

`MapColorSep[aration]`(*rasmap*, *color*)\| *rasmap.color*

rasmap	input raster map using a Picture domain or the Color domain
color	`red` | `green`| `blue` | `yellow` | `magenta` | `cyan` | `grey` | `gray` | `hue` | `saturation` | `sat` | `intensity` | `intens`

Color Composite

`MapHeckbert`(*map list*, *nr of colors*)
`MapColorComp[Linear]`(*map list*, *range1*, *range2*, *range3*)
`MapColorCompHistEq`(*map list*, *range1*, *range2*, *range3*)
`MapColorComp24[Linear]`(*map list* [, *range1*, *range2*, *range3*])
`MapColorComp24HistEq`(*map list*, *range1*, *range2*, *range3*)
`MapColorComp24HSI`(*map list*)

map list	existing map list which contains three raster maps of the Image domain or definition of map list as: `mlist(ImageRed, ImageGreen, ImageBlue)`
range1,2,3	min:max | perc
perc	0 ≤ real value < 50
nr of colors	2 ≤ integer value ≤ 255

Cluster

`MapCluster`(*map list*, *nr of clusters*)

map list	a map list which contains 1, 2, 3, or 4 raster maps that use the Image domain
nr of clusters	an integer value between 2 and 60 for the number of clusters in the output map.

Sample -

Classify

`MapClassify`(sample set, *classifier*)

classifier	`ClassifierBox`(factor) | `ClassifierMinDist`([threshold]) | `ClassifierMinMahaDist`([threshold]) | `ClassifierMaxLikelihood`([threshold])

Resample

`MapResample`(rasmap, georef, *resamp meth* `[`, `Patch` | `NoPatch]`)

resamp meth	`NearestNeighbour` | `BiLinear` | `BiCubic`

STATISTICS

Histogram

`TableHistogram`(*rasmap*)

rasmap	input raster map of any domain type. Mind: the output raster histogram table will always have the same name as the input raster map.

`TableHistogramPnt`(*pntmap*)

pntmap	input point map of any domain type. Mind: the output point histogram table will always have the same name as the input point map.

`TableHistogramPol`(*polmap*)

polmap	input polygon map of any domain type. Mind: the output polygon histogram table will always have the same name as the input polygon map.

`TableHistogramSeg`(*segmap*)

segmap	input segment map of any domain type. Mind: the output segment histogram table will always have the same name as the input segment map.

RASTER

Autocorrelation

`TableAutoCorrSemiVar`(rasmap, *max shift*)

max shift	maximum pixel shift; integer value > 0

MAP LIST

Principal Components

`MatrixPrincComp`(*map list*)

map list	map list containing raster maps with a value domain

Factor Analysis
`MatrixFactorAnal`(*map list*)

map list	map list containing raster maps with a value domain

POLYGONS

Neighbour Polygons
`TableNeighbourPol`(*polmap*)

polmap	input polygon map with a Class, ID or Group domain

SEGMENTS

Direction Histogram
`TableSegDir`(*segmap*)

POINTS

Spatial correlation
`TableSpatCorr`(*pntmap*)
`TableSpatCorr`(*pntmap*, *lagspacing* [, *direction* [, *tolerance* [, *bandwidth*]]])

pntmap	input point map with a value domain
lagspacing	parameter to specify length of linear distance intervals; if not specified, the output table will use logarithmic distance intervals
direction	optional parameter to find point pairs in this direction; clockwise angle from Y-axis; 0° ≤ direction ≤ 90°
tolerance	optional parameter to specify half of the opening angle with which points in the specified direction should be found; 0° < tolerance ≤ 45°
bandwidth	optional parameter to limit the tolerance angle to a certain maximum width

Pattern Analysis
`TablePattAnal`(*pntmap*)

pntmap	input point map with more than two points

INTERPOLATION

Densify map
`MapDensify`(*rasmap*, *factor*, *interpol meth*)

rasmap	raster map with a value domain for a BiLinear or Bicubic interpolation; raster map with any domain for Nearest Neighbour.
factor	real value > 1
interpol meth	`BiLinear`|`BiCubic`|`NearestNeighbour`

Contour Interpolation

`MapInterpolContour`(*segmap*, georef)

segmap	input segment map with a value domain

`MapInterpolContour`(*rasmap*)

rasmap	input raster map with a value domain; mind: algorithm only works well for rasterized contour lines

Point Interpolation

Nearest Point

`MapNearestPoint`(pntmap, georef)

Moving Average

`MapMovingAverage`(*pntmap*, georef, *weight func*)

pntmap	input point map with a value domain
weight func	`InvDist`(*Exp*,*LimDist*) | `Linear`(*Exp*,*LimDist*)
Exp	weight exponent
LimDist	limiting distance

Trend Surface

`MapTrendSurface`(*pntmap*, georef, *surface type*)

pntmap	input point map with a value domain
surface type	`Plane` | `Linear2` | `Parabolic2` | `2` | `3` | `4` | `5` | `6`

Moving Surface

`MapMovingSurface`(*pntmap*, georef, *surface type*, *weight func*)

pntmap	input point map with a value domain
surface type	`Plane` | `Linear2` | `Parabolic2` | `2` | `3` | `4` | `5` | `6`
weight func	`InvDist`(*Exp*,*LimDist*) | `Linear`(*Exp*,*LimDist*)
Exp	weight exponent
LimDist	limiting distance

Kriging

`MapKrigingSimple`(*pntmap*, georef, *semivarmodel*
[,errormap [, *remove duplic* [, *tolerance*]]])

`MapKrigingOrdinary`(*pntmap*, georef, *semivarmodel*, *limdist*
[,errormap [, *min*, *max* [, *remove duplic* [, *tolerance*]]]])

pntmap	input point map with a value domain
SemiVarModel	*Model(nugget, sill, range)* | `Power`*(nugget, slope, power)*
Model	`Spherical` | `Exponential` | `Gaussian` | `Wave` | `Circular` | `RatQuad`
limdist	limiting distance
min, max	optional parameter to specify the minimum and maximum number of points to be taken into account within the limiting distance
remove duplic	`no` | `average` | `firstval`, optional parameter to handle possible coinciding points

tolerance	optional parameter to specify a distance value in meters with which is determined whether points are coinciding or not

VECTOR OPERATIONS

Unique ID

`PolygonMapNumbering`(polmap [, *newdomain*])
`SegmentMapNumbering`(segmap [, *newdomain*])
`PointMapNumbering`(pntmap [, *newdomain*])

newdomain	optional parameter to specify a name for the output ID domain; if not specified, the output domain will be stored by the output map (internal domain).

POLYGONS

Attribute Map

`PolygonMapAttribute`(*polmap*, *column*)

polmap	polygon map with a Class, ID, or Group domain.
column	column with a Value, Class, ID, or Group domain; by default a column from the attribute table of the polygon map.

Mask Polygons

PolygonMapMask(polmap, *"mask"*)

"mask"	a mask consists of (multiple) search strings; asterisks and question marks can be used as wild cards; on the command line, the total mask needs to be surrounded by double quotes.

Assign Labels

`PolygonMapLabels`(polmap, pntmap)

Transform Polygons

`PolygonMapTransform`(polmap, coordsys)

SEGMENTS

Attribute Map

`SegmentMapAttribute`(*segmap*, *column*)

segmap	segment map with a Class, ID, or Group domain.
column	column with a Value, Class, ID, or Group domain; by default a column from the attribute table of the segment map.

Mask Segments

`SegmentMapMask`(segmap, *"mask"*)

"mask"	a mask consists of (multiple) search strings; asterisks and question marks can be used as wild cards; on the command line, the total mask needs to be surrounded by double quotes.

Assign Labels

`SegmentMapLabels`(segmap, pntmap [, *set domain*])

set domain	`yes` | `no` ; optional parameter to set the domain of the output segment map to the domain of the input segment map or to the domain of the input point map; if this parameter is not specified, the output segment map will use the same domain as the input segment map.
`no`	domain of output segment map is domain of input segment map.
`yes`	domain of output segment map is domain of input point map.

Sub Map

SegmentMapSubMap(segmap, minX, minY, maxX, maxY)

minX	minimum X-coordinate of output map
minY	minimum Y-coordinate of output map
maxX	maximum X-coordinate of output map
maxY	maximum Y-coordinate of output map

Glue Segment Maps

`SegmentMapGlue`(*segmap1*, *"mask1"*, *segmap2*, *"mask2"* [,...] [, *newdomain*])

`SegmentMapGlue`(*minX*, *minY*, *maxX*, *maxY*, *segmap1*,*"mask1"*, *segmap2*, *"mask2"* [, ...] [, *newdomain*])

segmap1,2, ...	are the input segment map names to be combined into one output segment map
"mask1"	a mask consists of (multiple) search strings; asterisks and question marks can be used as wild cards; on the command line, the total mask needs to be surrounded by double quotes.
minX	minimum X-coordinate of output map.
minY	minimum Y-coordinate of output map.
maxX	maximum X-coordinate of output map.
maxY	maximum Y- coordinate of output map.
newdomain	optional parameter, when merging Class or ID maps, to specify a name for the output domain into which all items of the input domains are merged; if not specified, the output domain will be stored by the output map (internal domain).

Densify Coords

`SegmentMapDensifyCoords`(segmap, *distance*)

distance	is the distance in meters at which extra intermediate points should be inserted into segments; real value > 0

Transform Segments

`SegmentMapTransform`(segmap, coordsys)

Tunneling

`SegmentMapTunneling`(segmap, *tunnel width*, *remove node*)

tunnel width	tunnel width in meters; real value >= 0
remove node	`yes` | `no` ; remove superfluous nodes or not

POINTS

Attribute Map

`PointMapAttribute`(*pntmap*, *column*)

pntmap	point map with a Class, ID, or Group domain
column	column with a Value, Class, ID, or Group domain; by default a column from the attribute table of the point map.

Mask Points

`PointMapMask`(pntmap, *"mask"*)

"mask"	a mask consists of (multiple) search strings; asterisks and question marks can be used as wild cards; on the command line, the total mask needs to be surrounded by double quotes.

Sub Map

`PointMapSubMap`(pntmap, minX, minY, maxX, maxY)

minX	minimum X-coordinate of output map
minY	minimum Y-coordinate of output map
maxX	maximum X-coordinate of output map
maxY	maximum Y-coordinate of output map

Glue Point Maps

`PointMapGlue`(*pntmap1*, *"mask1"*, *pntmap2*, *"mask2"* [,...] [, *newdomain*])

`PointMapGlue`(*minX*, *minY*, *maxX*, *maxY*, *pntmap1*, *"mask1"*, *pntmap2*, "mask2" [, ...] [, *newdomain*])

pntmap1,2, ...	are the input point maps to be combined into one output point map
"mask1"	a mask consists of (multiple) search strings; asterisks and question marks can be used as wild cards; on the command line, the total mask needs to be surrounded by double quotes.
minX	minimum X-coordinate of output map
minY	minimum Y-coordinate of output map
maxX	maximum X-coordinate of output map
maxY	maximum Y- coordinate of output map
newdomain	optional parameter, when merging Class or ID maps, to specify a name for the output domain into which all items of the input domains are merged; if not specified, the output domain will be stored by the output map (internal domain).

Transform Points

`PointMapTransform`(pntmap, coordsys)

RASTERIZE

Polygon to Raster

`MapRasterizePolygon`(polmap, georef)

Segment to Raster

`MapRasterizeSegment`(segmap, georef)

Segment Density

`MapSegmentDensity`(segmap, *"mask"*, georef)

"mask"	a mask consists of (multiple) search strings; asterisks and question marks can be used as wild cards; on the command line, the total mask needs to be surrounded by double quotes.

Point to Raster

`MapRasterizePoint`(pntmap, georef, *point size*)

point size	size in pixels; integer value > 0

Point Density

`MapRasterizePointCount`(pntmap, georef, *point size*)

`MapRasterizePointSum`(pntmap, georef, *point size*)

point size	size in pixels; integer value > 0

VECTORIZE

Raster to Polygon

`PolygonMapFromRas`(*rasmap* [, `8` | `4` [, `smooth` | `nosmooth`]])

rasmap	input raster map cannot have georef None
`8|4`	distinguish 8-connected or 4-connected areas; default is 8.
smooth	smooth polygon boundaries; default.
nosmooth	do not smooth polygon boundaries.

Raster to Segment

`SegmentMapFromRasAreaBnd`(*rasmap*, `8` | `4`, `smooth` | `nosmooth`, `single` | `composite`)

rasmap	input raster map cannot have georef `None`
`8	4`	distinguish 8-connected or 4-connected areas
`smooth`	smooth segments
`nosmooth`	do not smooth segments
`single`	assign the name 'Segments' to all output segments (internal output domain).
`composite`	use the names of the pixels on both sides of the output segment and construct composite names for the output segment like `Agri|Forest` (internal output domain).

Raster to Point

`PointMapFromRas`(*rasmap*)

rasmap	input raster map cannot have georef None

Polygon to Segment

`SegmentMapPolBoundaries`(polmap, *"mask"*, `single` | `composite`)

"mask"	a mask consists of (multiple) search strings; asterisks and question marks can be used as wild cards; on the command line, the total mask needs to be surrounded by double quotes.
`single`	assign the name 'Segments' to all output segments (internal output domain).
`composite`	use the names of the polygons on both sides of each output segment and construct composite names for the segments like `Agri` | `Forest` (internal output domain).

Polygon to Point

`PointMapPolLabels`(polmap [, `AlsoUndefs`])

`AlsoUndefs`	optional parameter to also obtain label points for polygons, which have no class name, ID or value.

Segment to Polygon

`PolygonMapFromSegment`(segmap [, *"mask"* [, `auto`]])

`PolygonMapFromSegment`(segmap [, *"mask"*, *domain* | *labelpntmap* [, `auto`]])

"mask"	a mask consists of (multiple) search strings; asterisks and question marks can be used as wild cards; on the command line, the total mask needs to be surrounded by double quotes; if not specified all segments are used.
domain	optional parameter to polygonize the segments to an existing domain; after polygonization, you can break the dependencies and edit the polygon map.
labelpntmap	optional parameter to polygonize the segments and use a point map with label points to assign names to the output polygons. If both the *domain* and *labelpntmap* parameters are not specified, the segments are polygonized and are assigned default names such as Pol 1, Pol 2, etc. (internal domain).
`auto`	optional parameter to automatically correct segments; deletes false polygons, deletes segments with dead ends, insert nodes when needed; if not specified and an error is found, the program stops and no polygon map is calculated.

Segment to Point

`PointMapSegCoords`(segmap)

`PointMapSegDist`(segmap, distance)

distance	distance interval in meters; real value > 0

`PointMapSegNodes`(segmap)

TABLE OPERATIONS

Transpose Table

`TableTranspose`(*table, col domain*)

`TableTranspose`(*table, col domain, valuerange*)

table	input table with a domain none, class or ID and maximum 1000 records
col domain	the domain that will be used for all columns in the transposed table
valuerange	if the column domain is a value domain, specify the value range as min:max : prec that will be used for all columns in the transposed table

Change Domain

`TableChangeDomain`(*table*, `None` | *domain*)
`TableChangeDomain`(*table*, *column*[, `avg` | `min` | `max` | `sum` | `last` | `no`])

domain	parameter to specify the name of an existing class or ID domain on disk. The output table will use this class or ID domain.
column	parameter to specify the name of a class or ID column in the input table; the output table will use the domain of the specified column. When no aggregation function is used, the classes or IDs in the specified column should be unique.
`avg` | `min` | `max` | `sum` | `last` | `no`	parameter to specify the aggregation method for other value columns, when the classes or IDs in the specified column are not unique: average, minimum value, maximum value, sum, last value encountered or no aggregation.

Table to PointMap

`PointMapFromTable`(table,[*colX*, *colY*,] coordsys [, *prefix* | *attribcol*])

colX, *colY*	parameters to specify names of columns in the table which contain the X- and Y-coordinates for the points; do not need to be specified when table contains columns with names X and Y.
prefix	optional parameter for a table with domain `None` to create a new ID domain for the output point map and table; if not specified, the new ID domain will use prefix `Pnt`.
attribcol	optional parameter to use a domain of a column in the table as the domain of the output point map.

Glue Tables

`TableGlue`(table1, table2 [, *more tables*] [, `vertical`])
`TableGlue`([*newdomain*,] table1, table2 [, *more tables*])

more tables	optional parameter to specify more input tables. You can specify as many input tables as you like; comma delimited.
`vertical`	in case the input tables have domain `None`: optional parameter to specify that tables should be merged vertically; if not specified, columns of the input tables will all appear as new columns in the output table (horizontally).

newdomain in case the input tables have class or ID domains: optional parameter to specify a name for the domain of the output table; if not specified; the output domain will obtain the same name as the output table.

Appendix D

ILWIS expressions (alphabetic)

Note: This appendix replaces Appendix E.2.2 in the ILWIS 2.1 Reference Guide.

In the listing below, all ILWIS expressions are ordered by output object type, and then alphabetical.

Operations resulting in a raster map

`MapAggregate`*AggFunc*(rasmap, *groupfactor*, `group` [,*rowoffset*, *coloffset*] [,*newgeoref*])
`MapAggregate`*AggFunc*(rasmap, *groupfactor*, `nogroup` [,*rowoffset*, *coloffset*])
`MapApply3D`(rasmap, georef3D)
`MapAreaNumbering`(rasmap, `8|4` [, *newdomain*])
`MapAttribute`(rasmap, attribute column)
`MapCalculate`(expression)
`MapClassify`(sample set, *classifier*)
`MapCluster`(*map list, nr of clusters*)
`MapColorComp24[Linear]`(*map list* [,*range1, range2, range3*])
`MapColorComp24HistEq`(*map list, range1, range2, range3*)
`MapColorComp24HSI`(*map list*)
`MapColorComp[Linear]`(*map list, range1, range2, range3*)
`MapColorCompHistEq`(*map list*, *range1*, *range2*, *range3*)
`MapColorSep[aration]`(*rasmap,color*)
`MapCross`(rasmap1, rasmap2, output cross table)
`MapDensify`(rasmap, factor, interpolation method)
`MapDistance`(source rasmap [, weight rasmap`|1` [, output Thiessen rasmap]])
`MapFilter`(rasmap, *filter* | *filter expression*)
`MapGlue`(rasmap1, rasmap2 [, *rasmaps*][,*newdomain*] [, `replace`])
`MapHeckbert`(map list, *nr of colors*)
`MapInterpolContour`(segmap, georef)
`MapInterpolContour`(rasmap)
`MapIter[Prop]`(*StartMap*, IterExpr [, *nr of iterations*])
`MapKrigingOrdinary(`*pntmap*, georef, *semivarmodel*, *limdist* [, errormap [, m*in*, *max* [, *remove duplic* [, *tolerance*]]]])
`MapKrigingSimple(`*pntmap*, georef, *semivarmodel* [, errormap [, *remove duplic* [, *tolerance*]]])
`MapNearestPoint`(pntmap, georef)
`MapMirrorRotate`(rasmap, *rotate type*)

MapMovingAverage(pntmap, georef, *weight function*)
MapMovingSurface(pntmap, georef, *surface type*, *weight function*)
MapRasterizePoint[Count | Sum](pntmap, georef, *pointsize*)
MapRasterizePolygon(polmap, georef)
MapRasterizeSegment(segmap, georef)
MapResample(rasmap, georef, resample method[, Patch|NoPatch])
MapSegmentDensity(segmap [, *"mask"*] , georef)
MapSlicing(rasmap, *domain* group)
MapStretch[Linear](rasmap, *range from*, *domain* [, *range to*])
MapStretchHistEq(rasmap, *range from*, *intervals*)
MapSubMap(rasmap, first row, first col [, nr rows, nr cols] [, *newgeoref*])
MapThiessen(source rasmap [, weight rasmap|1], output Distance rasmap)
MapTrendSurface(pntmap, georef, *surface type*)

Operations resulting in a polygon map

PolygonMapAttribute(polmap, attribute column)
PolygonMapFromRas(rasmap [, 8|4 [, smooth|nosmooth]])
PolygonMapFromSegment(segmap, *"mask"* [, *domain|labelpntmap*] [, auto])
PolygonMapLabels(polmap, pntmap)
PolygonMapMask(polmap, *"mask"*)
PolygonMapNumbering(polmap [, *newdomain*])
PolygonMapTransform(polmap, coordsys)

Operations resulting in a segment map

SegmentMapAttribute(segmap, attribute column)
SegmentMapDensifyCoords(segmap, *distance*)
SegmentMapFromRasAreaBnd(rasmap, 8|4, smooth|nosmooth, single|composite)
SegmentMapGlue(segmap1, *"mask1"*, segmap2, *"mask2"* [, ...] [,*newdomain*])
SegmentMapGlue(*minX*, *minY*, *maxX*, *maxY*,segmap1, *"mask1"*, segmap2, *"mask2"* [, ...] [,*newdomain*])
SegmentMapLabels(segmap, pntmap, *set domain*)
SegmentMapMask(segmap, *"mask"*)
SegmentMapNumbering(segmap [, *newdomain*])
SegmentMapPolBoundaries(polmap, *"mask"*, single|composite)
SegmentMapSubMap(segmap, *minX*, *minY*, *maxX*, *maxY*)
SegmentMapTransform(segmap, coordsys)
SegmentMapTunneling(segmap, *tunnel width*, *remove nodes*)

Operations resulting in a point map

PointMapAttribute(pntmap, attribute column)
PointMapFromRas(rasmap)
PointMapFromTable(table, [*colX*, *colY*,] coordsys [,*prefix* | *attribcolumn*])
PointMapGlue(pntmap1, *"mask1"*, pntmap2, *"mask2"* [, ...] [,*newdomain*])

`PointMapGlue`(*minX*, *minY*, *maxX*, *maxY*,pntmap1, *"mask1"*, pntmap2, *"mask2"* [, ...] [,*newdomain*])
`PointMapMask`(pntmap, *"mask"*)
`PointMapNumbering`(pntmap [, *newdomain*])
`PointMapPolLabels`(polmap | `AlsoUndefs`)
`PointMapSegCoords`(segmap)
`PointMapSegDist`(segmap, *distance*)
`PointMapSegNodes`(segmap)
`PointMapSubMap`(pntmap, *minX*, *minY*, *maxX*, *maxY*)
`PointMapTransform`(pntmap, coordsys)

Operations resulting in a table
`TableAutoCorrSemiVar`(rasmap, *pixel shift*)
`TableChangeDomain`(table, `None` | *domainondisk*)
`TableChangeDomain`(table, columnname [, `avg`| `min`| `max`| `sum`| `last`| `no`])
`TableCross`(rasmap1, rasmap2)
`TableCross`(rasmap1, rasmap2, output cross rasmap)
`TableCross`(rasmap1, rasmap2[, output cross rasmap] [,`IgnoreUndefs` | `IgnoreUndef1` | `IgnoreUndef2`])
`TableGlue`(table1, table2 [, *more tables*] [, `vertical`])
`TableGlue`([*newdomain*,] table1, table2 [, *more tables*])
`TableHistogram`(rasmap)
`TableHistogramPnt`(pntmap)
`TableHistogramPol`(polmap)
`TableHistogramSeg`(segmap)
`TableNeighbourPol`(polmap)
`TablePattAnal`(pntmap)
`TableSegDir`(segmap)
`TableSpatCorr`(pntmap[, *lagspacing* [, *direction* [, *tolerance* [, *bandwidth*]]]])
`TableTranspose`(table, column *domain* [,*value range*])

Operations resulting in a matrix
`MatrixFactorAnal`(map list)
`MatrixPrincComp`(map list)

Aggregation or join operations in tables resulting in a column
`ColumnAggregateAvg`(column [, *group* [, *weight*]])
`ColumnAggregateCnt`(column [, *group*])
`ColumnAggregateMax`(column [, *group*])
`ColumnAggregateMed`(column [, *group* [, *weight*]])
`ColumnAggregateMin`(column [, *group*])
`ColumnAggregatePrd`(column [, *group* [, *weight*]])
`ColumnAggregateStd`(column [, *group* [, *weight*]])
`ColumnAggregateSum`(column [, *group*])

`ColumnJoin`(table, column)
`ColumnJoin`(table, column, *key1*)
`ColumnJoin2ndKey`(table, column, *viakey2*)
`ColumnJoin2ndKey`(table, column, *key1*, *viakey2*)
`ColumnJoin`*`AggFunc`*(table, column, *groupbykey2*)
`ColumnJoin`*`AggFunc`*(table, column, *groupbykey2*, *weight*)
`ColumnJoin`*`AggFunc`*(table, column, *groupbykey2*, *weight*, *key1*)

Special syntax to create attribute maps
Map.column
Polygonmap.mpa.column
Segmentmap.mps.column
Pointmap.mpp.column

Special syntax to perform color separation
Map.color

Appendix E

ILWIS script language (syntax)

Note: This appendix replaces Appendix F in the ILWIS 2.1 Reference Guide.

A script is a sequenced list of ILWIS expressions. By creating a script, you can build a complete GIS or Remote Sensing analysis for your own research discipline. Scripts are more or less equivalent to batch files in ILWIS version 1.4.

General information

In a script, you can use any MapCalc or TabCalc expression, any expression for an operation, you can use parameters, you can call other scripts, you can display ILWIS objects on the screen, and further you can use a number of commands for file management, to handle object properties, to break dependencies and release disk space, to edit class or ID domains, etc.

When you run a script, no dialog boxes appear and no questions are asked; all lines in the script are simply performed. Error messages appear in case syntax errors are detected in a MapCalc expression, in a TabCalc expression, or in an expression for another operation, or in a script command. Further error messages appear when a script command is not recognized, or when required objects are not found. A script line is ignored when the syntax is correct and necessary objects are found but the command cannot be performed otherwise (e.g. creating objects that already exist, missing or wrong extensions during a copy).

Parameters in scripts

Parameters in a script can replace (parts of) object names, operations, etc. Parameters in scripts work as DOS replaceable parameters in DOS batch files, and must be written in the script as `%1, %2, %3,` up to `%9`. The parameters of a script have to be filled out on the command line when you run the script. The first text string found after the script name will replace `%1` in the script, etc. For more information, see How to use parameters in scripts.

To run a script

To run a script from the command line of the Main window, type:
Run scriptname parameter parameter.
If a script has no parameters, you can directly double-click the script in the Catalog. For more information, see How to run scripts.

Example
An example of a script is presented in Map and Table calculation : creating and running scripts.

☞ Single text lines of a script, i.e. the commands and expressions described below, can also be typed on the command line of the Main window. To avoid any dialog boxes, it is advised to use a semicolon (';') at the end of such a line. In a script, semicolons are not allowed.

Expressions for calculations and other operations

Most text lines in a script will consist of MapCalc and TabCalc expressions and expressions of operations that you can also type on the command line of the Main window or on the command line of a table window. You should be familiar with these expressions. For an overview of MapCalc and TabCalc operators and functions, refer to Appendices: MapCalc and TabCalc operators and functions. For an overview of expressions for other operations, refer to Appendices: ILWIS expressions. For more information on the creating of expressions, see Appendices: construction of expressions.

MapCalc and Tabcalc

For MapCalc expressions, no special syntax is required: you can simply type the MapCalc expression as you would type it on the command line of the Main window. For example, to `sum` maps `map1` and `map2` to create `map3`, type in the script:

```
map3=map1+map2
```

For TabCalc expressions, it is necessary that you type `tabcalc` ***`tablename`*** in front of the tabcalc expression. For example, to `sum` columns `col1` and `col2` in table `MyTable` and to store the results in column `col3`, type in the script:

```
tabcalc MyTable col3=col1+col2
```

If you like, you can also perform table calculations on other objects that can be opened as a table, e.g. histograms, point maps, class representations. Then, specify the extension of the object after the table name:

```
tabcalc tablename.ext a=b+c
```

If you want to perform a series of table calculations in one table, it is advised to use the following script commands:

```
opentbl tablename.ext
```

Keep the table *tablename.ext* open as the first line before a series of TabCalc expressions on one table.

```
closetbl tablename.ext
```

Close the open table *tablename.ext* as the last line after a series of TabCalc expressions on one table.

ILWIS operations

To perform other ILWIS operations, you can use any ILWIS expression as described in Appendices C: ILWIS expressions and D: ILWIS expressions (alphabetic).

E.1 Additional script commands

A number of additional script commands is available for file management, to show objects, handle object properties, edit object properties, create objects, calling other scripts, etc.

In many of the following script commands, object names and extensions of their object definition files must be specified. In some script commands, you are allowed to use wildcards `*` and `?` to specify object names and their extensions (*object.ext* and *table.ext.col*).

When using a script command that works on a column in a table (*table.ext.col*), you can ignore the extension when the table has extension .TBT. Table extensions only need to be specified when the column is stored in a histogram, a point map, a class representation, a georeference tiepoints, etc.

Further, in the list below, optional parameters of script commands are shown between square brackets. Omit these square brackets when writing a script. Square brackets are only recognized for TabCalc expressions to indicate a specific record in a table.

Remarks, comments, messages and pause

`rem` *This is a remark* — All text on this line after `rem` is ignored by the script. In this manner, you can document your script expressions.

`//` *This is a remark* — All text on this line after `//` is ignored by the script. In this manner, you can document your script expressions.

`begincomment`
line 1 of my multiple line comment
line 2 of my multiple line comment
`endcomment` — All lines of text between the commands `begincomment` and `endcomment` are ignored by the script. In this manner, you can document your script expressions and you can temporarily exclude parts of your script.

`message` *Message on my screen*
Obtain a message box on the screen with any text; the text can be as long as you like. After pressing the OK button in the message box, the script will continue. In this manner, you can display texts on the screen during demos, etc.

`pause` *seconds* — Stop the script for a certain amount of time (seconds). You can use this command for instance when you want to show multiple maps and give the user time to view each one of them.

Open/Show an object

`show` *object.ext*	Open/show object *object.ext.*
`open` *object.ext*	Open/show object *object.ext.*
`open –noask` *object.ext*	Open/show object *object.ext* with its default display options, i.e. a Display Options dialog box will be skipped. You can use this command for instance to quickly show a map on the screen.
`closeall`	Close all ILWIS windows except the ILWIS Main window. You can use this command for instance to close any map and/or table windows being displayed on the screen.

File management

`cd` *path*	Change directory to directory *path.*
`cd` *d:path*	Change drive to *drive d*: and change directory to directory *path.*
`md` [*drive*:]*path*	Make directory *path*. Optionally make directory *path* on drive *drive*
`mkdir` [*drive*:]*path*	Make directory *path*. Optionally make directory on *path* on drive *drive.*
`rd` [*drive*:]*path*	Remove directory *path*. Optionally remove directory *path* from drive *drive.*
`rmdir` [*drive*:]*path*	Remove directory *path*. Optionally remove directory *path* from drive *drive.*
`rd` [*drive*:]*path* `-force`	Remove directory *path* while deleting all files in that directory. Optionally remove directory *path* and all files in that directory from drive *drive.*
`rmdir` [*drive*:]*path* `-force`	Remove directory *path* while deleting all files in that directory. Optionally remove directory *path* and all files in that directory from drive *drive.*

An error message appears when changing to a directory that does not exist, or when removing a directory that does not exist. The script line is ignored, when making a directory that already exists, or when deleting files that do not exist.

`copy` *object.ext objname*	Copy object *object.ext* to new name *objname*.
`copy` *object.ext objname* `-breakdep`	
	Copy object *object.ext* to new name *objname* while breaking the dependency links of *object.ext*.
`copy` *object.ext path*	Copy object *object.ext* to existing directory *path*. Wildcards are allowed.
`copy` *object.ext path* `-breakdep`	
	Copy object *object.ext* to existing directory

	path while *breaking the dependencies of object.ext*. Wildcards are allowed.
`copyfile` *file.ext filename.ext*	Copy file *file.ext* to new file *filename.ext*.
`copyfile` *file.ext path*	Copy file *file.ext* to existing directory *path*. Wildcards are allowed.

When copying objects (or files), you cannot copy objects to another directory and give the object another name at the same time.
When copying an object to another directory, existing objects in that directory are not overwritten. In the same way, when copying an object in the current directory to an object name, which already exists, the script line is ignored.

`del` *object.ext*	Delete object *object.ext*. Wildcards are allowed.
`del` *object.ext* `-force`	Tries to delete object *object.ext* which is not completely valid (i.e. an error occurs when the object is opened). Wildcards are allowed.
`delcol` *table.ext.column*	Delete column *table.ext.column*.
`delfile` *file.ext*	Delete file *file.ext* as if this file was deleted in DOS or the File Manager. Wildcards are allowed.

The del and delcol commands check whether the object is not read-only or whether a column is not table-owned; these commands do not take into account whether an object is still used by other objects. The script line is ignored when objects, columns or files do not exist.

Handling properties of dependent objects

`update` *object.ext*	Make the dependent map or table *object.ext* up-to-date. Wildcards are allowed.
`updatecol` *table.ext.column*	Make dependent column *table.ext.column* up-to-date.
`breakdep` *object.ext*	Break the dependency links of dependent map or table *object.ext*. Wildcards are allowed.
`breakdep` *object.ext* `-force`	Tries to break the dependency links of dependent map *object.ext* which is not completely valid (i.e. an error occurs when the object is opened). Wildcards are allowed.
`breakdepcol` *table.ext.column*	Break the dependency links of dependent column *column* in table *table.ext*.
`reldisksp` *object.ext*	Delete the data file(s) of dependent object *object.ext*.
`calc` *object.ext*	Recalculate the data files of dependent map or table *object.ext*. Wildcards are allowed.
`calccol` *table.ext.col*	Recalculate dependent column *table.ext.col*.

Editing properties of editable source objects (advanced)

`changedom` *object.ext domname* [*valuerange*]
: Change the domain of raster, polygon, segment or point map *object.ext* to existing domain *domname*, while converting the class names, IDs, or values of the original domain into new domain *domname*. Optionally, in case of a value domain *domname*, set the value range of the object to valuerange. Specify the value range as *min:max:precision*, as *min:max*, or as *::precision*. This is a rather safe way to change the domain of a map into another domain or to change the precision of a map.

`setdom` object.ext *domname* [*valuerange*] [`-force`]
: Set the domain of *object.ext* to *domname*. Wildcards are allowed for *object.ext*. Optionally, in case of a value domain *domname*, set the value range of the object to *valuerange*. Specify the value range as *min:max:precision*, as *min:max*, or as *::precision*. When you also specify the `-force` flag, no checks are performed.

`setvr` *object.ext valuerange*
: Set the value range of *object.ext* to *valuerange*. Wildcards are allowed for *object.ext*. Specify the value range as *min:max:precision*, as *min:max*, or as *::precision*.

`setgrf` *rasmap georef*
: Set the georeference of raster map *rasmap* to *georef*. Wildcards are allowed for *rasmap*.

`setcsy` *map.ext coordsys*
: Set the coordinate system of point, segment or polygon map *map.ext* to *coordsys*. Wildcards are allowed for *map.ext*.

`setcsy` *georef coordsys*
: Set the coordinate system of georeference *georef* to *coordsys*. Wildcards are allowed for *georef*.

`setreadonly` *object.ext*
: Mark object *object.ext* as read only. Read only objects cannot be edited or deleted. Wildcards are allowed.

`setreadwrite` *object.ext*
: Remove the read only flag for object *object.ext*: the objects are editable and deleteable. Wildcards are allowed.

`setatttable` *map.ext atttable*
: Set the attribute table of class or ID map *map.ext* to *atttable*. Wildcards are allowed for *map.ext*.

`setatttable` *map.ext*	Remove the link between class or ID map *map.ext* and its attribute table. Wildcards are allowed.

The `setdom`, `setvr`, `setgrf`, `setscy` and `setatttable` commands are only performed on objects that are not read only.

Creating (empty) maps and tables

`crmap` *map georef domname*	Create raster map map with existing georeference georef and existing domain domname.
`crsegmap` *map crdsys domname*	Create segment map with existing coordinate system crdsys and existing domain domname.
`crpntmap` *map crdsys domname*	Create point map map with existing coordinate system crdsys and existing domain domname.
`crtbl` *table domname*	Create table table using existing Class or ID domain domname.
`crtbl` *table nrrecs*	Create table table using domain None and with a number of records specified as nrrecs.

Creating domains

`crdom` *domname* [`-type=class|ID|group`] `-items`=*number* [`-prefix`=prefix]

Create domain *domname* with a number of items specified as `-items`=*number*. Optionally, specify `-type=class` to create a class domain, or `–type=ID` to create an ID domain, or `–type=Group` to create a group domain. If parameter `-type` is omitted, then a class domain is created. Optionally specify `-prefix=`*prefix* to obtain classes or IDs with a certain *prefix*. If parameter `-prefix` is omitted when creating a class domain, classes will obtain prefix `class`, thus `class 1`, `class 2`, etc. If parameter `-prefix` is omitted when creating an ID domain, the IDs will obtain prefix `ID`, thus `ID 1`, `ID 2`, etc.

`crdom` *domname* `-type=value min`=*value* max=*value* [`-prec`=*value*]

Create value domain *domname* with a specified value range between `min=`*number* and `max=`*number*. Optionally, specify a precision for the value domain as `-prec`=value.

Examples:

`crdom` *domname* `-type=class -items=10 -prefix=cl`
: Create class domain *domname* and add ten items to this domain with class names "cl 1", "cl 2", .. "cl 10".

`crdom` *domname* `-type=id -items=100 -prefix=prov`
: Create ID domain *domname* and add hundred items to this domain with IDs "prov 1", "prov 2", .. "prov 100".

`crdom` *domname* `-items=0`
: Create class domain *domname* without any items. You can add items with the `additemtodomain` command.

`crdom` *domname* `-type=value -min=100 -max=200`
: Create value domain *domname* with a value range between 100 and 200 (precision is 1).

`crdom` *domname* `-type=value -min=10 -max=20 -prec=0.01`
: Create value domain *domname* with a value range between 10.00 and 20.00 and a precision of 0.01.

The `crdom` command is ignored when domain domname already exists.

Editing a class or ID domain

`additemtodomain` *domname* *class* [*classcode*]
: Add item class, optionally with code *classcode*, to class or ID domain *domname*. Class names, which contain spaces, must be enclosed by double quotes.

`additemtodomaingroup` *domname upperlimit class [classcode]*
: Add an item to group domain *domname*. The item is defined by an *upperlimit* (a real value), a class name *class* and, optionally, a *classcode*

`mergedom` *domname1 domname2*
: Merge the items of class or ID domain *domname2* into class or ID domain domname1.

Create representations

`crrpr` *rprname domname*
: Create representation *rprname* for class or value domain *domname*.

The `crrpr` command is ignored when representation *rprname* already exists.

Create georeference corners

`crgrf` *grfname nrrows nrcols* `[-crdsys=`coordsysname`]`
        `-lowleft=(`*minX,minY*`)`
        `-upright=(`*maxX,maxY*`)` | `-pixsize=`*value*
        `[-centercorners+]`

Create georeference corners *grfname* with the number of rows as specified by *nrrows*, the number of columns as specified by *nrcols*, the coordinates for the lower left corner as specified by `-lowleft`=(*minX*, *minY*), and either the coordinates for the upper right corner specified by `-upright`=(*maxX*, *maxY*) or the pixel size as specified by `-pixsize`=value.
Mind: in a script, you are not allowed to use spaces within a coordinate expression (X,Y). Optionally, a coordinate system *coordsysname* can be specified by using parameter `–crdsys` =*coordsysname*, otherwise the system coordinate system `Unknown` will be used. Furthermore, you can optionally add coordinates to the centers of the corner pixels by using parameter `-centercorners+`. When this parameter is not used, coordinates will be added to the corners of the corner pixels.

Examples:

`crgrf` *grfname* `500 1000 -crdsys=cs -lowleft=(0,0) –upright =(5000,10000)`

Create georeference corners *grfname* with 500 rows and 1000 columns and using coordinate system `cs`. The coordinate boundaries are defined by the lower left coordinate (0,0) and the upper right coordinate (5000,10000).

`crgrf` *grfname* `500 1000 -crdsys=cs -lowleft=(0,0) –pixsize=10`

Create georeference corners *grfname* with 500 rows and 1000 columns and using coordinate system `cs`. The georeference has as lower left coordinate (0,0) and as pixel size 10 m.

Mind: In a script, you are NOT allowed to use spaces within a coordinate expression (X,Y). The `crgrf` command is ignored when georeference *grfname* already exists.

Creating a two-dimensional table

`cr2dim` *2dimtablename indomname1 indomname2 outdomname3 [valuerange]*

Create two-dimensional table *2dimtablename* using existing input domains *indomname1* and *indomname2* and use existing domain *outdomname3* as the domain of the fields in the table. If *outdomname3* is a value domain, you can optionally specify the value range of this domain as *min:max* or as *min:max:precision*.

Converting domains

`domclasstoid` *domname[.ext]*
: Convert class domain *domname* into an ID domain.

`domidtoclass` *domname[.ext]*
: Convert ID domain *domname* into a class domain.

`dompictoclass` *domname[.ext]*
: Convert Picture domain *domname.ext* into a class domain.

These are rather safe ways to convert one domain into another. When the domain you want to convert is an internal domain, which is stored in a map, you need to specify the extension of the map after the domain name.

Calling other scripts

`run` *script2*	Run another script named *script2* (without parameters).
`run` *script2 param1 param2*	Run another script name *script2*; fill out parameters.

If *script2* is not found, an error message appears.

Start other Windows applications

`!`*Command line*	Performs *Command line* as if entered in the Windows (File) Run dialog box. Starts any Windows application, batch file, or DOS application (with a .PIF file available). Applications should have one of the following extensions: `.exe`, `.com`, `.bat`, `.pif`. Type the application name directly after the exclamation mark (no spaces allowed). Example: to start Word and open document MyDoc, type: `!Winword MyDoc`

Import files from ILWIS 1.4 to ILWIS 2.

`Import14` *file14.ext* [*outputdir*]
: Batch-wise import of *file14.ext* in the current directory, or optionally to the specified output directory *outputdir*. Wildcards are allowed. The ILWIS 2 object(s) keep the name(s) of the 1.4 file(s); new extensions are created

during import. Domains, representations, georeferences etc. are created using defaults.

`Import14` *file14.ext* [*ilwis2name*|*outputdir*] [`-dmt`=*domtype*] [`-dom`=*domainname*] [`-grf`=*georef*] [`-att`=*tablename*]

Import an ILWIS 1.4 file *file14.ext* according to your wishes. All parameters shown above between square brackets can but do not have to be used.

ilwis2name|*outputdir*	Either specify an ILWIS 2 object name as *ilwis2name* for the 1.4 file to be imported or specify an output directory as *outputdir* in which the imported object should appear. If this parameter is omitted, then *file14.ext* is imported in the current directory and the ILWIS 2 object(s) keep the name(s) of the 1.4 file(s).
`-dmt`=*domtype*	Specify the domain type *domtype* for the output object as: `Picture`, `Image`, `Value`, `Class`, or `ID`. If you specify domain type Class or ID, you also have to use the `-dom` option to specify the name of that domain.
`-dom`=*domname*	If you specified domain type `Class` or `ID` in the previous option, then also specify a new or existing *domname* for the output object.
`-grf`=*georef*	When importing a raster map, you can specify the name of a new or existing georeference *georef*.
`-att`=*tablename*	When importing a 1.4 point table which contains besides the `X!`, `Y!` and `Name$` columns also attribute columns, you can specify the name of the ILWIS 2 attribute table as *tablename*. The point table is then imported as an ILWIS 2 map and the other columns (i.e. columns other than `X!, Y!` and `Name$`) are imported as an attribute table of the point map.

Example:

```
Import14 orotm4.mpd -dmt=image -grf=tmgeoref
```

Import 1.4 raster map `Orotm4` as an image using existing georeference tiepoints `tmgeoref`.

Importing files into ILWIS

`import` format(file.ext, ilwobj) Import a data file *file.ext* into an ILWIS object with the name *ilwobj*. The extension of the input file *file.ext* **must** be specified. The extension for the output ILWIS object(s) *ilwobj*

	will be automatically created during import. For *format*, you have to specify one of the following formats: `arcinfonas` | `ascii` | `bmp` | `bna` | `dbase` | `dxf` | `e00` | `erdas` | `gif` | `ida` | `idrisi` | `ilwis14pnt` | `ilwis14pol` | `ilwis14ras` | `ilwis14seg` | `ilwis14tbl` | `infocam` | `lin` | `pcx` | `shape` | `smt` | `tiff`.
`arcinfonas`	Import an Arc/Info non-compressed ASCII raster file to an ILWIS raster map.
`ascii`	Import an ILWIS 1.x ASCII raster file (.ASC) to an ILWIS raster map.
`bmp`	Import a Windows bitmap (.BMP) to an ILWIS raster map.
`bna`	Import an Atlas vector data file (.BNA) to an ILWIS segment map.
`dbase`	Import a dBase III/IV file (.DBF) to an ILWIS table. This option will come up with the Import dBase III/IV table dialog box.
`dxf`	Import an AutoCad .DXF file to an ILWIS point and/or segment and/or polygon map.
`e00`	Import an Arc/Info file in interchange format (.E00) to an ILWIS raster and/or polygon and/or segment and/or point map. When attributes are available, also an ILWIS table will be created.
`erdas`	Import an Erdas .GIS file into an ILWIS raster map; or import an Erdas .LAN file into a single ILWIS raster map or into an ILWIS map list containing multiple raster maps (bands).
`gif`	Import a gif image (.GIF) to an ILWIS raster map.
`ida`	Import an IDA image (.IMG) to an ILWIS raster map.
`idrisi`	Import an Idrisi image (.DOC). to an ILWIS raster map.
`ilwis14pnt`	Import an ILWIS 1.x point table (.PNT) to an ILWIS 2 point map.
`ilwis14pol`	Import an ILWIS 1.x polygon map (.POL) to an ILWIS 2 polygon map.
`ilwis14ras`	Import an ILWIS 1.x raster file (.MPD) to an ILWIS 2 raster map.
`ilwis14seg`	Import an ILWIS 1.x segment maps (.SEG) to an ILWIS 2 segment map.
`ilwis14tbl`	Import an ILWIS 1.x table (.TBL) to an ILWIS 2 table.

`infocam`	Import an Infocam sequential file (.SEQ) to an ILWIS point and/or segment and/or polygon map. When attributes are available, also an ILWIS table will be created.
`lin`	Import an Arc/Info file created with the Ungenerate command (.LIN and .PTS) to an ILWIS segment and/or point map. When the extension of the Arc/Info file is .PTS, you will obtain a point map, otherwise you will obtain a segment map.
`pcx`	Import a PaintBrush image (.PCX) to an ILWIS raster map.
`shape`	Import Arc/View Shape files (.SHP, .SHX, DBF) to an ILWIS point and/or segment and/or polygon map. Furthermore, an ILWIS table will be created.
`smt`	Import an ILWIS 1.x ASCII vector file (.SMT) to an ILWIS segment map.
`tiff`	Import a Tiff image (.TIF) to an ILWIS raster map. When the Tiff image contains GeoTiff information, a georeference will be created for the imported map; furthermore, it is attempted to create a coordinate system (you can find projection information in the description of the coordinate system).

Examples:

`import` erdas(soil.gis, ilwsoil)

Imports the Erdas file SOIL.GIS into ILWIS; a raster map with the name ILWSOIL.MPR will be created. The import will also look for the availability of an Erdas file called SOIL.TRL that may accompany the .GIS file.

`import` e00(parcel.e00, ilwparc)

Imports the Arc/Info interchange file PARCEL.E00 into ILWIS; a segment map, a polygon map, a raster map and a point map can be created (all named ILWPARC), where the vector maps can also be linked to attribute tables.

For more information on Import, see also the Import dialog box, or the `import.def` file in your ILWIS directory.

☞ Besides using the Import command in a script, you can also use it within ILWIS on the command line; then, the complete command must be followed by a semicolon.

Export files from ILWIS 2 to ILWIS 1.4

`export14` *object2.ext name14*

Export ILWIS 2 raster map, polygon map, segment map, point map or table *object2.ext* to an ILWIS 1.4 file *name14*.

Exporting ILWIS 2 maps and tables

`export` *format*(*ilwobj.ext*, *filename*)

Export an ILWIS 2 map or table *ilwobj.ext* to another data file *filename*.The extension of the ILWIS 2 map or table *ilwobj.ext* **must** be specified; `.mpl` for a map list, `.mpr` for a raster map, `.mpa` for a polygon map, `.mps` for a segment map, `.mpp` for a point map and `.tbt` for a table. Furthermore, ILWIS 2 objects which are stored as a table can be exported as tables: histograms (`.his`, `.hsa`, `.hss`, `.hsp`), 2-dimensional tables (`.ta2`), domain class/ID/group (`.dom`), representation class (`.rpr`), georeference tiepoints (`.grf`), coordinate system tiepoints (`.csy`). The extension(s) for the output data file(s) filename will be automatically created during export. For format, you have to specify one of the following formats: `arcinfonas` | `arcinfopts` | `ascii` | `bmp` | `bna` | `cartcv` | `cartvte` | `dbase` | `dbasesdf` | `delimited` | `dxf` | `e00` | `erdasdig` | `erdasgis` | `erdaslan` | `geosoft` | `gif` | `gina` | `hpgl` | `ida` | `idrisi` | `ilwis14` | `infocam` | `lin` | `pcx` | `shapefile` | `sif` | `smt` | `themak2` | `tiff` | `usemap`.

`arcinfonas` — Export an ILWIS raster map to an Arc/Info non-compressed ASCII file (.NAS). Internally, the Convert14 program is used.

`arcinfopts` — Export an ILWIS point map to an Arc/Info .PTS (ASCII) file. From this file, (label) points can be created using the Arc/Info Generate command. Instead of the command `arcinfopts`, you can also use the command `arcgen`.

`ascii` — Export an ILWIS raster map to an ILWIS 1.x ASCII file.

`bmp` — Export an ILWIS raster map to a Windows bitmap (.BMP).

`bna`	Export an ILWIS segment map to an Atlas vector map (.BNA).
`cartcv`	Export an ILWIS polygon or segment map to a Cart/o/graphix .CV file. Internally, the Convert14 program is used.
`cartvte`	Export an ILWIS polygon or segment map to a Cart/o/graphix .VTE file. Internally, the Convert14 program is used.
`dbase`	Export an ILWIS table to a dBase III/IV file (.DBF).
`dbasesdf`	Export an ILWIS table to an ASCII dBase III/IV file (.SDF).
`delimited`	Export an ILWIS table to an ASCII comma delimited file (.TXT).
`dxf`	Export an ILWIS polygon, segment or point map to an AutoCad .DXF file.
`e00`	Export an ILWIS polygon, segment or point map to an Arc/Info .E00 file.
`erdasdig`	Export an ILWIS polygon or segment map to an Erdas .DIG file. Internally, the Convert14 program is used.
`erdasgis`	Export an ILWIS raster map to an Erdas .GIS file; in case you exported a domain Class map, also a trailer file will be created (.TRL). The ILWIS map should preferably be north-oriented.
`erdaslan`	Export an ILWIS map list, which contains multiple bands of a satellite image, or export a single ILWIS raster map to an Erdas .LAN file. The ILWIS map(s) should preferably be north-oriented.
`geosoft`	Export an ILWIS raster map to a Geosoft Grid file (.GRD). Internally, the Convert14 program is used.
`gif`	Export an ILWIS raster map to a GIF image (.GIF). Internally, the Convert14 program is used.
`gina`	Export an ILWIS segment map to a Gina file (.GIA). Internally, the Convert14 program is used.
`hpgl`	Export an ILWIS segment map to an HPGL file (.HPG). Internally, the Convert14 program is used.
`ida`	Export an ILWIS raster map to an IDA image (.IMG). Internally, the Convert14 program is used.

`idrisi`	Export an ILWIS raster map to an Idrisi map (.DOC, .IMG).
`ilwis14`	Export an ILWIS raster, polygon, segment, point map or table to an ILWIS 1.x raster, polygon, segment, point map or table (.MPD, .POL, SEG, PNT, TBL and others).
`infocam`	Export an ILWIS polygon, segment or point map to an InfoCam sequential file (.SEQ).
`lin`	Export an ILWIS segment map to an Arc/Info .LIN file (ASCII). From this file, lines can be created using the Arc/Info Generate command.
`pcx`	Export an ILWIS raster map to a PaintBrush image (.PCX). Internally, the Convert14 program is used.
`shapefile`	Export an ILWIS polygon, segment or point map to Arc/View Shape files (.SHP, .SHX, .DBF).
`sif`	Export an ILWIS segment map to an ASCII Intergraph file in Standard Interchange Format (.SIF). Internally, the Convert14 program is used.
`smt`	Export an ILWIS segment map to an ILWIS 1.x ASCII vector file (.SMT).
`themak2`	Export an ILWIS raster, polygon, or segment map to a Themak2 file (.THR, .THP, THS). Internally, the Convert14 program is used.
`tiff`	Export an ILWIS raster map to a Tiff image (.TIF).
`usemap`	Export an ILWIS polygon map to a UseMap file (.USE). Internally, the Convert14 program is used.

Examples:

```
export e00(water.mps, e00water)
```

Exports the ILWIS segment map WATER.MPS to the file E00WATER.E00 in the Arc/Info interchange format. This example also illustrates the need for the explicit mentioning of the ILWIS map extension: as the E00 export is defined for segments, polygons and points there is no other way than the extension to tell export which type should be exported.

```
export erdasgis(soil.mpr, erdsoil)
```

Exports the ILWIS raster map SOIL.MPR to the file ERDSOIL.GIS. If the ILWIS map has a class domain, ILWIS will also create a SOIL.TRL file.

For more information, see also the Export dialog box, or the `expras.def`, `expmpl.def`, `exppol.def`, `expseg.def`, `exppnt.def`, `exptbl.def` files in your ILWIS directory.

☞ Besides using the Export command in a script, you can also use it within ILWIS on the command line; then, the complete command must be followed by a semicolon.

Appendix F

Errata ILWIS 2.1 Reference Guide

p. 39 In the first tip ☞, the first line was left out. The complete tip is:

You can increase or decrease the number of characters shown by Info in a map window and in the map editors by setting the 'width' in the properties of the domain. The default width for a class domain is 15; the default width for an identifier domain is 6.

p. 54 Replace: Also, when only the object definition file of a dependent object exists, … pixel info is able to …

With: **Special functionality**

When only the object definition file of a dependent map exists, i.e. when a dependent map has not been calculated yet and the data file of the dependent map does not yet exist, then pixel info is able to …

After paragraph "Working with dependent maps, … stored on disk

Insert: Furthermore, when for value raster maps you marked the 'Interpolation' check box in the Properties dialog box of a raster map, the Pixel info window will show interpolated values on sub-pixel level. For more information, refer to the Raster Map Properties dialog box.

Finally, you can also add coordinate systems to the pixel information window. When a transformation is possible between the coordinate system of the current map and the added coordinate system, the pixel information window will show coordinates transformed to the new coordinate system, i.e. a transformation is performed on the fly. In this way you can already see the results of a transformation to another coordinate system before you actually perform this transformation, thus before using a Transform operation or the Resample operation, and you can compare different coordinate systems with each other.

For more information, see possible coordinate system transformations and an example of adding multiple coordinate systems in the pixel information window.

p. 131 Segment editor Menu commands:

In the File menu, insert above Polygonize: Remove Redundant Nodes

p. 163, p. 168, p. 209, App. 30

New arithmetic operator:

^	*a*^*b*	exponential operator; *a*^*b* = POW(*a*,*b*) = a^b

p. 164, p. 222, App. 34

Statistical functions on Value columns have been extended:

AVG(*col*)	average of column *col*
MIN(*col*)	minimum of column *col*
MAX(*col*)	maximum of column *col*
SUM(*col*)	sum of column *col*
STD(*col*)	standard deviation of column *col*
STDEV(*col*)	standard deviation of column *col*
STERR(*col*)	standard error of column *col*
VAR(*col*)	variance of column *col*
COV(*col1, col2*)	covariance of column *col1* and column *col2*
COVAR(*col1, col2*)	covariance of column *col1* and column *col2*
CORR(*col1, col2*)	correlation of column *col1* and column *col2*
TTEST(*TrueValue, col)*	
	Student's *t*-test on column *col* where *TrueValue* is the value to which the mean of your variable stochastically converges (m).
CHISQUARE(*col, colExpValue*)	
	C^2 -test on the observed values (f_i) in column *col* where *colExpValue* is the column with the expected values (F_i)

p. 164, p. 236, App. 35

New Conversion functions CODE(*s*) and NAME(*s*); changed description VALUE(*s*) function:

CODE(*s*)	returns the code of *s* as a string; *s* must have a Class or ID domain; when there is no code, an empty string is returned
NAME(*s*)	returns the name of *s* as a string; *s* must have a Class or ID domain
VALUE(*s*)	returns numbers in string *s* as a value

p. 183, p. 231, App. 31, App. 34

Calculating with undefineds:

ISUNDEF(*a*)	tests whether *a* is undefined.
IFUNDEF(*a*,*b*)	if condition *a* is undefined, then return the outcome of expression *b*, else return *a*.

IFUNDEF(*a*,*b*,*c*)	if condition *a* is undefined, then return the outcome of expression *b*, else return the outcome of expression *c*.
IFNOTUNDEF(*a*,*b*)	if condition *a* is not undefined, then return the outcome of expression *b*, else return undefined.
IFNOTUNDEF(*a*,*b*,*c*)	if condition *a* is not undefined, then return the outcome of expression *b*, else return the outcome of expression *c*.

p. 195 Reads: `IFF(Demnew > 3340, Start, NBMAX(Start1#))`
Should read: `IFF(Demnew > 3340, Start, NBMAX(Start#))`

p. 227

INMASK(*s*, "*mask*")	tests whether *s* conforms to one of the strings specified in the "*mask*". Argument *s* can be a column name using domain type class, ID, group or string, or the outcome of an expression using domain type class, ID, or group or a string. In the mask, you can specify multiple search strings, each separated by a comma, and you can use wildcards ? and *. The total mask has to be enclosed by double quotes.

p. 228 Reads: `Result3=IFF( …,(”not suitable”)`
Should read: `Result3=IFF( …,”not suitable”)`

p. 243 Reads: `? MAPVALUE(tmb1, coord(800000,8080000),cochabam)`
Should read: `? MAPVALUE(tmb1, coord(800000,8080000,cochabam))`

p. 248 Reads: `Altitude = MapValue(DEM.mpr, PNTCRD(COORD(X,Y)))`
Should read: `Altitude = MapValue(DEM, COORD(X,Y))`

p. 270: Pictures of input and output map should be switched.

p. 278, 280, 281
By specifying a Row Offset and/or a Column Offset, the operation does not *start* from the specified row or column number but the specified number of rows and columns are *skipped*.

p. 282-289
In all topics on Distance calculation, insert:
In the Distance calculation dialog box, a source map can be any raster map with a class or ID domain. On the command line, you can also use raster maps with a value domain.

p. 337 Reads: `Hue = 255/PI * …`
Should read: `Hue = 255/2PI * …`

p. 465 Under Dialog box options, after ‘Input point map’, insert:

Attribute: In case the input point map is linked to an attribute table, select this check box and select an attribute column from the attribute table linked to obtain a raster map from the attribute values of the points.

p. 466 At the top of this page, after ‘Georeference’, insert:

Value range: In case the input and output map use a value domain, accept the default value range, or specify your own range of possible values in the output map.

Precision: In case the input and output map use a value domain, accept the default precision of output values, or specify your own precision.

p. 516 Replace the explanation under point 5 with:

To calculate a slope map in percentages from these maps DX and DY, type on the command line of the Main window:

```
SLOPEPCT = 100 * HYP(DX,DY)/ PIXSIZE(DEM)
```

HYP is an internal Mapcalc/Tabcalc function.
Function PIXSIZE returns the pixel size of a raster map; for DEM, fill out the name of your DEM created in step 2.
SLOPEPCT is the output map name of the slope map in percentages.

p. 563 How to edit maps

Replace: In a polygon map, new polygons can be inserted and existing ones can be deleted, codes of polygons can be edited, the shape of a polygon boundary can be changed and unnecessary polygon boundaries can be deleted;

With: In a polygon map, the class names, IDs, or values of selected polygons can be edited, labels can be created from polygons, labels can be applied to polygons and segment can be extracted from polygons;

p. 577 First formula: `Landuse.hsp.Area`
Should read: `Landuse.hsa.Area`

App. 34, 35

In section For maps and columns of domain Class, ID, or Group, insert:
INMASK(*s*, "*mask*")
For an explanation of this function, see above.

App. 34 Reads: IN *s2* IN *s1*; tests whether *s2* is part of *s1*
Should read: IN(*s1*,*s2*) tests whether *s2* is part of *s1*

Appendix G

Software License Agreement

The International Institute for Aerospace Survey and Earth Sciences (ITC), P.O. Box 6, 7500AA, Enschede, The Netherlands (hereinafter referred to as ITC) herewith states that the use of the software program on the enclosed CD is subject to the terms of this ILWIS License Agreement. You should not open this packet until you have read the terms written below. By opening the packet, you signify that you have read this agreement and accept its terms.

1. DEFINITIONS
'Licensed Software' shall be the product containing the software - in whole or in part - as described on the Order Form and any updates furnished by ITC in connection with the licensed Software, including basic and related materials pertinent to the said software. 'Derivative Software' shall be the 'Licensed Software' which has been merged into other software materials to form a 'Derivative Work'.

2. LICENSE
The license granted under this Agreement authorizes the Licensee on a non-exclusive and non-transferable basis to use the Licensed Software and any Derivative Work only within and on behalf of the Organisation as described above.

3. RESTRICTION
Licensee agrees not to provide or otherwise make available the Licensed Software and any Derivative Work to any person other than Licensee's employees. A separate license - to be obtained through ITC - is required for use of the Licensed- and any Derivative Software by other Organisations.
Licensee agrees to take appropriate action with its employees to satisfy Licensee's obligations under this Agreement with respect to copying, protection and security of the Licensed Software.

4. TITLE
Title to and ownership of the Licensed- and Derivative Software and copies thereof shall at all times remain with ITC.

5. OBLIGATIONS
Licensee agrees that with the delivery of the Licensed Software ITC enters no obligations whatsoever.

6. LIABILITY
Licensee agrees that ITC shall not be liable for any loss, claim or damage arising out of Licensee's possession or use of the Licensed- and Derivative Software.

7. TERM
This Agreement is effective as per breaking the seal and shall remain in force for an indefinite period of time.

8. TERMINATION
Licensee agrees that with the termination of this Agreement the Licensed- any Derivative Software and all copies thereof shall be destroyed..

9. GOVERNING LAW
This Agreement is governed by Dutch law.

Appendix H

Customers Services Report Form

Please send a copy of the form to:

ITC
ILWIS Help desk
P.O. Box 6
7500 AA Enschede
The Netherlands
Fax: +31 53 4874484
E-mail: support.ilwis@itc.nl

Key:..
License:..
Version:..

Name..
Organization:...
Address:...
Country:...
Telephone:...
Fax:..
E-mail:...

Define your problems :
Problem type:

- ☐ Computer reboots
- ☐ Computer crashes
- ☐ Program quits at unexpected moment
- ☐ Error message, namely :..................................
- ☐ Incorrect results

Problem frequency:

- ☐ Regularly
- ☐ Irregularly
- ☐ Data related
- ☐ In connection with:..

Hardware used and platform type:..
....
....
....
....
....
....

Operations in use:..
....
....
....
....
....
....
....
....

Specify procedure in detail:..
....
....
....
....
....
....
....
....

Expected results:..
....
....
....
....
....
....
....

What happened, please specify in detail:...
....
....
....
....
....
....
....
....

Index

A

Additonal script commands, 333
Adobe Acrobat reader 3.0, 24
`AGGAVG`, 97
`AGGCNT`, 97
`AGGMAX`, 97
`AGGMED`, 97
`AGGMIN`, 97
`AGGPRD`, 97
Aggregate values, 94
`AGGSTD`, 97
`AGGSUM`, 97
AND operator, 93
Annotation
  Bitmap, 74, 225, 269
  Box, 73, 225, 269
  Graticule, 74, 225, 269
  Grid lines, 74, 225, 269
  Legend, 73, 225, 269
  Metafile, 74, 225, 269
  North arrow, 73, 225, 269
  Scale bar, 73, 225, 269
  Simple text. *See* Annotation: Single text
  Single text, 71, 225, 269
Annotation Text, 70, 72, 225, 293
Annotation Text editor, 88
`ansi.sys`, 20
Application Guide data, 20
Arc/Info
  Export to E00, 256
  Export to LIN, 256
  Export to non-compressed ASCII, 256
  Export to PTS, 256
  Import from E00, 265
  Import from LIN, 265
  Import from non-compressed ASCII, 265
  Import from PTS, 265
Arc/View
  Export to Shape, 256
  Import from Shape, 265
ASCII comma delimited
  Export to, 256
Aspect, 67
Atlas
  Export to BNA, 256
  Import from BNA, 265
Attribute table, 288
AutoCad DXF
  Export to .DXF, 256
  Import from .DXF, 265
AVG3X3, 120, 128, 130

B

Baud Rate, 25
Beginner's Guide Data, 20
Binary filters, 125, 126
BINMAJOR, 125
Bitmap
  Export to .BMP, 256
  Import from .BMP, 265

C

CalComp 2000, 28
CalComp Drawing Board 2, 28
CalComp Drawing Board 3, 28
Cart/o/graphics
  Export to .CV or .VTE, 256
Change domain of table, 211, 309
Classification, 249
Color composite, 136, 228, 242, 299
  24 bit, 137
  Dynamic, 138
  interactive. *See* Show Map List as Color Composite
  permanent, 136, 243
  Standard, 138
Column, 288
`ColumnAggregateAvg`, 95
`ColumnAggregateCnt`, 96
`ColumnAggregateMax`, 96
`ColumnAggregateMed`, 96
`ColumnAggregateMin`, 96
`ColumnAggregatePrd`, 96
`ColumnAggregateStd`, 96

ColumnAggregateSum, 96
ColumnJoin, 102
Communication settings, 25
config.sys, 20
Configuration of digitizers, 25
Configuration software, 25
Confusion matrix, 231
CONN8TO4, 125
Contour interpolation, 233
Coordinate system, 62, 290
Coordinate system boundary, 62
Coordinate system formula, 62, 68, 196, 199
Coordinate system latlon, 62, 196, 199
Coordinate system projection, 62, 199
Coordinate system tiepoints, 63, 199
Coordinate system Tiepoints editor, 84
Creating and running scripts, 104
Cross, 38, 108, 296
    With undefined values, 109
Customers Services Report Form, 18, 355

D

D2FDX2, 120, 128
D2FDXDY, 120, 128
D2FDY2, 120, 128
Data Bits, 25
dBase
    Export to .DBF, 256
    Export to .SDF, 256
    Import from .DBF, 265
DFDDN, 120, 128
DFDUP, 120, 128
DFDX, 120, 128
DFDY, 120, 128
Digital Elevation Model, 203, 233, 236, 266, 277
Digital Terrain Model. *See* Digital Elevation Model
Digitizers, 18, 25
    CalComp 2000, 28
    CalComp Drawing Board 2, 28
    CalComp Drawing Board 3, 28
    Numonics 2205, 29
    Numonics 2207, 29
    Numonics GraphicMaster, 29
    Numonics GraphicMaster II, 30
    SummaSketch, 30
DILATE4, 125
DILATE8, 125
Dip switches, 25
Directory structure after setup, 23
Display drivers, 19
Display the release notes, 23
Distance, 38
Distance calculation, 38
Domain, 253, 289
DXF
    Export to .DXF, 256
    Import from .DXF, 265

E

EDGESENH, 120, 128
Edit
    Annotation, 225
    Annotation Text, 70
    Coordinate system, 65
    Georeference, 59
    Map list, 52
    Point map, 40, 50
    Polygon map, 40, 43
    Raster map, 40
    Representation, 56, 254
    Representation class, 77
    Segment map, 40, 46
Editor
    Annotation Text, 88
    Coordinate system Tiepoints, 84
    Georeference Tiepoints, 80
    Representation class, 77, 248, 254
    Representation gradual, 249
    Representation value, 249, 254
E-mail addresses, 2, 18
Erdas
    Export to .DIG, 256
    Export to .GIS or .LAN, 256
    Import from .GIS, .LAN, 265
Export, 256
Export raster formats, 256
Export table formats, 256
Export vector formats, 256
Expressions for calculations and other operations, 332

F

Fiducial mark, 246, 266
File structure after setup, 23
Filter, 38, 119, 292, 298

`Filter expressions`, 119
Filter types, 120, 124, 125
Flag codes, 27
Format, 27
Function, 293

G

General raster
  Import from, 265
Georeference, 57, 290
  3D, 58, 244, 252
  Corners, 57, 228, 244, 252, 284
  Digital Elevation Model, 271
  Direct linear, 58, 81, 227, 233, 244, 252, 266, 271
  Orthophoto, 58, 81, 227, 233, 244, 246, 252, 266, 273, 277
  Tiepoints, 57, 80, 228, 244, 252, 266, 271
Georeference Tiepoints editor, 80, 229, 279
Geosoft
  Export to, 256
GeoTIFF
  Export to .TIF, 256
  Import from .TIF, 265
GIF
  Export to .GIF, 256
  Import from .GIF, 265
Gina
  Export to .GIA, 256
Glue raster maps, 111, 298
  Georeference, 112, 114, 118
Glue tables, 218, 310
Gradient or derivative, 120
Gridding. *See* point interpolation

H

Hardware and software requirements, 18
  Recommended, 18
Hardware key, 17
Hardware Key registration program, 23
Help files, 22
Histogram, 291
How to
  Add Annotation to a map window, 225
  Add coordinates to an image, 227
  Calculate a confusion matrix, 231
  Calculate a cross-section through the terrain, 236
  Calculate a Digital Elevation Model, 233
  Calculate the Optimum Index Factor, 235
  Change the domain of a map, 239
  Create a color composite, 242
  Create a georeference, 244
  Create a representation, 247
  Create a sample set for image classification, 249
  Create an orthophoto, 246
  Display a map list as a slide show, 250
  Edit a georeference, 252
  Edit a representation, 253
  Export data, 256
  Import ASCII point data, 257
  Import data, 264
  Monoplot, 266
  Save a map view, 270
  Save annotation, 269
  Screen digitize, 270
  Use parameters in scripts, 275
  Use the Georeference Tiepoints editor, 273
HPGL
  Export to HPGL, 256

I

IDA
  Export to IMG, 256
  Import from .IMG, 265
Idrisi
  Export to raster, 256
  Import from raster, 265
ILWIS 1.4
  Export to raster, 256
  Export to table, 256
  Export to vector, 256
  Import from raster, 265
  Import from table, 265
  Import from vector, 265
ILWIS 2.1 package, 17
ILWIS ASCII
  Export to 1.4 .ASC, 256
  Export to SMT, 256
  Import from .ASC, 265
  Import segments from SMT, 265
ILWIS expressions, 40, 311
ILWIS expressions (alphabetic), 327
ILWIS help file, 22
ILWIS icons
  Data objects, 287

Program folder icons, 22
ILWIS menu help file, 22
ILWIS operations, 333
ILWIS operations help file, 22
ILWIS script language, 331
Import raster formats, 265
Import table formats, 265
Import vector formats, 265
INBND4, 125
INBND8, 125
InfoCam
Export to .SEQ, 256
Import from .SEQ, 265
Installing
Add components, 21
Adobe Acrobat pdf reader, 24
Digitizers, 25
From CD-ROM, 19
Uninstall, 22
Intergraph
Export to .SIF, 256
ITC, 2

J

Join columns, 98

K

Kriging, 50, 158, 164, 181, 285, 304

L

LAPLACE, 120, 128
LIFEGAME, 125
Linear filters, 120
Logical operators, 93

M

Main window
Menu commands, 33
Maintenance, 18
MAJORITY, 125, 133
MAJUNDEF, 125, 133
MAJZERO, 125, 133
Manual digitizer Setup, 25
Map Calculation, 40
Map list, 51, 250, 288
Map view, 291
MapCalc and Tabcalc, 332
`MapColorComp`, 139
`MapColorComp24`, 139
`MapColorComp24HistEq`, 140
`MapColorComp24HSI`, 140
`MapColorComp24Linear`, 140
`MapColorCompHistEq`, 139
`MapColorCompLinear`, 139
MapCross, 110
`MapFilter`, 119, 129
`MapGlue`, 117
`MapHeckbert`, 139
`MapKrigingOrdinary`, 187
`MapKrigingSimple`, 187
`MapMovingAverage`, 169
`MapMovingSurface`, 178
`MapRasterizePolygon`, 202
`MapRasterizeSegment`, 204
`MapResample`, 147
`MapTrendSurface`, 173
Matrix, 292
`MED3x3`, 124, 131
`MED5x5`, 124, 131
Monoplot, 271
Moving average, 50, 163, 167, 170, 181, 190, 233, 284, 303
Moving surface, 50, 164, 175, 233, 304

N

Nearest Neighbour. *See* Nearest point
Nearest point, 163, 165, 303
NOT operator, 94
Numonics 2205, 29
Numonics 2207, 29
Numonics GraphicMaster, 29
Numonics GraphicMaster II, 30

O

OIF, 150
Optimum Index Factor, 138, 150, 235
OR operator, 93

Ordinary Kriging, 164, 182, 191
OUTBND4, 125
OUTBND8, 125

P

Parity, 25
Pattern filters, 126
PCI-ILWIS Nederland B.V, 2
PCX
  Export to .PCX, 256
  Import from .PCX, 265
PDF files, 24
PEPPSALT, 125
Plotters, 18
Point interpolation, 50, 163, 284, 303
  Kriging, 50, 164, 304
  Moving average, 50, 163, 303
  Moving surface, 50, 164, 304
  Nearest point, 163, 303
  Trend surface, 50, 164, 304
Point map, 47, 288
Point statistics, 152
`PointMapFromTable`, 216
`PointMapPolLabels`, 210
`PointMapTransform`, 198
Polygon map, 41, 287
Polygonize, 41
`PolygonMapTransform`, 198
Polygons to points, 208
Polygons to raster, 200
Printer and plotter setup, 31
Printers, 18
Program files, 20
Program folder icons, 22
Projection, 65
Projection characteristics
  Conformality, 68
  Equidistance, 68
  Equivalence, 68
Projection types, 66
  Azimuthal projections, 66
  Conical projections, 67
  Cylindrical projections, 66

R

Rank order filters, 124
Raster map, 38, 287
Raster to polygons, 205
Rasterize, 39
  Polygons to raster, 200
  Segments to raster, 202
Recommended hardware and software requirements, 18
Register ILWIS Key program, 21
Registration code, 21
Release notes, 23
Representation, 53, 253, 289
Representation class, 53, 248
Representation class editor, 77, 248, 254
Representation gradual, 54, 239, 248
Representation gradual editor, 249
Representation value, 54, 239, 248
Representation value editor, 249, 254
Required hard disk space, 19
  Applications Guide data, 19
  User's Guide data, 19
Resample, 145, 232, 247, 300
Resampling methods, 145
  Bicubic, 145, 149, 164, 168, 176, 181
  Bilinear, 145, 149, 164, 168
  Nearest neighbour, 145, 148
Resolution, 19, 25, 27

S

Sample set, 250, 291
Script, 275, 293
Segment map, 44, 287
`SegmentMapTransform`, 198
Segments to raster, 202
Semi-variogram, 157, 158, 181, 190, 283
Serial port, 25
Services, 17
Set up
  Digitizer manually, 25
  Digitizer using a WinTab driver, 25
  Printer and plotter, 31
Setup program
  Select Components, 20
  Select Directories, 20
  Select Program Folder, 20
  Setup Complete, 20
SHADOW, 120, 128
Show Map List as Color Composite, 107, 136, 228, 243, 295
SHRINK4, 125

SHRINK8, 125
Simple Kriging, 164, 182, 190
Single text, 72
Software license agreement, 17
Software Maintenance Contract, 18
Spatial correlation, 152, 153, 173, 181, 190, 283, 302
Spatial correlation methods
  Bidirectional, 155, 156
  Omnidirectional, 155, 156
Standard deviation filters, 126
Start ILWIS 2.1, 22
Stop Bits, 25
Structure after setup
  Directory, 23
  Documentation in .PDF files, 24
  Files, 23
System representations, 57

T

Table, 288
Table calculation, 94
Table to point map, 214, 310
`TableChangeDomain`, 213
`TableCross`, 110
TableGlue, 222
`TableSpatCorr`, 161
Technical support, 18
Themak2
  Export to .THR, 256
  Export to polygons .THP, 256
  Export to segments .THS, 256
Thiessen. *See* Nearest point
TIFF
  Export to .TIF, 256
  Import from .TIF, 265
Transform coordinates, 198
Transform map operation, 195
Transformation
  Affine, 278
  Bilinear, 278
  Projective, 279
Trend surface, 50, 164, 172, 304
Two-dimensional table, 292

U

Uninstall program, 23
Uninstalling the ILWIS 2.1 software, 22
UseMap
  Export to, 256
User's Guide data, 20
User-defined filters, 127
  Average filter, 130
  Linear filter, 128
  Majority filter, 133
  Median filter, 131
  Pattern filter, 135
  Rank order filter, 131
  Standard deviation filter, 135

V

Vectorize
  Polygons to points, 208
  Raster to polygons, 205
Viewing documentation .PDF files, 24

W

Warranty, 17
Web addresses, 2, 18
Windows bitmap
  Export to .BMP, 256
  Import from .BMP, 265
WinTab driver, 25

X

XOR operator, 94