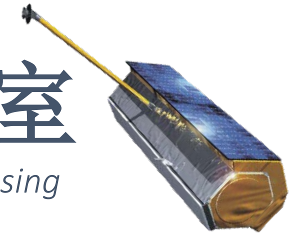
测绘遥感信息工程国家重点实验室

State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing



Synthetic Aperture Radar Tomography – practical course

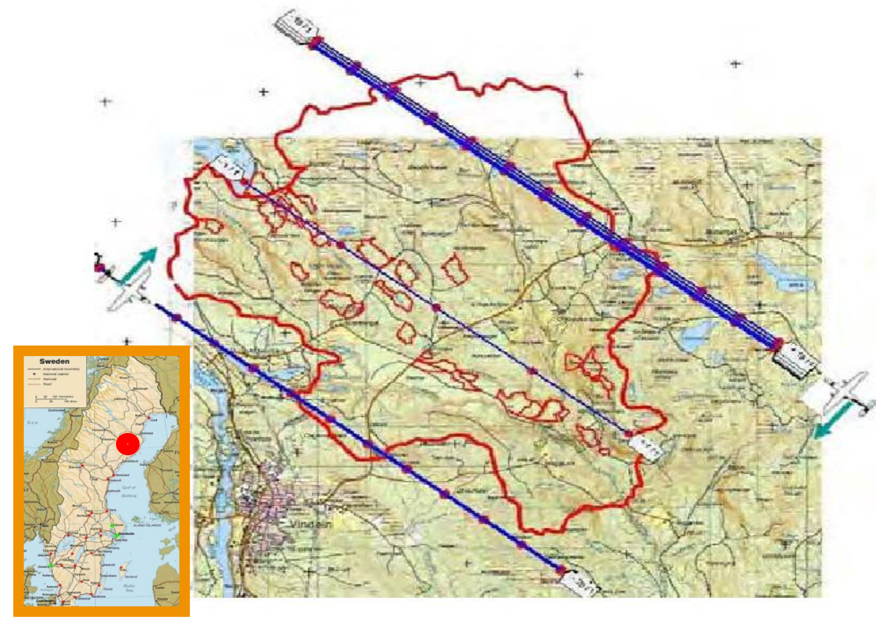
Timo Balz, Stefano Tebaldini, Laurent Ferro-Famil

TomoSAR_Main.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
% DEMONSTRATIVE TOMOGRAPHIC SAR PROCESSING FOR FOREST ANALYSIS
% AUTHOR: STEFANO TEBALDINI, POLITECNICO DI MILANO
% EMAIL: stefano.tebaldini@polimi.it
% TEL: +390223993614
%
% THE FOLLOWING SCRIPT AND ALL RELATED SCRIPTS/FUNCTIONS AND DATA ARE INTENDED AS
% MATERIAL FOR THE TOMOSAR TRAINING COURSE HELD IN BEIJING IN FEBRUARY 2015
% BY LAURENT FERRO-FAMIL AND STEFANO TEBALDINI
%
% THIS SOFTWARE WAS DEVELOPED AND TESTED USING MATLAB R2011b
%
% SAR DATA USED IN THIS SCRIPT ARE PART OF THE SAR DATA-SET ACQUIRED BY DLR
% IN 2008 IN THE FRAME OF THE ESA CAMPAIGN BIOSAR 2008
% DATA FOCUSING, COREGISTRATION, PHASE FLATTENING, AND GENERATION OF KZ
% MAPS WERE CARRIED OUT BY DLR.
% DATA PHASE CALIBRATION WAS CARRIED OUT BY THE AUTHOR
%
% TERRAIN ELEVATION DATA USED IN THIS SCRIPT ARE EXTRACTED FROM
% THE LIDAR DATA-SET ACQUIRED BY THE SWEDISH DEFENCE RESEARCH AGENCY (FOI)
% AND HILDUR AND SVEN WINQUIST'S FOUNDATION IN THE FRAME OF THE ESA
% CAMPAIGN BIOSAR 2008
% PROCESSING OF LIDAR DATA AND PROJECTION ONTO SAR GEOMETRY WAS CARRIED OUT
% BY THE AUTHOR
%
% YOU ARE WELCOME TO ADDRESS ME QUESTIONS/COMMENTS/CORRECTIONS AT
% stefano.tebaldini@polimi.it
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
```

Data

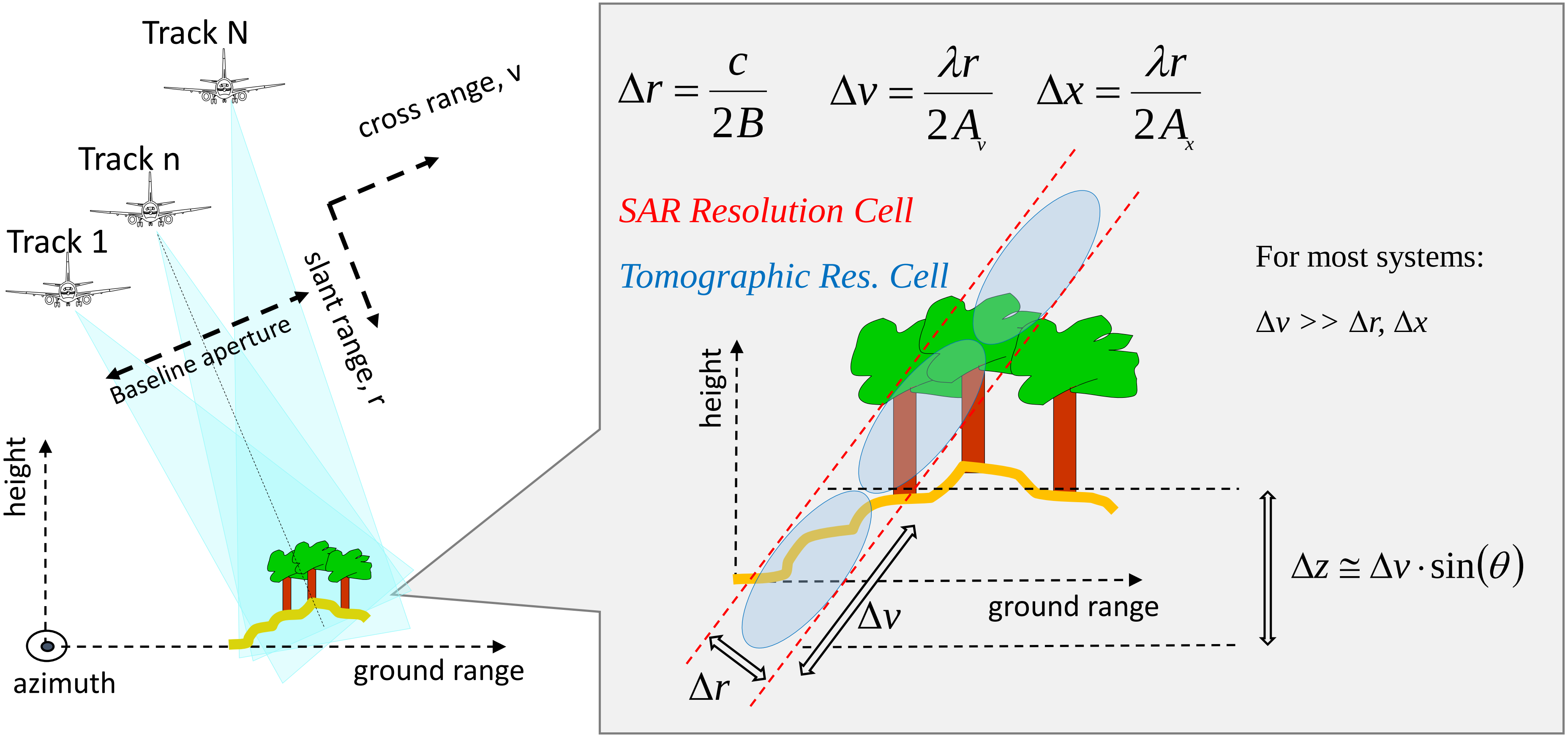
Campaign	BioSAR 2008 - ESA
System	E-SAR - DLR
Site	Krycklan river catchment, Northern Sweden
Scene	Boreal forest Pine, Spruce, Birch, Mixed stand
Topography	Hilly
Tomographic Tracks	6 + 6 – Fully Polarimetric (South-West and North-East)
Carrier Frequency	P-Band and L-Band
Slant range resolution	1.5 m
Azimuth resolution	1.6 m
Vertical resolution (P-Band)	20 m (near range) to >80 m (far range)
Vertical resolution (L-Band)	6 m (near range) to 25 m (far range)



Forward model

Resolution is determined by pulse bandwidth along the slant range direction, and by the lengths of the synthetic apertures in the azimuth and cross range directions

⇒The SAR resolution cell is split into multiple layers, according to baseline aperture



$$\Delta r = \frac{c}{2B} \qquad \Delta v = \frac{\lambda r}{2A_v} \qquad \Delta x = \frac{\lambda r}{2A_x}$$

$$\Delta z \cong \Delta v \cdot \sin(\theta)$$

Vertical wavenumber

Each focused SLC SAR image is obtained as the Fourier Transform of the scene complex reflectivity along the cross-range coordinate

$$y_n(r,x) = \int s(r,x,v)\exp\left(-j\frac{4\pi}{\lambda r}b_n v\right)dv$$

$y_n(r,x)$: SLC pixel in the *n-th* image

$s(r,x,v)$: average complex reflectivity of the scene within the SAR 2D resolution cell at (r,x)

b_n : normal baseline for the *n-th* image

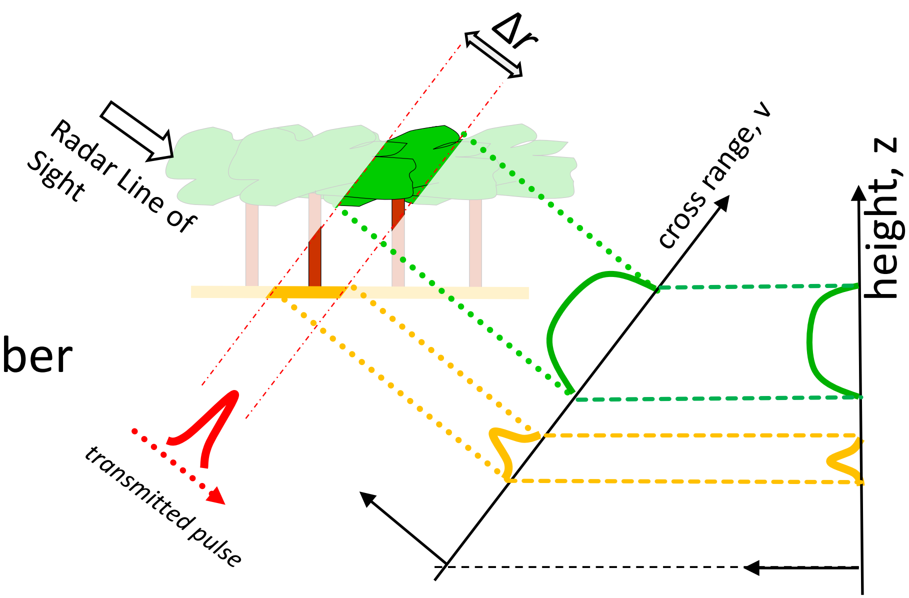
λ : carrier wavelength

Change of variable from cross range to height

$$z = v \cdot \sin\theta$$

$$\Downarrow$$

$$y_n(r,x) = \int s(r,x,z)\exp(-jk_z(n)\cdot z)dz$$



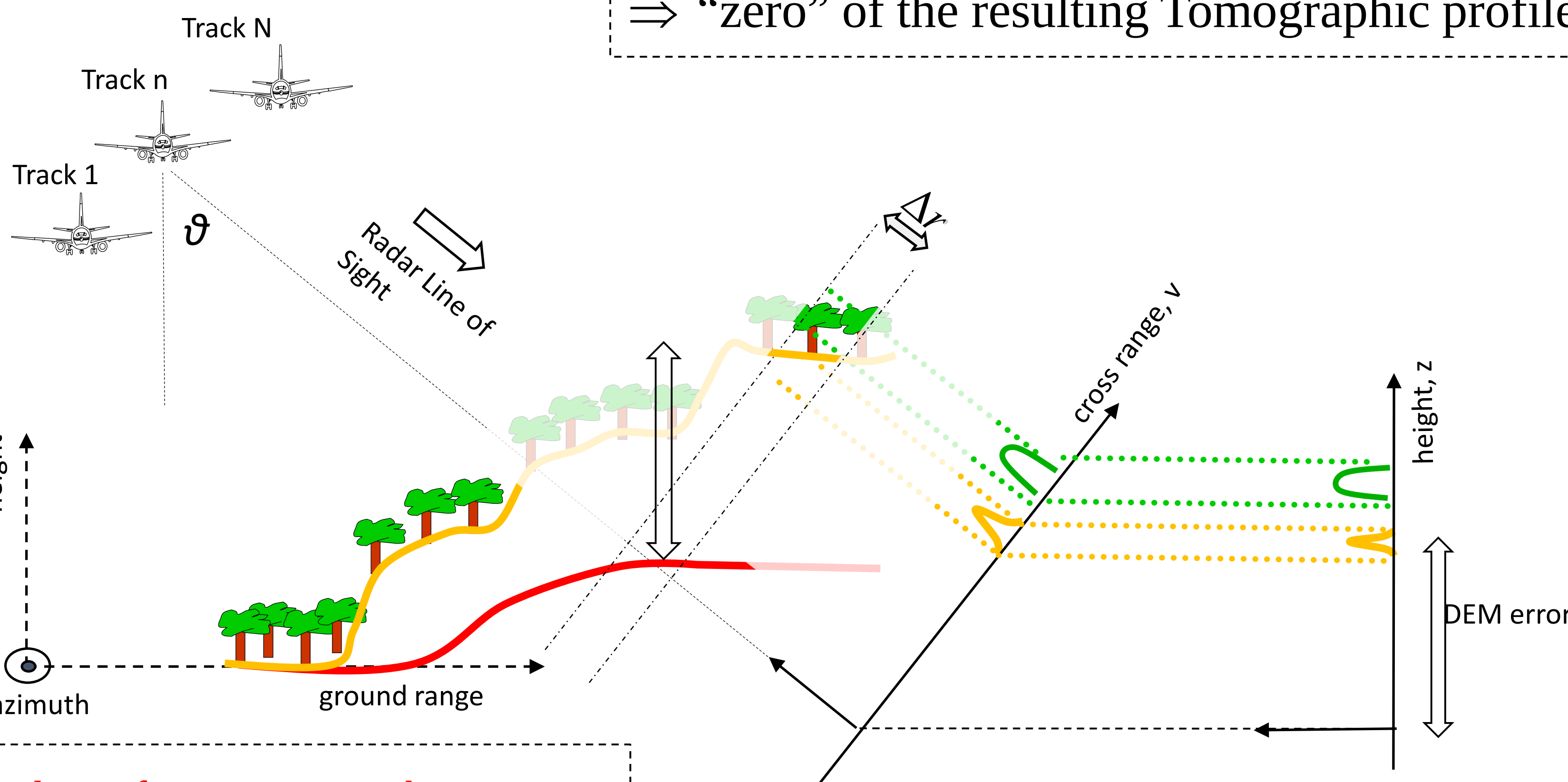
k_z is usually referred to as vertical wavenumber or phase to height conversion factor

$$k_z(n) = \frac{4\pi}{\lambda r}\frac{b_n}{\sin\theta}$$

Reference height

$$y_n(r,x) = \int s(r,x,z)\exp(-jk_z(n)\cdot z)\cdot dz$$

Note: z is always intended as height with respect to a Digital Elevation Model (DEM)
⇒ "zero" of the resulting Tomographic profiles



Red = Reference terrain elevation

Orange = True terrain elevation

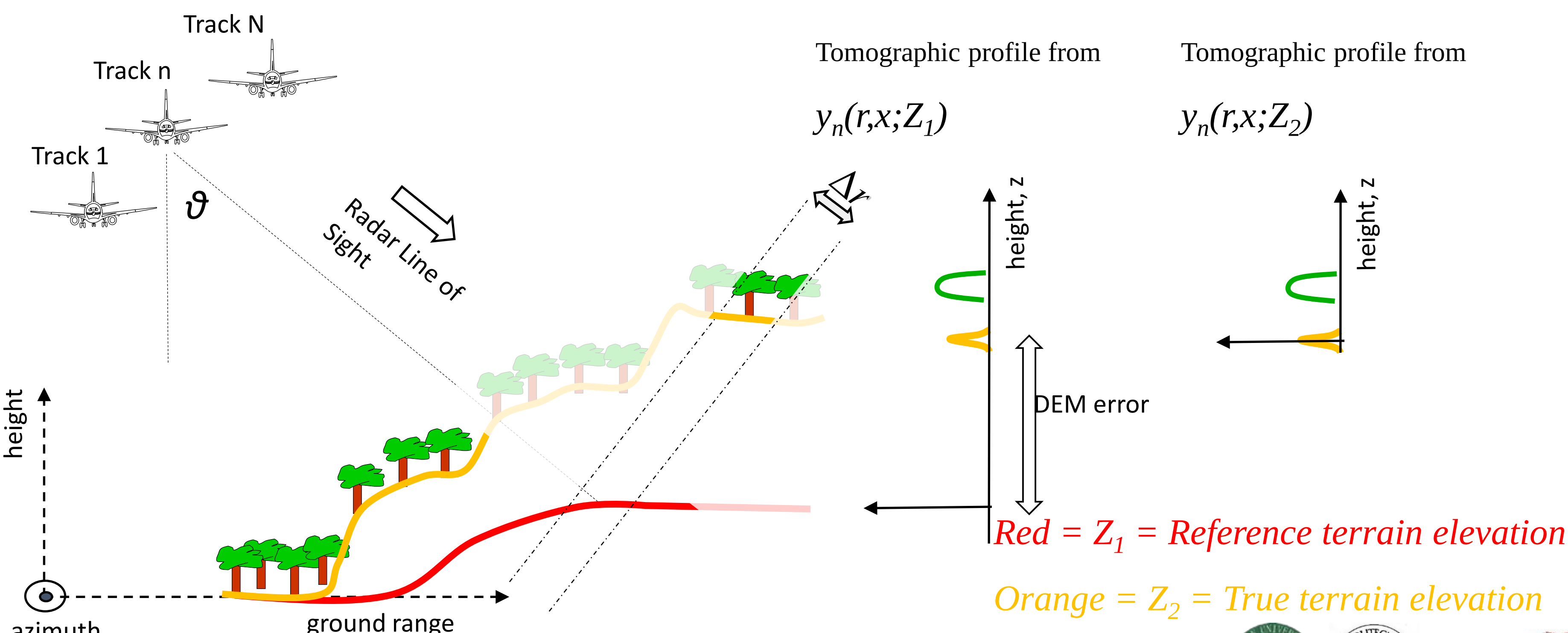
DEM subtraction

The dependence on height is limited to the phase terms $k_z z$

$$y_n(r,x) = \int s(r,x,z)\exp(-jk_z(n)\cdot z)\cdot dz$$

$\Rightarrow$ Passing from one reference DEM to another $\Leftrightarrow$ phase steering from Z_1 to Z_2

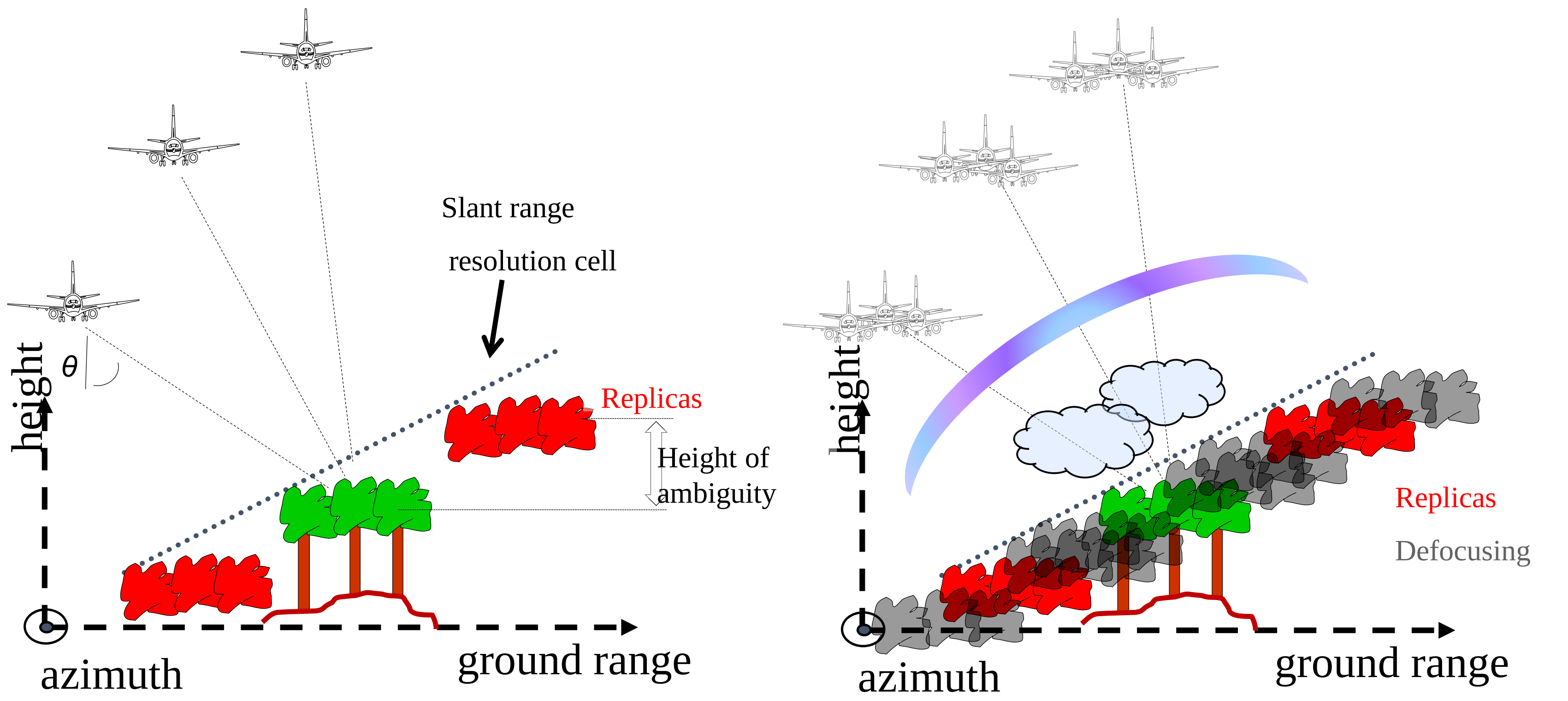
$$y_n(r,x;Z_2) = y_n(r,x;Z_1)\exp^{-jk_z(n)\cdot(Z_1-Z_2)}$$



Phase calibration

Phase jitters in different passes result in signal defocusing

- Spaceborne: tropospheric and ionospheric phase screens
- Airborne: uncompensated platform motions *on the order of a fraction of a wavelength*

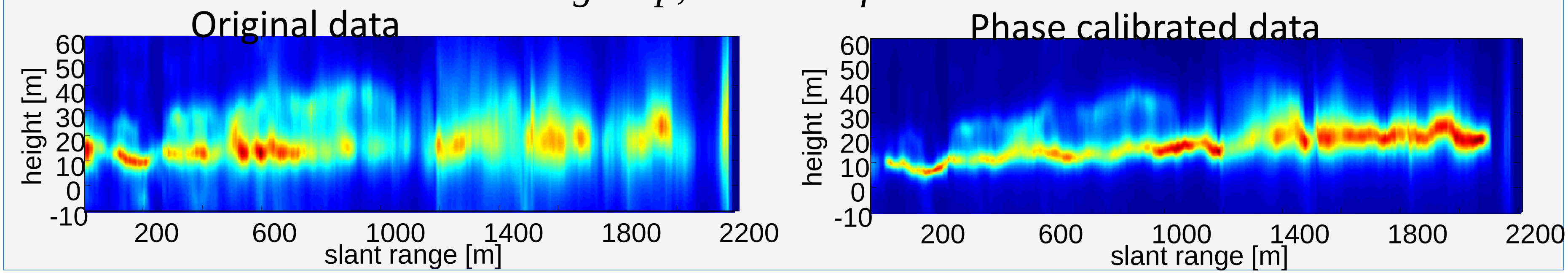


Phase calibration

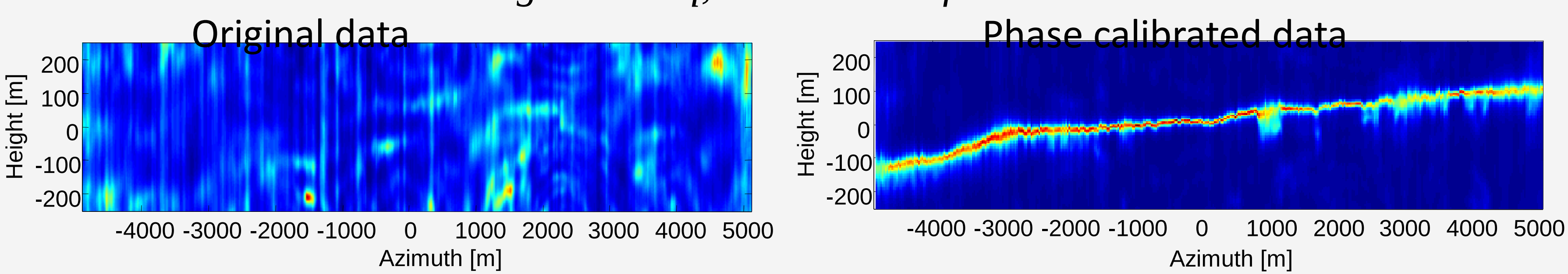
Current navigational systems employed by airborne SARs do not provide, in general, sub-wavelength accuracy concerning the location of one flight line with respect to another

⇨ Need for a data-driven Phase Calibration procedure

TomoSAR at Remningstorp, Sweden – from BioSAR 2007



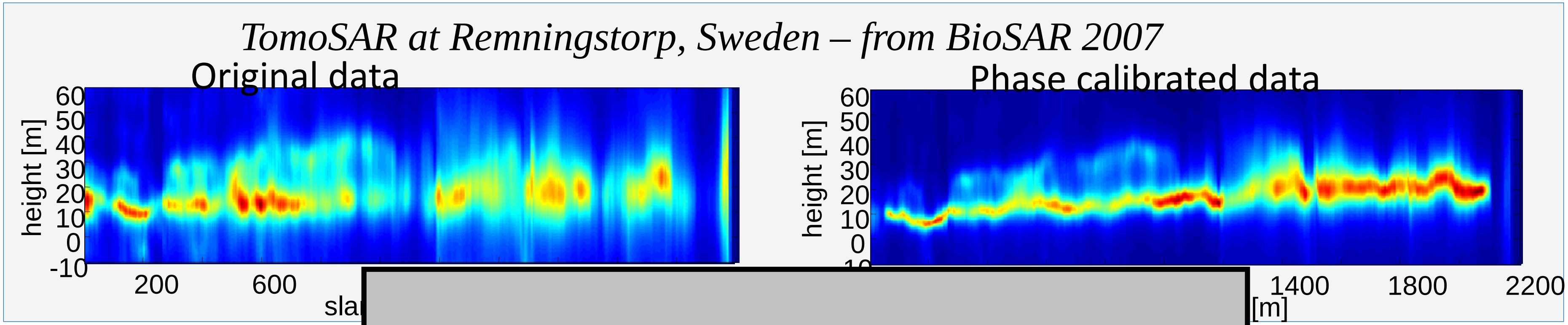
TomoSAR at Kangerlussuaq, Greenland – from IceSAR 2012



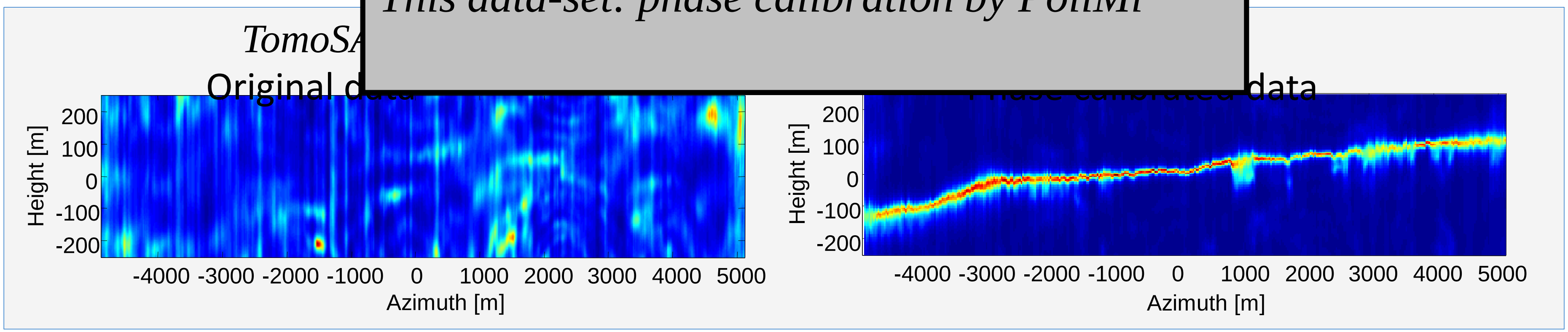
Phase calibration

Current navigational systems employed by airborne SARs do not provide, in general, sub-wavelength accuracy concerning the location of one flight line with respect to another

➡ Need for a data-driven Phase Calibration procedure



This data-set: phase calibration by PoliMi



TomoSAR_Main.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% LOAD DATA
if not(exist('I'))
    load('BioSAR_2_L_Band_sample_data')
    Master = 1
    [Nr,Na,N] = size(I{1})
    N_pol = length(I)
    rem_dem_flag = 1
    if rem_dem_flag % remove dem phases (optional)
        for pol = 1:N_pol
            for n = 1:N
                dem_phase = kz(:,:,n).*(DEM - DEM_avg);
                I{pol}(:,:,n) = I{pol}(:,:,n).*exp(1i*dem_phase);
            end
        end
    end
    Ch = {'HH','HV','VV'}
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
```

Notes:

Data can be referenced to a flat DEM (`DEM_avg`) or to the Lidar DEM (`DEM`)

TomoSAR_Main.m

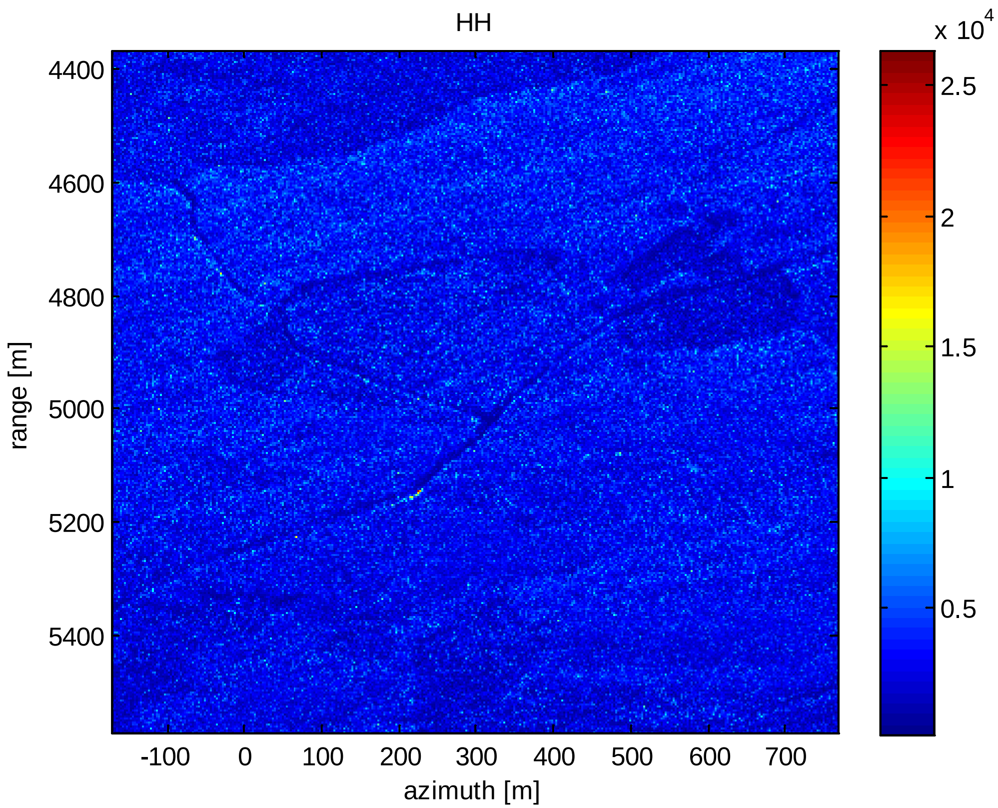
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% Let's look at the data first....
for pol = 1:N_pol
    figure, imagesc(az_ax,rg_ax,sum(abs(I{pol}),3)), colorbar
    title(Ch{pol})
    xlabel('azimuth [m]')
    ylabel('range [m]')
end
 figure, imagesc(az_ax,rg_ax,DEM), colorbar
 title('DEM [m]')
 xlabel('azimuth [m]')
 ylabel('range [m]')
 figure, imagesc(az_ax,rg_ax,FOR_H,[0 35]), colorbar
 title('Forest height [m]')
 xlabel('azimuth [m]')
 ylabel('range [m]')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
```

Notes:

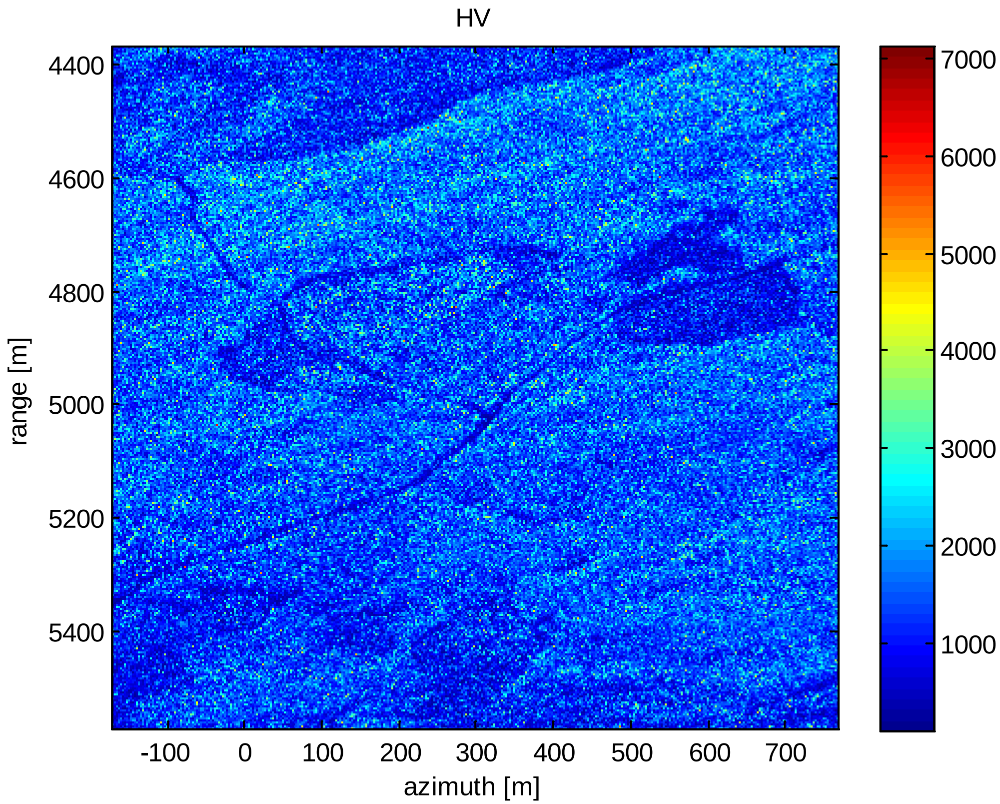
DEM = Lidar DEM

FOR_H= Lidar forest height

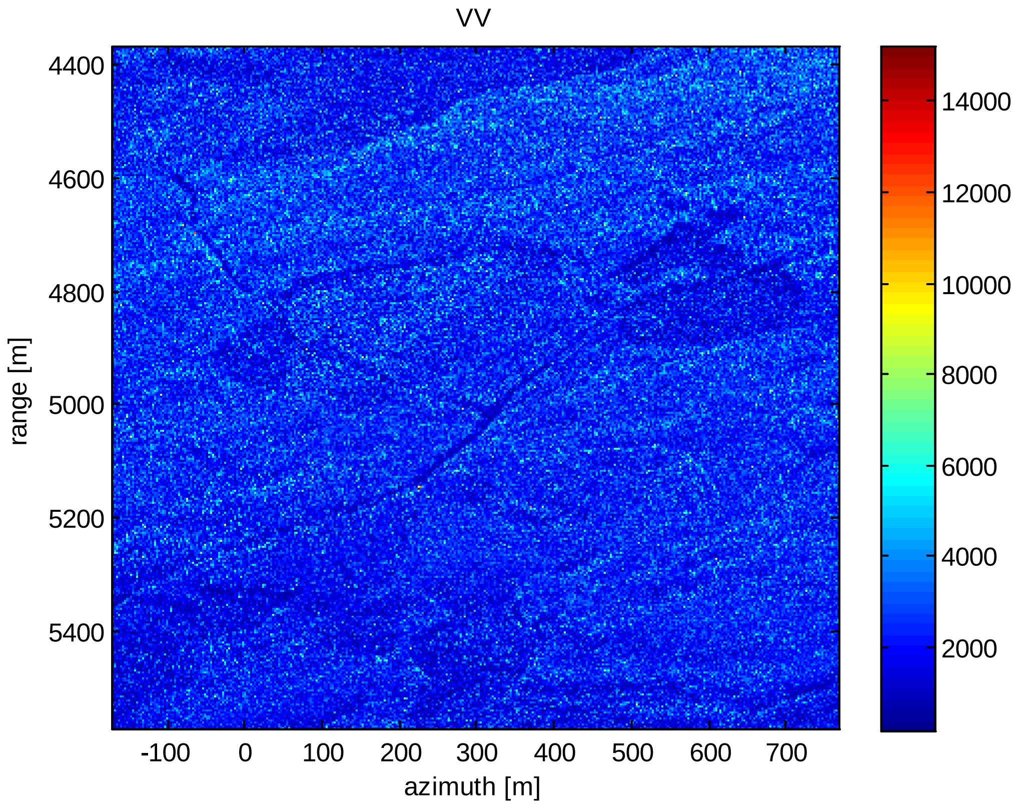
Results


HH
x 10^4
2.5
2
1.5
1
0.5
4400
4600
4800
5000
5200
5400
range [m]
-100
0
100
200
300
400
500
600
700
azimuth [m]

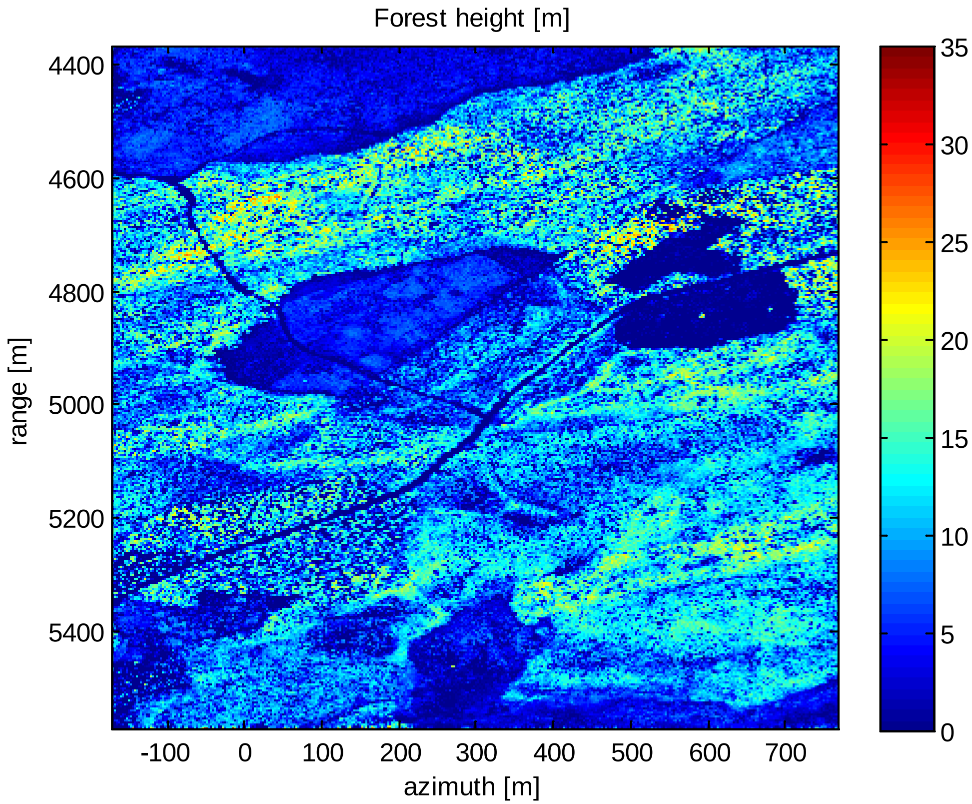
Results


HV
4400
4600
4800
5000
5200
5400
range [m]
-100
0
100
200
300
400
500
600
700
azimuth [m]
7000
6000
5000
4000
3000
2000
1000

Results

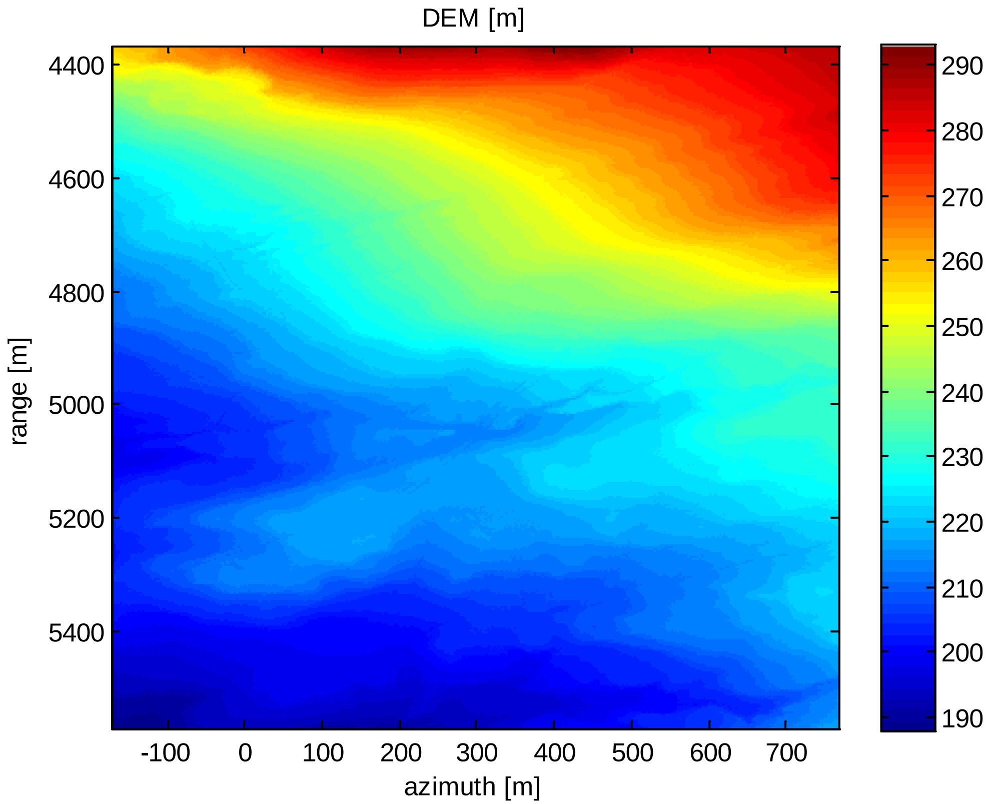

VV
4400
4600
4800
5000
5200
5400
range [m]
-100
0
100
200
300
400
500
600
700
azimuth [m]
14000
12000
10000
8000
6000
4000
2000

Results



Forest height [m]
4400
4600
4800
5000
5200
5400
range [m]
-100
0
100
200
300
400
500
600
700
azimuth [m]
35
30
25
20
15
10
5
0

Results



DEM [m]
4400
4600
4800
5000
5200
5400
range [m]
-100
0
100
200
300
400
500
600
700
azimuth [m]
290
280
270
260
250
240
230
220
210
200
190

TomoSAR_Main.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% COHERENCE EVALUATION
% estimation window (in meters)
Wa_m = 30
Wr_m = 30
[COV_4D,a_sub,r_sub] = Generate_covariance_matrix(I{1},az_ax,rg_ax,Wa_m,Wr_m);

figure, InSAR_view(abs(COV_4D),[0 1]), colorbar
title('InSAR coherences')
figure, InSAR_view(angle(COV_4D),[-pi pi]), colorbar
title('InSAR phases')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
```

Notes:

`COV_4D` *is a 4D data structure representing the complex coherence as a function of each interferometric pair, i.e.:* $y_{nm}(r,x)$

`Generate_covariance_matrix.m` = *function to evaluate COV_4D from SLC images*

`InSAR_view` = *function to view COV_4D as a big 2D matrix*

Generate_covariance_matrix.m

```
function [Cov,x_sub,y_sub] = Generate_covariance_matrix(F,x_ax,y_ax,Wx_m,Wy_m)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[Ny,Nx,N] = size(F);
% pixel sampling
dx = x_ax(2)-x_ax(1);
dy = y_ax(2)-y_ax(1);
% filter along x
Lx = round(Wx_m/2/dx);
filter_x = hamming(2*Lx+1);
% sub-sampling along x
x_sub = Lx+1:max(round(Lx/2),1):Nx-Lx;
% filter along y
Ly = round(Wy_m/2/dy);
filter_y = hamming(2*Ly+1);
% sub-sampling along y
y_sub = Ly+1:max(round(Ly/2),1):Ny-Ly;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Generate_covariance_matrix.m

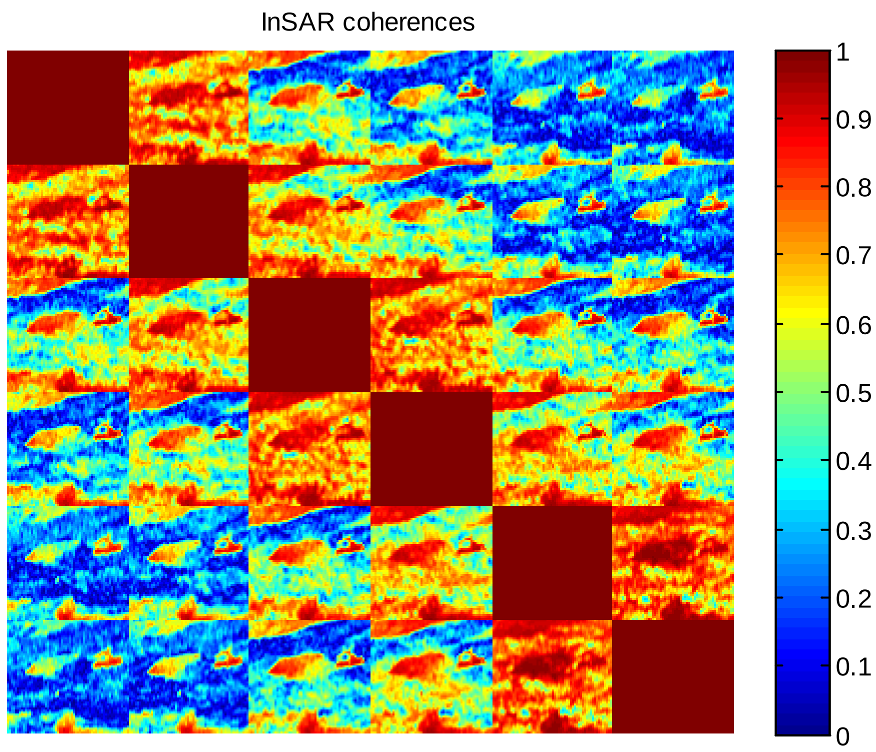
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Covariance matrix evaluation
Nx_sub = length(x_sub);
Ny_sub = length(y_sub);
Cov = ones(Ny_sub,Nx_sub,N,N);
for n = 1:N
    In = F(:,:,n); % n-th image
    % second-order moment
    Cnn = filter_and_sub_sample(In.*conj(In),filter_x,filter_y,x_sub,y_sub);
    for m = n:N
        Im = F(:,:,m);
        Cmm = filter_and_sub_sample(Im.*conj(Im),filter_x,filter_y,x_sub,y_sub);
        Cnm = filter_and_sub_sample(Im.*conj(In),filter_x,filter_y,x_sub,y_sub);
        % coherence
        coe = Cnm./sqrt(Cnn.*Cmm);
        Cov(:,:,n,m) = coe;
        Cov(:,:,m,n) = conj(coe);
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Cnm = filter_and_sub_sample(Cnm,filter_x,filter_y,x_sub,y_sub)
% filter and sub-sample
t = Cnm;
t = conv2(t,filter_x(:)','same');
t = t(:,x_sub);
t = conv2(t,filter_y(:),'same');
t = t(y_sub,:);
Cnm = t;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

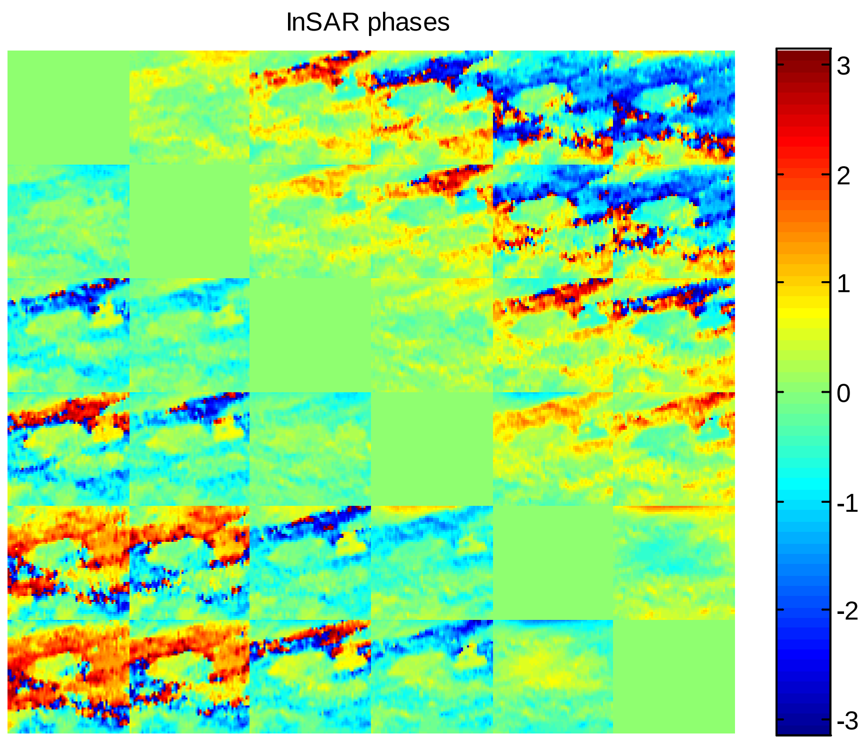
UNIVERSITÉ DE RENNES 1

InSar_View.m

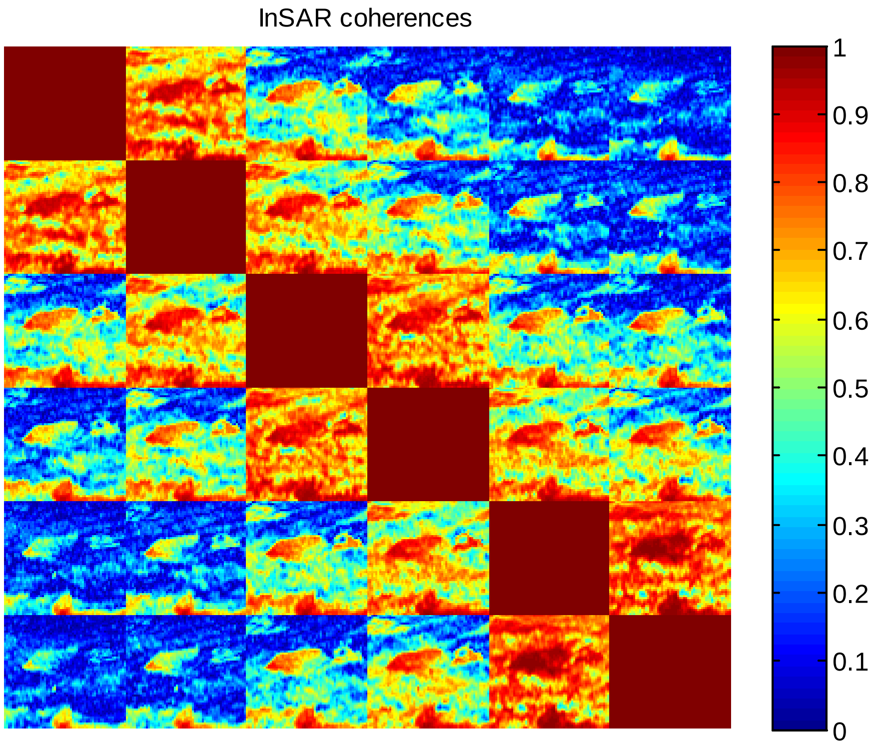
```
function InSAR_view(DX,cax)

[Nx_out,Ny_out,N,a] = size(DX);
if a == N
    flag_4D = 1;
else
    flag_4D = 0;
end
DDX = zeros(N*Nx_out,N*Ny_out);
for n = 1:N
    ind_n = [1:Nx_out] + Nx_out*(n-1);
    for m = 1:N
        ind_m = [1:Ny_out] + Ny_out*(m-1);
        if flag_4D
            DDX(ind_n,ind_m) = DX(:,:,n,m);
        else
            DDX(ind_n,ind_m) = DX(:,:,m) - DX(:,:,n);
        end
    end
end
if exist('cax')==1
    if max(abs(cax-[-pi pi]))==0
    disp('phase')
    DDX = angle(exp(1i*DDX));
end
    imagesc(DDX,cax)
else
    imagesc(DDX)
end
axis off
```

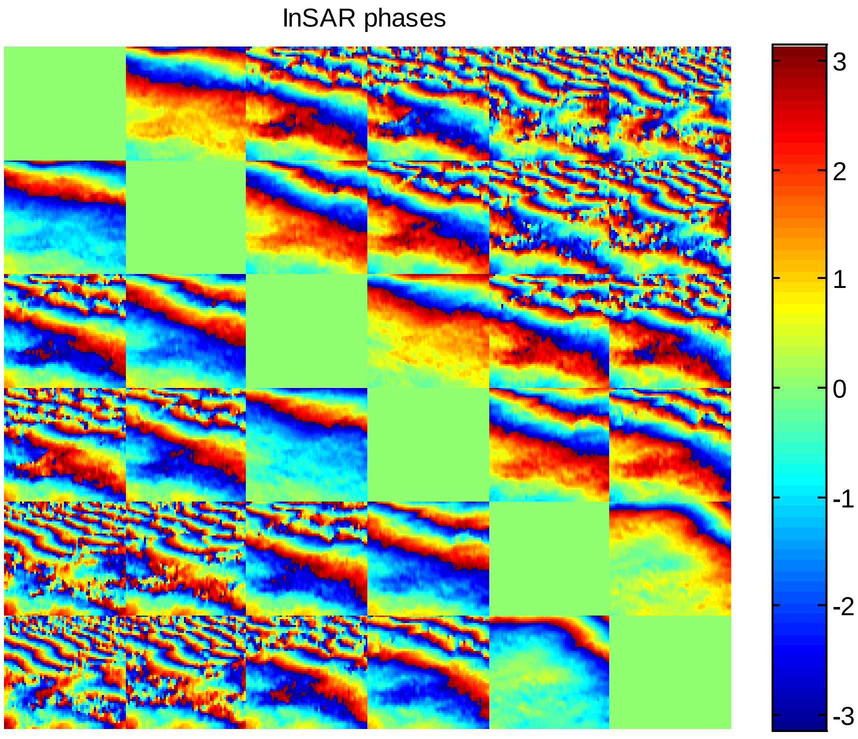
Results – InSAR coherence – DEM subtracted





Results – InSAR coherence – DEM not subtracted





Notes:

Noticeable topographic phases

Lower coherence magnitudes

TomoSAR_Main.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% TOMOGRAPHIC PROCESSING (3D focusing)
% vertical axis (in meters)
if rem_dem_flag % height w.r.t. DEM
    dz = 0.5;
    z_ax = [-20:dz:40];
else % % height w.r.t. average DEM
    dz = 1;
    z_ax = [-150:dz:150];
end
Nz = length(z_ax);
% half the number of azimuth looks to be processed
Lx = 10
% azimuth position to be processed (meters)
az_profile_m = 590;
az_profile_m = 678
az_profile_m = -92
% Focus in SAR geometry
TomoSAR_focusing
if rem_dem_flag == 0
    % the following routines have been written assuming DEM phases are removed
    return
end
% Geocode to ground geometry and compare to Lidar forest height
Geocode_TomoSAR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
```

TomoSAR_Focusing.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% pixel index
[t,a0] = min(abs(az_ax-az_profile_m));
az_ind = a0 + [-Lx:Lx];
% Focusing
for pol = 1:N_pol
    Tomo_3D{pol} = zeros(Nz,Nr,length(az_ind));
    for z = 1:Nz
        t = I{pol}(:,az_ind,:).*exp(1i*kz(:,az_ind,:).*z_ax(z));
        Tomo_3D{pol}(z,:,:) = mean(t,3);
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Notes:

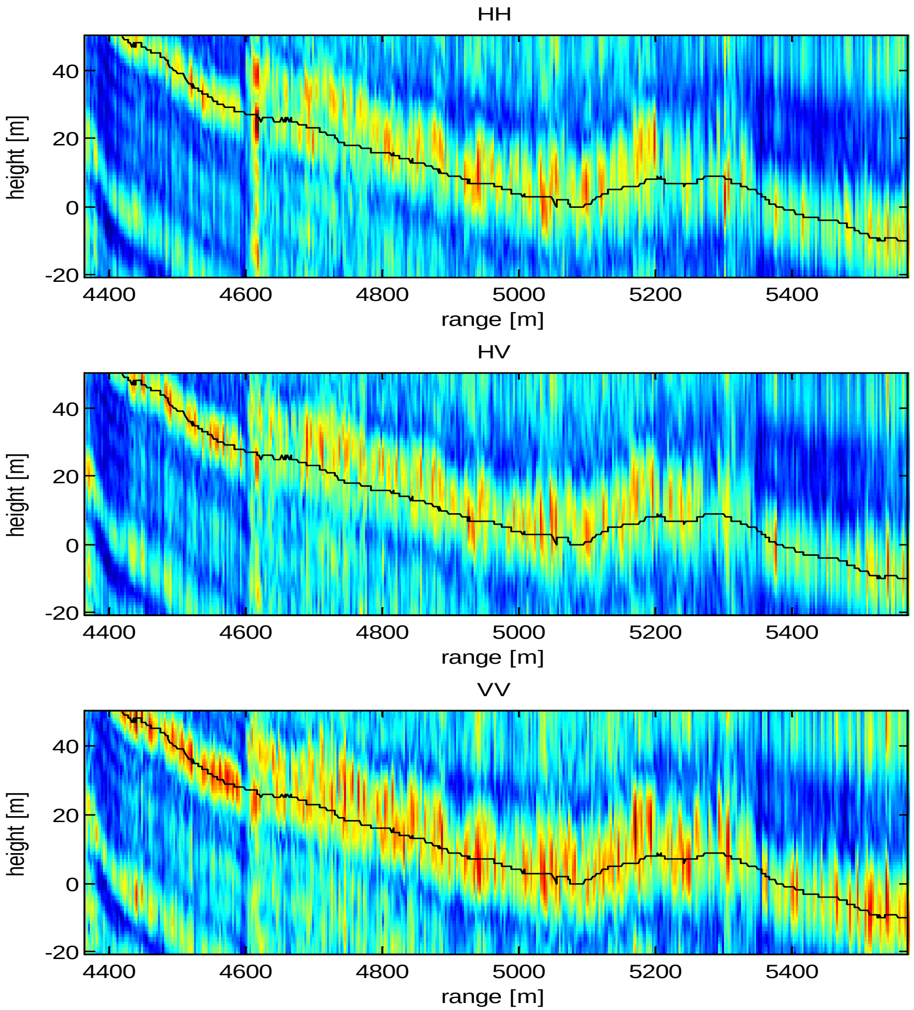
Just a discrete Fourier Transform

$$y_n(r,x) = \int s(r,x,z)\exp(-jk_z(n)\cdot z)\cdot dz \quad \Longrightarrow \quad \hat{s}(r,x,z) = \sum_n y_n(r,x)\exp(jk_z(n)\cdot z)$$

Results – Tomographic Profiles

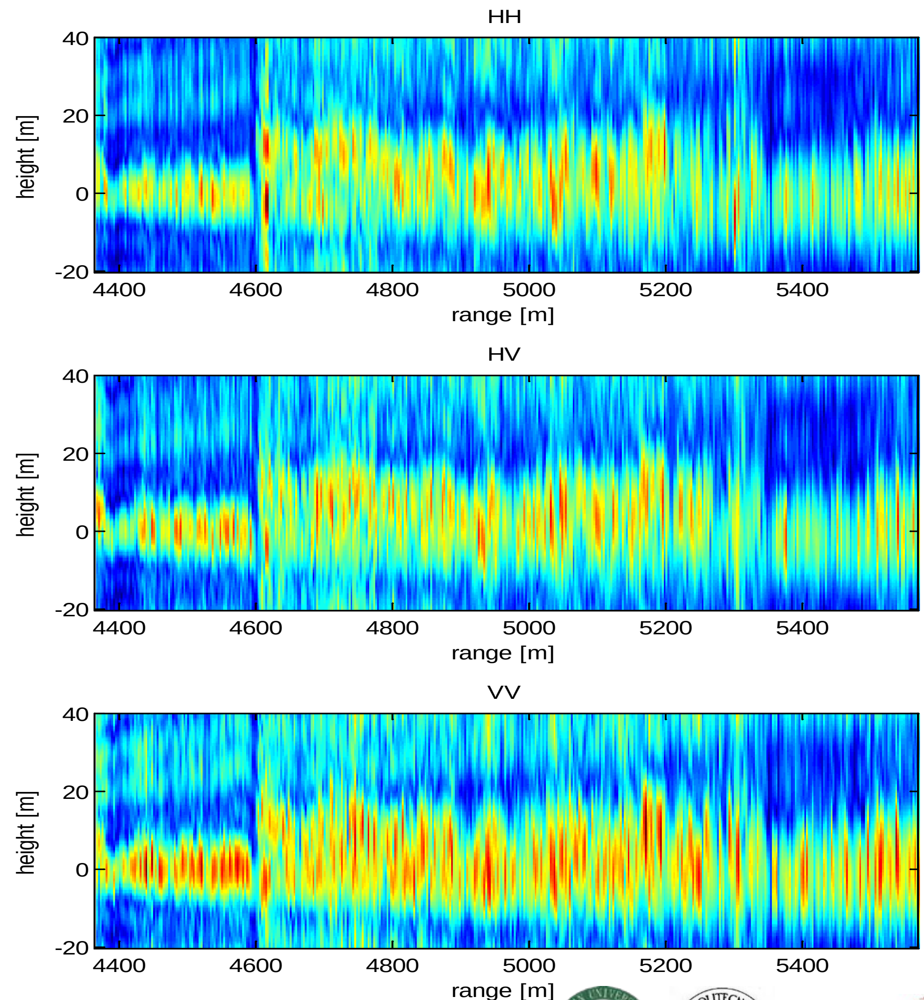
Lidar DEM not subtracted (`rem_dem_flag=0`)

$\Rightarrow$ *Reference height = DEM_avg = 200 m*



LIDAR DEM subtracted (`rem_dem_flag=1`)

$\Rightarrow$ *Reference height = LIDAR DEM*



Geocoding

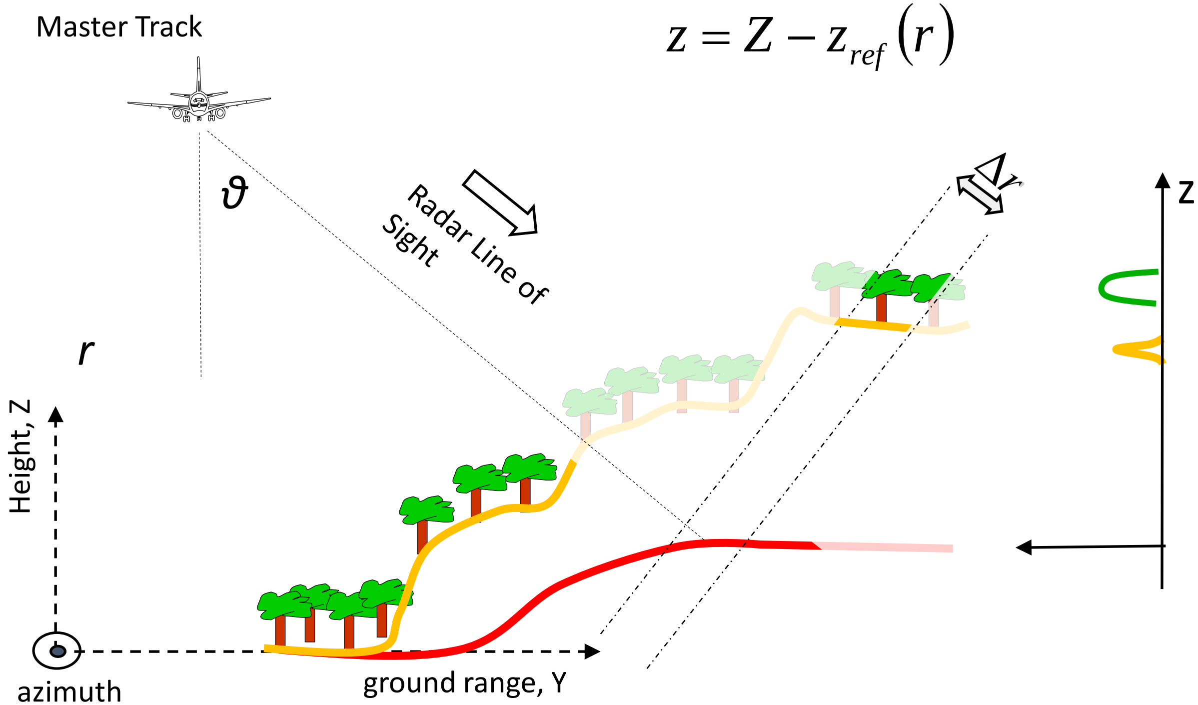
Tomographic profiles have been generated in the coordinate system (*r*,*z*):

- r = (Zero-Doppler) distance from the Master track
- z = height w.r.t. the reference DEM

⇒ A point at coordinates (*Y*,*Z*) in the ground range plane is found at

$$r = \sqrt{\left(Y_{Master} - Y\right)^2 + \left(Z_{Master} - Z\right)^2}$$

$$z = Z - z_{ref}\left(r\right)$$



Red = $z_{ref}(r)$ = Reference terrain elevation

Geocode_TomoSAR.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% pixel index
[t,a0] = min(abs(az_ax-az_profile_m));
% Master position
Sy = interp1(S{Master}.x,S{Master}.y,az_profile_m);
Sz = interp1(S{Master}.x,S{Master}.z,az_profile_m);
% Terrain elevation
dem = DEM(:,a0)';
% Forest height
for_h = FOR_H(:,a0)';
% ground range as a function of slant range
y_of_r = sqrt(rg_ax.^2 - (Sz-dem).^2) + Sy;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% absolute ground range axis
dy = 1;
y_ax_abs = [min(y_of_r)-5:dy:max(y_of_r)+5];
% absolute height axis
z_ax_abs = [min(dem)-10:dz:max(dem)+30];

% ground range as a function of slant range
y_of_r = sqrt(rg_ax.^2 - (Sz-dem).^2) + Sy;

% resample lidar dem and lidar forest height from range to ground range
dem_gr = interp1(y_of_r,dem,y_ax_abs,'linear',nan);
for_h_gr = interp1(y_of_r,for_h,y_ax_abs,'linear',nan);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Geocode_TomoSAR.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ground range, height coordinates
[Za,Ya] = ndgrid(z_ax_abs,y_ax_abs);
% slant range
R = sqrt( (Sy-Ya).^2 + (Sz-Za).^2 );
% reference dem
Z_ref = interp1(rg_ax,dem,R,'linear','extrap');
% height w.r.t. reference dem
Z = Za - Z_ref;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Geocode tomograms
for pol = 1:3
    tomo_sar = Tomo_filt{pol};
    tomo_sar = tomo_sar./max(tomo_sar(:));
    % Geocoded tomogram
    tomo_geo = interp2(rg_ax,z_ax,tomo_sar,R,Z);

    % Geocoded tomogram - height w.r.t. Lidar
    tomo_geo(isnan(tomo_geo)) = 0;
    for y = 1:length(y_ax_abs)
        tomo_geo_rel(:,y) = interp1(z_ax_abs,tomo_geo(:,y),z_ax + dem_gr(y));
    end

          %%%%%%%%%%%%%%%%%%%% Draw pictures here%%%%%%%%%%%%%%%%%%%%%%%%%%5

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Geocoding - Results

SAR geometry

Slant range – height w.r.t. reference DEM

Note: Lidar forest height not matched

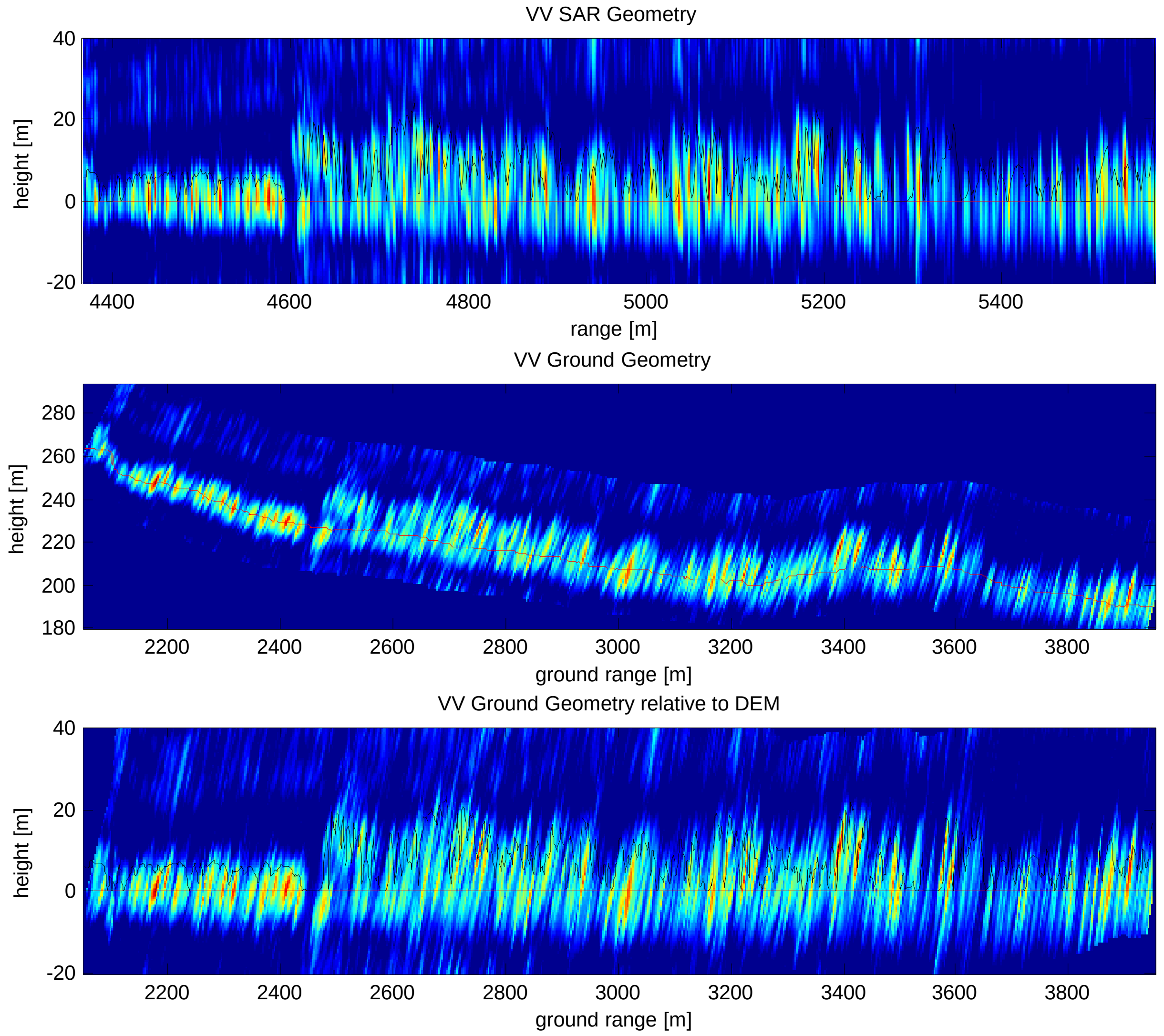
Ground geometry

Ground range – height

Ground geometry w.r.t. reference DEM

Ground range – height w.r.t. reference DEM

Note: Lidar forest height well matched



Red = Lidar terrain Black = Lidar forest height

%%
%%
%%
%%
%%%%%%%%%%%%%%%%%% THANK YOU %%%%%%%%%%%%%%%%%
%%
%%
%
% FEEL FREE TO CONTACT ME AT:
% EMAIL: stefano.tebaldini@polimi.it
% TEL: +390223993614
%%
%%

Questions?

balz@whu.edu.cn