```
close all
clear all
clc


%% Raw data
% Load data
load CH_07_MoCo_5cm.mat

% Plot original image
figure; image(abs(ch(:,1:10:end))/100); title("Data after process");
colormap('gray'); axis("off");

%% Parameteres and importamt results
% Bandwidth = 60 MHz
% fs = 60 MHz (each sample 2.5 meters (horizontal))
% A = 0.2 m
% fc = 2340 MHz
% direct signal = -100
% sol = c = 3e8

sol = 299792458; % Speed of light
fs = 60e6; % Sample frequency
fc = 2.34e9; % Carrier frequency
lambda = sol/2340e6; % Wave length

R = ((0:1199)'+100)*(sol/(2*fs)); % Range resolution = c/2*B
x = (-32768:32767)*0.05; % Azimute resolution, samples separated by 5 cm
d = sqrt(R.^2*ones(1,65536) + ones(1200,1)*x.^2); % Distance to object
theta = atan2(ones(1200,1)*x, R*ones(1,65536)) * (180/pi); % beam angle of the
antenna - pointing direction


A = sinc(theta/30) .* (abs(theta)<=90) .* (abs(theta)>=1); % Antenna Pattern

% resises the image, adds zeros to left and right CH is now a 1200 by 65536
CH = single([zeros(1200,(2^16-14000)/2) conj(ch)
zeros(1200,(2^16-14002)/2)]);
figure; image(abs(CH(:,:))/500); title("Raw data but resised");

S = A .* exp(-4j*pi/lambda*d); % SAR signal
S = single(S);

%% SAR processing gain
% Azimuth compression
azimuthCompression = abs(ifft(conj(fft(fftshift(S,2), [], 2)).*fft(CH, [],2),
[],2)) / 5000;
figure; image(azimuthCompression); title("Azimuth Compression");
colormap('gray'); axis("off"); % Correlation of conjugated of S by CH,
represents a correlation in time

% Azimuth Decimation
azimuthDecimation = resample(double(abs(ifft(conj(fft(fftshift(S,2), [],
2)).*fft(CH, [],2), [],2))'), 1, 8)' / 5000;
figure; image(x, R, azimuthDecimation); title("After resample 8:1");
colormap('gray'); axis("off");% Same as before but with the correct axis values

%% Range Migration
%figure; image(abs(CH) / 500); title("raw data: line are not perfectly
%horizontal"); % Note that in raw data, the lines are not horizontal, we need
to compensate that. It's am hiperbolic because as the plane move, the distance
to the object increases
figure; image(abs(fftshift(fft( CH, [],2), 2)) / 1000); title("FFT of raw data.
Low frequencies in the middle"); colormap('gray'); axis("off"); % center of low
frequencies that correspond to the plane on top of the object. Smate as before
but with low freq in center

% distorce the line line in order to make them straight lines, R varies
% with an hiporbole, we want to make it staight by using an interpolation

w = 2*pi*(-32768:32767)/65536/0.05; % Frequency in space -> radiants per meter
(AKA doppler?)

CHF = fftshift(fft([zeros(100,65536); CH; zeros(400, 65536)], [], 2),2); % fft
of raw data with low frequencies in center, resized bu adding zeros on top and
bottom
%figure; image(abs(CHF)/20000); title("fft of raw data resized with low freq
in middle");

% Initial sample value
NN = round(1300./sqrt(1-(sol/2340e6*w/(4*pi)).^2)); % we have 1300 samples
times 1/ sqrt(...) we want N samples
% figure; plot(NN, '.-'); grid; title("Samples times the scale factor");

% Low pass filter along the columns | Compensating according to NN
for k= 1:65536
    dummy = fft(CHF(1: NN(k),k));
    dummy = dummy([1:650 end-649:end]); % Midle is fine, so we don't mess with
it
    CHF(1:1300,k) = ifft(dummy);
end

% Now we can consider the lines to be straight when in fact they are hiperbolic
figure; image(abs(CHF)/7000); title("Raw data after range migration");
colormap('gray'); axis("off");

%% Final Result
% Final image, plot the correlated and resampled signal, but with the ridge
regression compensation
figure; image(x, R, resample(double(abs(ifft(conj(fft(fftshift(S,2), [],
2)).*fftshift(CHF(101:1300,:),2), [],2))'), 1, 25)' / 7000); title("Final
Image") %estamos a fazer a correlação com a linha horizontal (,2) mas na verdade
esta linha é uma parábola
colormap('gray'); axis("off");
```