```
clc;
clear all;
% Speed of light and lambda calculation
C = 2.99792458e8;
freq_Hz = 1420e6;
lambda_m = C/freq_Hz;

%read an image and capture its size, flip and take fft and fftshift. plot its real, imag and absolute value
H = imread("/home/dsp/Downloads/friendlyVRI-master/models/compact.png");
subplot(4,4,1)
contour(H);
title(’input image’);
n = size(H);
nx = n(1);
ny = n(2);
modelImgArr = flipud(H);
modelFFTarr = fftshift(fft2(modelImgArr));
subplot(4,4,2)
contour(real(modelFFTarr))
title(’real’)
subplot(4,4,3)
contour(imag(modelFFTarr))
title(’imag’);
subplot(4,4,4)
contour(abs(modelFFTarr))
title(’abs’);

%Assumed resolution in arcsec
pixScaleImg_asec = 1;
pixScaleImg_lam = deg2rad(pixScaleImg_asec/3600.0);
fftScale_lam = 1.0/pixScaleImg_lam;
pixScaleFFTX_lam = 2.0*fftScale_lam/nx;
pixScaleFFTY_lam = 2.0*fftScale_lam/ny;
uvMaskArr = zeros(nx,ny);

% Calculate the hour angles
sampRate_deg = 60 * 15.0 / 3600.0;
sampRate_hr = 60 / 3600.0;
hastart= -1;
haend=  1;
nSamps = ((haend - (hastart))/sampRate_hr +1);
haArr_hr = linspace(hastart, haend, nSamps);
haArr_rad = deg2rad(haArr_hr * 15.0);

%Provide the parameters to get xyz coordinate and uv plane values(2 antennas)

lat1 = -30.312906;
lat1 = deg2rad(lat1);
%% For different positions of Antennas
%test1
% eastArr_m = [0 100];
%  northArr_m = [0 0];
%  UpArr_m = [0 0];
%Test2
%  eastArr_m = [0 -100];
%  northArr_m = [0 0];
%  UpArr_m = [0 0];
%Test3
%  eastArr_m = [0 0];
%  northArr_m = [0 100];
%  UpArr_m = [0 0];
%Test4
%  eastArr_m = [0 0];
%  northArr_m = [0 -100];
%  UpArr_m = [0 0];
%Test5
%  eastArr_m = [0 70.71];
%  northArr_m = [0 70.71];
%  UpArr_m = [0 0];
%Test6
%  eastArr_m = [0 -70.71];
%  northArr_m = [0 -70.71];
%  UpArr_m = [0 0];
%Test7
%  eastArr_m = [0 -70.71];
%  northArr_m = [0 70.71];
%  UpArr_m = [0 0];
%Test8
eastArr_m = [0 70.71];
northArr_m = [0 -70.71];
UpArr_m = [0 0];
%Plot the actual positions of Antenna
subplot(4,4,5)
plot3(eastArr_m,northArr_m,UpArr_m,'o')
xlabel('E')
ylabel('N')
zlabel('up')
title('ant pos')

nAnt = length(eastArr_m);
nBase = nAnt*((nAnt-1)/2);

            xArr_m = -northArr_m*sin(lat1);
            yArr_m = eastArr_m;
            zArr_m = northArr_m*cos(lat1);

%Calculation of baseline
Bx_m = zeros(nBase)
By_m = zeros(nBase)
Bz_m = zeros(nBase)

        n = 1;
            for i = 1: nAnt
                for j = i+1 : nAnt
                    Bx_m(n) = xArr_m(j) - xArr_m(i);
                    By_m(n) = yArr_m(j) - yArr_m(i);
                    Bz_m(n) = zArr_m(j) - zArr_m(i);
                    n += 1;
             end
        end
% Calculate vector of baseline lengths
lBase_m = sqrt(Bx_m.^2.0 + By_m.^2.0 + Bz_m.^2.0);

 %Angle of declination
 dec_deg = 20.0;
 dec_rad = deg2rad(-20);
 %Calculation of u,v plane
  for i = 1: nBase
                    u_m(i, :) = (Bx_m(i) * sin(haArr_rad) +
                                 By_m(i) * cos(haArr_rad));
                    v_m(i, :) = (-Bx_m(i) * sin(dec_rad) *
                                 cos(haArr_rad) +
                                 By_m(i) * sin(dec_rad) *
                                 sin(haArr_rad) +
                                 Bz_m(i) * cos(dec_rad));
  end

%Normalization with lambda
uArr_lam = u_m./lambda_m;
```

```
vArr_lam = v_m./lambda_m;
%wArr_lam = w_m./lambda_m;

subplot(4,4,6)
plot3([uArr_lam,-uArr_lam],[vArr_lam,-vArr_lam],’o’);
xlabel(’u’)
ylabel(’v’)
title(’ha +_1 de -20 l=21cm lat=-30’)

%Gridding
u_lam = uArr_lam(:);
v_lam = vArr_lam(:);

u_pixt = (u_lam+fftScale_lam)/pixScaleFFTX_lam;
v_pixt = (v_lam+fftScale_lam)/pixScaleFFTY_lam;
u2_pixt = (-u_lam+fftScale_lam)/pixScaleFFTX_lam;
v2_pixt = (-v_lam+fftScale_lam)/pixScaleFFTY_lam;

u_pix = fix(u_pixt);
v_pix = fix(v_pixt);
u2_pix = fix(u2_pixt);
v2_pix = fix(v2_pixt);

subplot(4,4,7)
plot([u_pix,u2_pix],[v_pix,v2_pix],’o’);
xlabel(’u’);
ylabel(’v’);
title(’translated pix’)

%Masking
for j = 1: length(u_pix)
uvMaskArr(v_pix(j), u_pix(j)) = 1;
uvMaskArr(v2_pix(j), u2_pix(j)) = 1;
end

subplot(4,4,8)
contour(uvMaskArr);
title(’uvMaskArr’);

 %Observed FFT
obsFFTarr = modelFFTarr.*uvMaskArr;

%subplot(4,4,9)
%contour(obsFFTarr);
%title(’obsFFTarr’);

subplot(4,4,9)
contour(real(obsFFTarr));
title(’real obsFFT’);

subplot(4,4,10)
contour(imag(obsFFTarr));
title(’imag obsFFT’);

subplot(4,4,11)
contour(abs(obsFFTarr));
title(’abs obsFFT’);

%beamArr = ifftshift(ifft2(uvMaskArr));

%Observed image

obsImgArr = ifft2(ifftshift(obsFFTarr));
%subplot(4,4,13)
%contour(obsImgArr);
```

```
%title(’obsImgArr’);

subplot(4,4,13)
contour(real(obsImgArr));
title(’real obsimg’);

subplot(4,4,14)
contour(imag(obsImgArr));
title(’imag obsimg’);

subplot(4,4,15)
contour(abs(obsImgArr));
title(’abs obsImg’)
%imshow(obsImgArr);
```