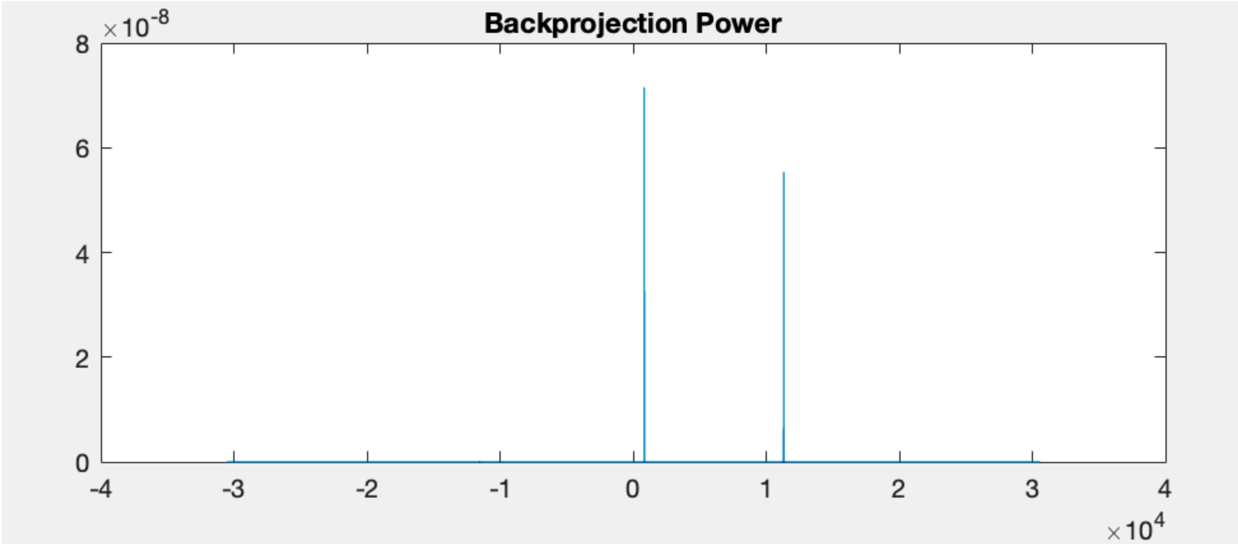
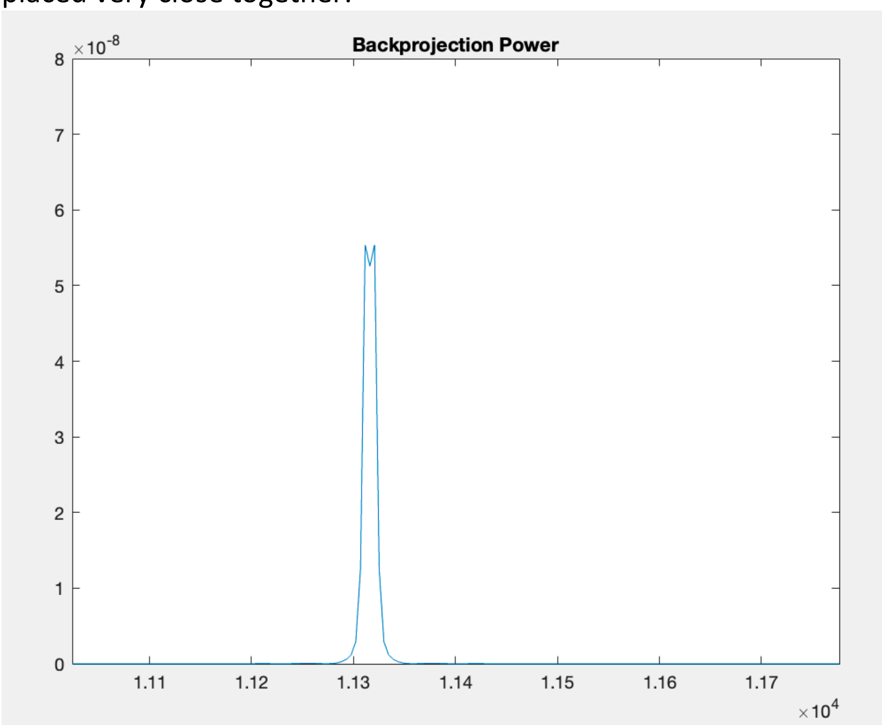
Backprojection 1D solution:



If we zoom in on the right-most peak, we get a hint that it is actually two corner reflectors placed very close together:



If we oversampled more, we would get a better hint of this.

MATLAB code:

```
% Backprojection algorithm from 1D simulated ALOS data
% Elizabeth Wig, 10/31/23

lambda = 0.23605710;    % Wavelength of ALOS (m)
c = 299792458;          % Speed of light (m/s)
tau = 0.00002700;       % Pulse length (s)
prf = 1655.629000;      % PRF (Hz)
v = 7592;               % Spacecraft velocity (m/s)
z = 692000;             % Spacecraft altitude (m)
theta = 38.7 * pi/180.; % Look angle (rad)
w = 3.1;                % Antenna width (m)
l = 8.9;                % Antenna length (m)
theta_L_a = 0.866 * lambda/l; % Half power beamwidth
r0 = z/cos(theta);     % Range to center bin

fid = fopen('alossim_1d.dat');
data_iq = fread(fid,'float32');
fclose(fid);

data = data_iq(1:2:end)+1i*data_iq(2:2:end);

% Paul's code
E_cr_sim_1d_cpx= data; %np.fromfile("alossim_1d.dat",dtype=np.complex64)
s_s_ref = - r0 * theta_L_a / 2.; % Start pixel of reference function
s_e_ref = + r0 * theta_L_a / 2.; % End pixel of reference function

Delta_s = v/prf; % Distance traveled between pulses by platform

n_s_ref = round((s_e_ref-s_s_ref)/Delta_s); % Number of points in az
s_sa_ref = linspace(s_s_ref,s_e_ref,n_s_ref); % Array of locations in
azimuth
rho_sa = sqrt(r0.^2+(s_sa_ref).^2); % Array of ranges in reference function
phi_sa = 4.*pi*rho_sa/lambda; % Phase of reference function
ref_sa = exp(1j*phi_sa); % Reference function of phase adjustments

s_0 = 0; % Reference azimuth position
s_s_sim = s_0 - 3. * r0 * theta_L_a / 2.;   % total of 3 beamwidths for
simulation
s_e_sim = s_0 + 3. * r0 * theta_L_a / 2.;  % total of 3 beamwidths for
simulation
n_s_sim = round((s_e_sim-s_s_sim)/Delta_s); % Number of points in sim
s_sim = linspace(s_s_sim, s_e_sim, n_s_sim);
E_bp = zeros(size(s_sim));


for i = 1:length(s_sim) % for i,s in enumerate(s_sim):
    s = s_sim(i); % azimuth location to plot
    s_s_im = (s - r0 * theta_L_a / 2.); % s at start
    s_s_im(s_s_im<s_sim(1))=s_sim(1); % Clip
    s_s_im(s_s_im>s_sim(end))=s_sim(end); % Clip
    s_e_im = (s + r0 * theta_L_a / 2.); % s at end
    s_e_im(s_e_im<s_sim(1))=s_sim(1); % Clip
    s_e_im(s_e_im>s_sim(end))=s_sim(end); % Clip
    n_s_im = round((s_e_im-s_s_im)/Delta_s);
```

```
    s_sa = linspace(s_s_im,s_e_im, n_s_im);
    stind = round((s_s_im-s_sim(1))/Delta_s)+1; % Pixel corresponding to
starting index
    enind = stind+n_s_im-1; % Pixel location at ending index
    if length(E_cr_sim_1d_cpx(stind:enind)) == length(ref_sa) % if edges
aren't cut off
        E_bp(i) = sum(E_cr_sim_1d_cpx(stind:enind).*ref_sa.');
        %  power of backprojection = sum of E from start to end * reference
function
    end
end

figure; plot(s_sim,real(E_bp)); title('Backprojection Signal');
figure; plot(s_sim, abs(E_bp).^2); title('Backprojection Power');
```