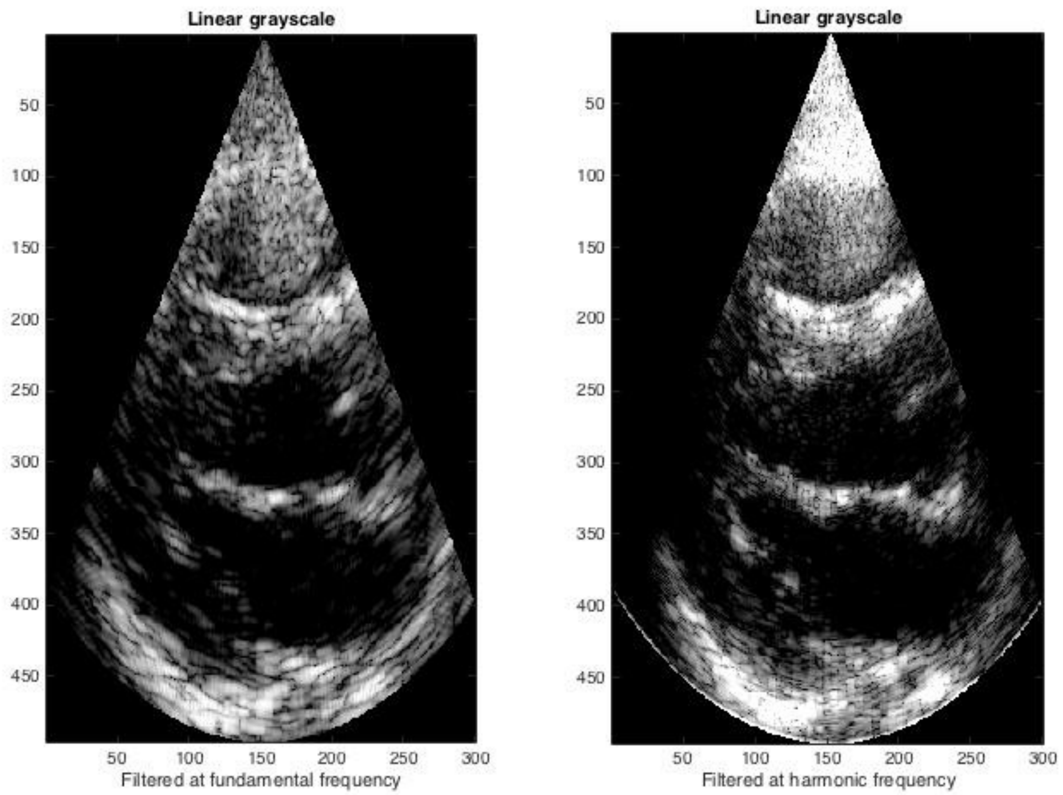
Exercise 5: Image Formation Signal Chain

Made by: Even Flørenæs

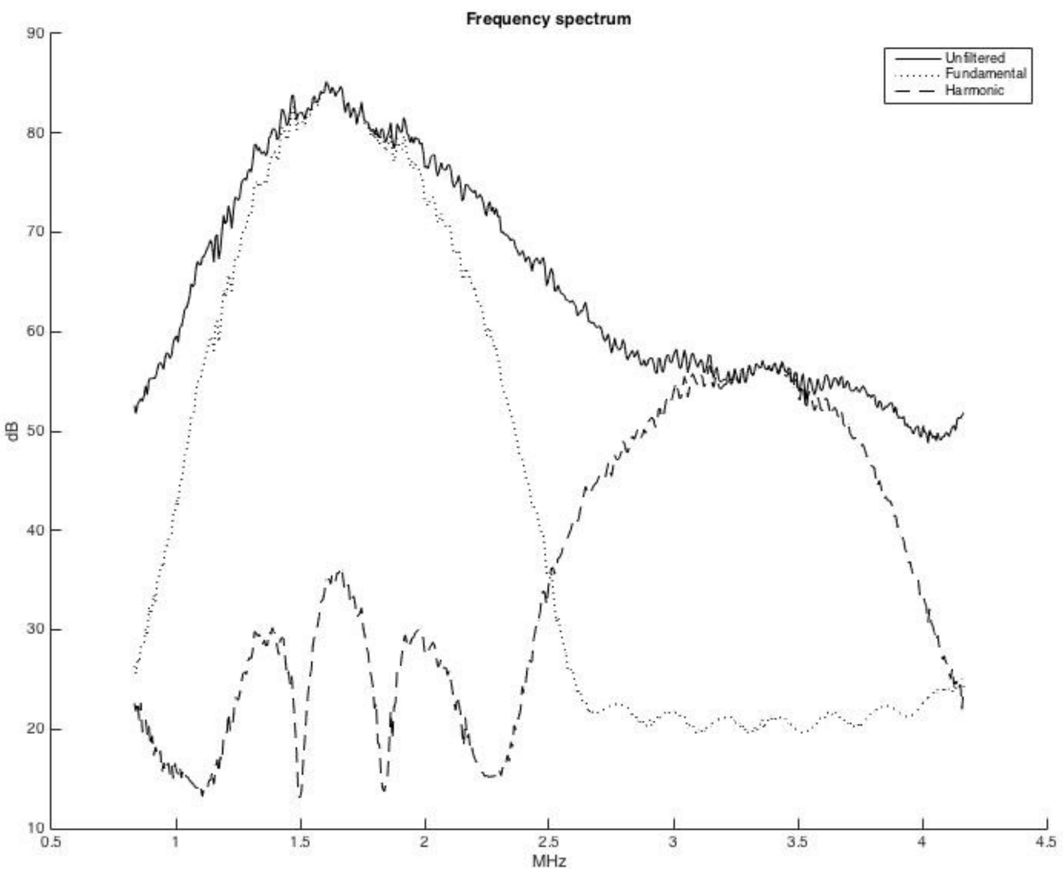
Task 1: Fundamental versus harmonic imaging

The figure below is made by filtering the reflections from ultrasound imaging first around the fundamental frequency and then around the harmonic frequency. Looking at the produced plots, which depicts the same image, one can see differences in the level of the detail in the image. The harmonic image has better quality then the fundamental image. This can be seen by looking at the areas around the bigger black spot in lower middle of the image. In the fundamental image the areas around the the black spot is noisy and lacks information of the structure in the area. In the harmonic image the area has more information provided by bigger differences in the colour, giving difference in mass density and compressibility.



Task 2: Frequency spectrum

The figure below shows the frequency content of the iq-signal unfiltered, filtered around fundamental frequency and around the harmonic frequency. The unfiltered has frequency content in the spectrum from below 1 MHz to above 4 MHz. The fundamental image contains most information from the frequency spectrum between below 1 MHz to 2.5 MHz, and the harmonic contains most information from 2.5 MHz to above 4 MHz. Producing an image by higher frequency content will reduce the amounts of reverberations and side lobes artifacts which can be presented in both unfiltered and fundamental imaging. Avoiding reverberations and side lobes artefacts in combination with enhanced axial and lateral resolution will lead to better signal-to-noise ratio.

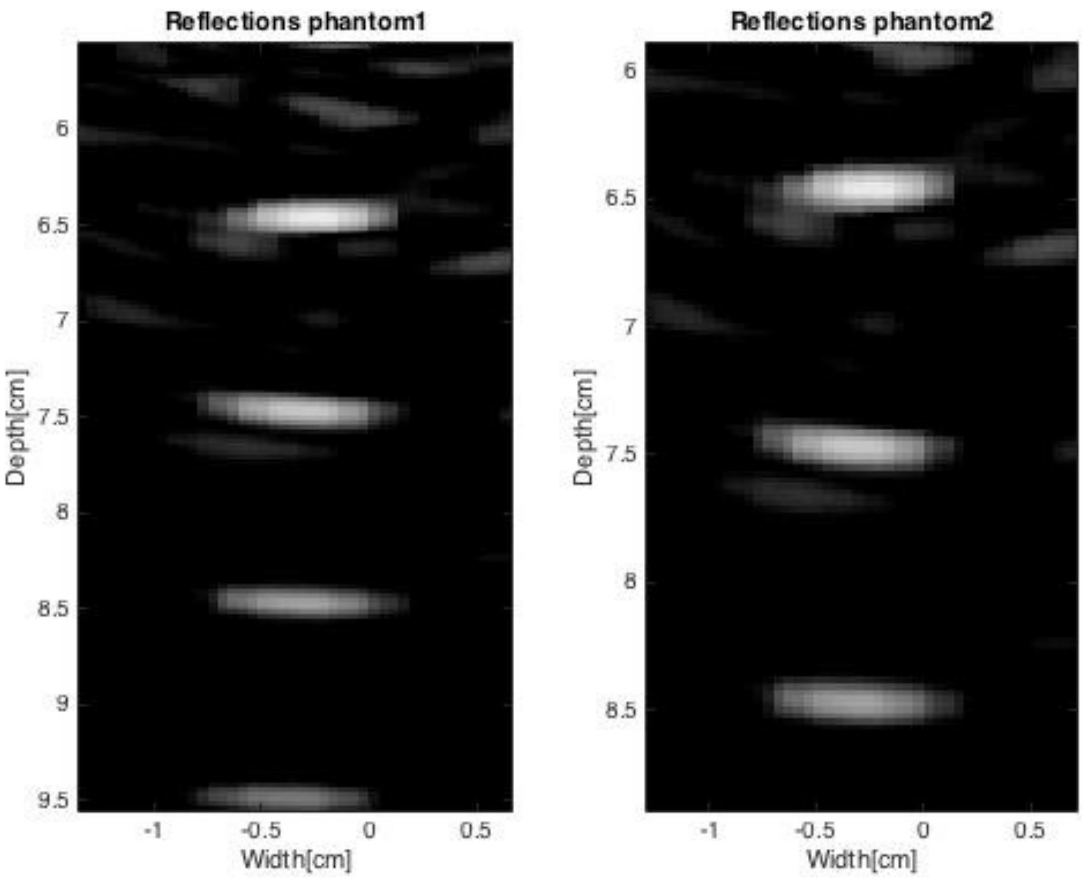


Task 3: Aperture

Beam width at depth 7.5 cm

Data collection	Beam width[cm]
Phantom 1	0,857
Phantom 2	0,8369

In the phantom 1 the signal transmitted used only half the aperture. In phantom 2 the whole aperture is used. For phantom 1 this will lead to loss in resolution in the reflections compared to using the whole aperture as in phantom 2. This is shown by measuring the width of the received pulses. The received pulse at 7.5 cm for phantom 1 is slightly wider than for phantom 2 for the same sent pulse. The width of the received pulse is a measurement of the resolution transverse to the beam, called lateral resolution. As the measurements shows, the lateral resolution is enhanced by using the whole aperture.

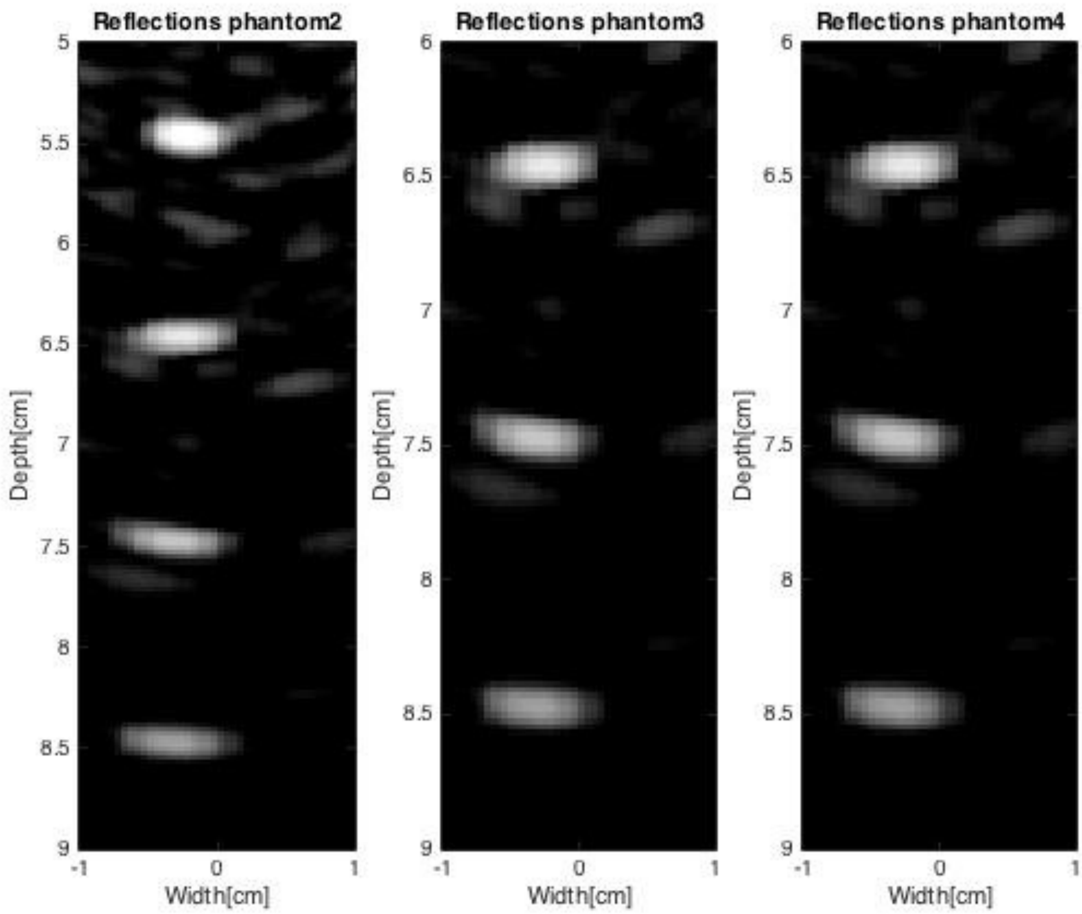


Task 4: Pulse length

Size along the beam at depth 7.5 cm

Data collection	Radial beam length [cm]
Phantom 2	0,18153
Phantom 3	0,16265
Phantom 4	0,15361

In this task the resolution in along the beam is depicted. The resolution along the direction of the beam is called the radial resolution, and it is determined by the length of the transmitted pulse. phantom 2 had a short pulse length, phantom 3 had a medium pulse length and phantom 4 had a long pulse length. This is shown in the resulting beam lengths in radial direction. In phantom 2 the beam length is 0.18 cm and gives an reflection with more potential for detailed reconstruction of the imaged area. For the longer transmitted pulse in phantom 3 and 4 we have a shorter beam length, and more potential for losing information in the imaged area.



Appendix I - Matlab script

```
%% Exercises without hand-in
file = 'phantom3d.mat';

load(file);

%%
% Image the tissue
figure,image(tissue);
colormap(gray(256));
xlabel('Beam number');
ylabel('Range number');
title('Beamspace');
%%
% Make the beam space axes
%theta axis
th1=s.tissue.StartAngle_rad;
dth=s.tissue.AngleIncrement_rad;
Nbeams=s.tissue.Beams;
th = th1:dth:((Nbeams-1)*dth+th1);
%range axis
r1=s.tissue.StartDepth_m;
dr=s.tissue.DepthIncrement_m;
Nr=s.tissue.Samples;
r=r1:dr:((Nr-1)*dr+r1);
%%
% Perform scan conversion and show resulting image
[tsc,xax,zax]=scanconvert(tissue,r,th);
figure,image(xax,zax,tsc);
colormap(gray(256));
axis image;
xlabel('cm');ylabel('cm');
title('Scanconverted image');

%%
f0 = s.iq.TxFreqIQ_Hz;
frs=s.iq.frsIQ_Hz;
fdemod=s.iq.fDemodIQ_Hz;
B=0.8e6; %0.8 MHz filter bandwidth
N=10; %filterorder
iqf=rectfreq(iq,frs,fdemod,f0,B,N);
amplitude=abs(iqf);
pow=amplitude.^2;

gain=-20;
dyn= 50; %30-70 dB
logpow=imagelog(pow,gain,dyn);
figure,image(logpow);
colormap(gray(dyn));
xlabel('Beam no.');
ylabel('Range no.');
title('Logaritmic compression')
```

```
figure;
imagesc(pow);
colormap(gray(dyn));
xlabel('Beam no.');
ylabel('Range no.');
title('Linear grayscale');


th1=s.iq.StartAngleIQ_rad;
dth=s.iq.AngleIncrementIQ_rad;
Nbeams=s.iq.BeamsIQ;
th = th1:dth:((Nbeams-1)*dth+th1);
r1=s.iq.StartDepthIQ_m;
dr=s.iq.DepthIncrementIQ_m;
Nr=s.iq.SamplesIQ;
r=r1:dr:((Nr-1)*dr+r1);
[logpowSc,xax,zax]=scanconvert(logpow,r,th);
figure,image(xax,zax,logpowSc);
xlabel('cm');ylabel('cm');
title('B-mode image generated from IQ data');
colormap(gray(dyn));

%%
load phantom_1;
n =65;
figure
imagelog(abs(iq).^2,-65,30);
fs=20e6; %RF sampling freq.
demod=s.iq.fDemodIQ_Hz; %IQ demodulation freq.
frsIq=s.iq.frsIQ_Hz; %IQ sampling freq
rf=iq2rf(iq,demod,frsIq,fs);

RFbeam=rf(:,n);
tax=[1:length(RFbeam)]/fs;
c=1540; %m/s
zax=tax*c/2*100; %cm
figure,plot(zax,RFbeam,'k');
ylabel('RF signal');

RFpow=RFbeam.^2;
figure,plot(zax,RFpow,'k');
ylabel('RF power');

RFsmooth=filter(ones(6,1)/6,1,RFpow);
figure,plot(zax,RFpow,'k');
ylabel('Smoothed RF power');

RFsmoothdB=10*log10(RFsmooth);
RFsmoothdB= RFsmoothdB-max(RFsmoothdB); %Normalization
figure,plot(zax,RFsmoothdB,'k');
ylabel('Smoothed RF power [dB]');

IQpow=abs(iq(:,n)).^2;
IQpowdB=10*log10(IQpow);
IQpowdB=IQpowdB-max(IQpowdB); %Normalizing
dz=s.iq.DepthIncrementIQ_m;
```

```
zaxIQ=dz*[1:length(IQpowdB)]*100; %IQ depth axis [cm]
plot(zaxIQ,IQpowdB,'k');
ylabel('IQ power [dB]');xlabel('cm');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Exercises with hand-in
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Task 1
file = 'lasse_1.mat';
load(file)
fundamentalFreq = s.iq.TxFreqIQ_Hz;
harmonicFreq = 2*fundamentalFreq;
samplingFreq = s.iq.frsIQ_Hz;
demodFreq = s.iq.fDemodIQ_Hz;
bandWidth = 0.8e6;
filterOrder = 10;

iqFilteredFundamentalFreq =
rectfreq(iq,samplingFreq,demodFreq,fundamentalFreq,bandWidth,filterOrder);
iqFilteredHarmonicFreq =
rectfreq(iq,samplingFreq,demodFreq,harmonicFreq,bandWidth,filterOrder);

% Axes for scan conversion
%theta axis
th1=s.iq.StartAngleIQ_rad;
dth=s.iq.AngleIncrementIQ_rad;
Nbeams=s.iq.BeamsIQ;
th = th1:dth:((Nbeams-1)*dth+th1);
%range axis
r1=s.iq.StartDepthIQ_m;
dr=s.iq.DepthIncrementIQ_m;
Nr=s.iq.SamplesIQ;
r=r1:dr:((Nr-1)*dr+r1);

[iqFundFreqScanConvert,xAxisFund,zAxisFund] = scanconvert(iqFilteredFundamentalFreq,r,th);
[iqHarFreqScanConvert,xAxisHar,zAxisHar] = scanconvert(iqFilteredHarmonicFreq,r,th);

iqFundFreqScanConvertAbs = abs(iqFundFreqScanConvert);
iqHarFreqScanConvertAbs = abs(iqHarFreqScanConvert);
powFundFreq = iqFundFreqScanConvertAbs.^2;
powHarFreq = iqHarFreqScanConvertAbs.^2;
gainFundFreq =-50;
gainHarFreq = -15;
dynamicRange = 50;

logPowFundFreq = imagelog(powFundFreq,gainFundFreq,dynamicRange);
logPowHarFreq = imagelog(powHarFreq,gainHarFreq,dynamicRange);
figure,subplot(1,2,1),image(logPowFundFreq),title('Linear grayscale'),xlabel('Filtered at
fundamental frequency'),colormap(gray(dyn));
axis equal tight
subplot(1,2,2),image(logPowHarFreq),title('Linear grayscale'),xlabel('Filtered at harmonic
frequency'),colormap(gray(dyn));
axis equal tight

%% Task 2
iqf = iqFilteredFundamentalFreq;
```

```
iqh = iqFilteredHarmonicFreq;
[Plin,Plog,fax] = iqspect(iq,samplingFreq,demodFreq,512);
[Pflin,Pflog,fax] = iqspect(iqf,samplingFreq,demodFreq,512);
[Phlin,Phlog,fax] = iqspect(iqh,samplingFreq,demodFreq,512);

figure; hold on
plot(fax/1e6,Plog,'k-');
plot(fax/1e6,Pflog,'k:');
plot(fax/1e6,Phlog,'k--');
title('Frequency spectrum');
legend('Unfiltered','Fundamental','Harmonic');
ylabel('dB');xlabel('MHz');
hold off
%% Task 3

images = {'phantom_1.mat','phantom_2.mat', 'phantom_3.mat', 'phantom_4.mat'};
powPhantom = cell(1,4);
for i = 1:length(images)
    Image = images{1};
    load(Image)
    fundamentalFreq = s.iq.TxFreqIQ_Hz;
    samplingFreq = s.iq.frsIQ_Hz;
    demodFreq = s.iq.fDemodIQ_Hz;
    bandWidth = 0.8e6;
    filterOrder = 10;

    th1=s.iq.StartAngleIQ_rad;
    dth=s.iq.AngleIncrementIQ_rad;
    Nbeams=s.iq.BeamsIQ;
    th = th1:dth:((Nbeams-1)*dth+th1);
    r1=s.iq.StartDepthIQ_m;
    dr=s.iq.DepthIncrementIQ_m;
    Nr=s.iq.SamplesIQ;
    r=r1:dr:((Nr-1)*dr+r1);

    iqFilteredFundFreq =
rectfreq(iq,samplingFreq,demodFreq,fundamentalFreq,bandWidth,filterOrder);


    iqFilteredFundFreqAbs = abs(iqFilteredFundFreq);
    powFundFreq = iqFilteredFundFreqAbs.^2;
    powPhantom{i} = powFundFreq;

end
powPhantom1 = powPhantom{1};
powPhantom2 = powPhantom{2};
powPhantom3 = powPhantom{3};
powPhantom4 = powPhantom{4};
gainPhantom1 = -60;
gainPhantom2 = -60;
gainPhantom3 = -60;
gainPhantom4 = -60;
dynamicRange = 30;
logPowPhantom1= imagelog(powPhantom1,gainPhantom1,dynamicRange);
logPowPhantom2= imagelog(powPhantom2,gainPhantom2,dynamicRange);
logPowPhantom3 = imagelog(powPhantom3,gainPhantom3,dynamicRange);
```

```
logPowPhantom4 = imagelog(powPhantom4,gainPhantom4,dynamicRange);
[logPowPhantom1Sc,xax1,zax1] = scanconvert(logPowPhantom1,r,th);
[logPowPhantom2Sc,~,~] = scanconvert(logPowPhantom2,r,th);
[logPowPhantom3Sc,~,~] = scanconvert(logPowPhantom3,r,th);
[logPowPhantom4Sc,~,~] = scanconvert(logPowPhantom4,r,th);

xax = xax1*100;
zax = zax1*100;

figure(13),subplot(1,2,1),image(xax,zax,logPowPhantom1Sc),ylim([5 9]),xlim([-1
1]),colormap(gray(dyn)),title('Reflections phantom1'),xlabel('Width[cm]'),ylabel('Depth[cm]');
%assumes scan converted images with axis scaled in [cm]
%axis equal tight
subplot(1,2,2),image(xax,zax,logPowPhantom2Sc),colormap(gray(dyn)),ylim([6 9]),xlim([-1
1]),title('Reflections phantom2'),xlabel('Width[cm]'),ylabel('Depth[cm]');
%axis equal tight

%% Task  4

figure(14),subplot(1,3,1),image(xax,zax,logPowPhantom2Sc),ylim([5 9]),xlim([-1
1]),colormap(gray(dyn)),title('Reflections phantom2'),xlabel('Width[cm]'),ylabel('Depth[cm]');
%assumes scan converted images with axis scaled in [cm]
%axis equal tight
subplot(1,3,2),image(xax,zax,logPowPhantom3Sc),colormap(gray(dyn)),ylim([6 9]),xlim([-1
1]),title('Reflections phantom3'),xlabel('Width[cm]'),ylabel('Depth[cm]');
%axis equal tight
subplot(1,3,3),image(xax,zax,logPowPhantom4Sc),colormap(gray(dyn)),ylim([6 9]),xlim([-1
1]),title('Reflections phantom4'),xlabel('Width[cm]'),ylabel('Depth[cm]');
[x,y]=ginput(2);
disp(['Distance: ',num2str(sqrt(diff(x).^2+diff(y).^2))]);
```