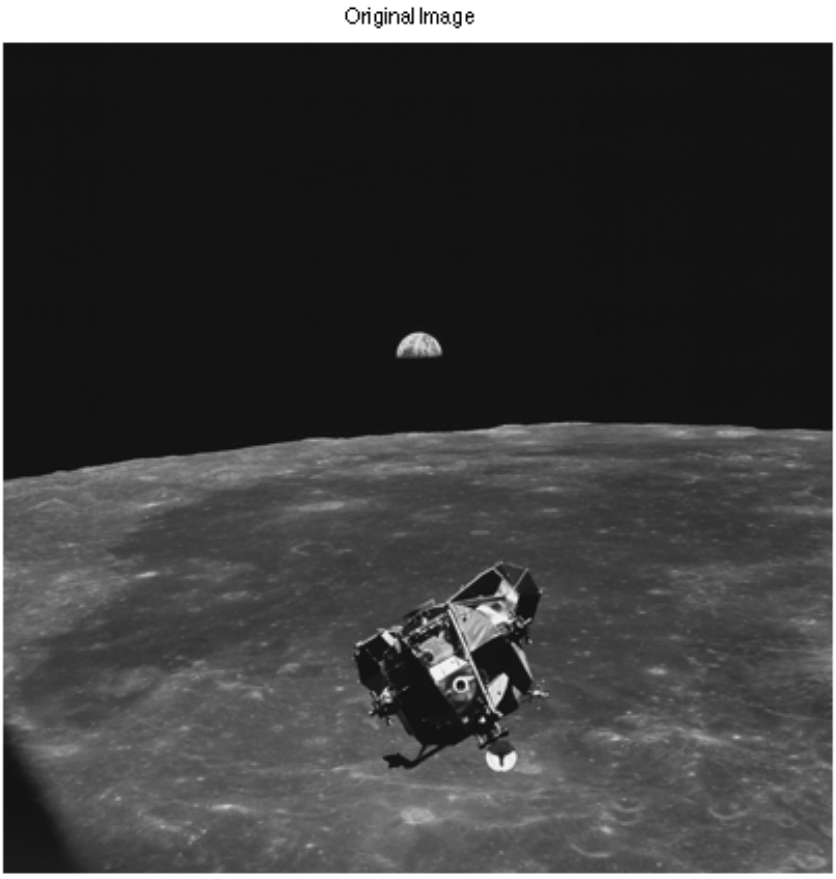
Table of Contents

EEE 178 Homework #4

MATLAB code and figures

```
clear all;
clc;
close all;
addpath ../commonFunctions

%image preprocessing!
I = getIMG('edgePhoto.jpg');
I = imresize(I,[400,400]);
I = rgb2gray(I);
I = im2double(I);
imshow(I);
title('Original Image');
```



Problem 1

An edge detection in the space domain.

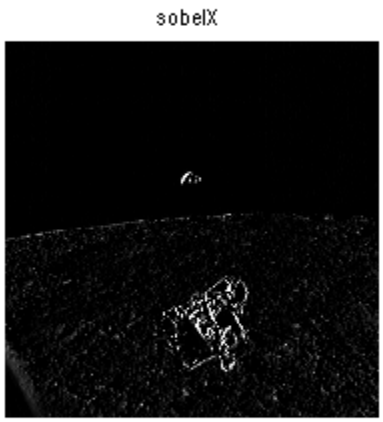
Parts 1 and 2

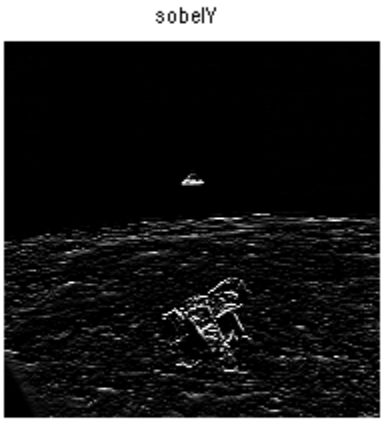
Apply the horizontal and vertical sobel mask to the image. show the results.

```
figure('name', 'sobel masks using custom convultion')
subplot(121)

sobelX=double([-1 0 1; -2 0 2; -1 0 1])
horiz = convolution(I, sobelX);
imshow(im2uint8(horiz));
title('sobelX')

subplot(122)
sobelY=double([1 2 1; 0 0 0; -1 -2 -1])
vert = convolution(I, sobelY);
imshow(im2uint8(vert));
```

```
title('sobelY')
```

```
sobelX =

    -1     0     1
    -2     0     2
    -1     0     1


sobelY =

     1     2     1
     0     0     0
    -1    -2    -1
```

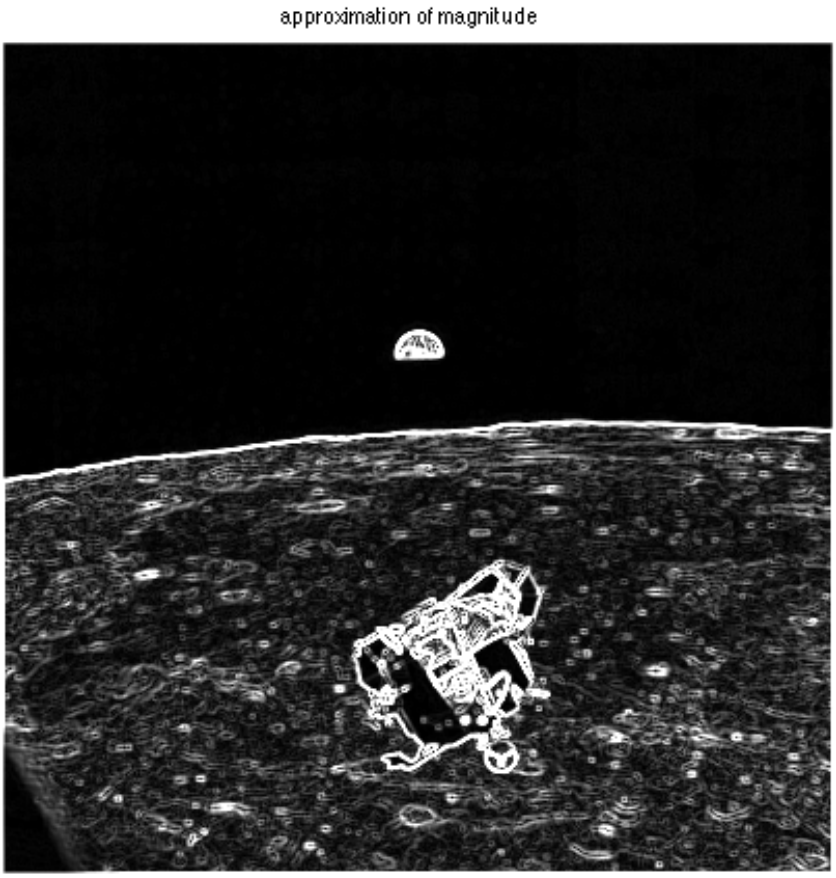




Part 3

Use the magnitude to combine edges and show result

```
figure('name','magnitude'); %pop pop
magnitude = abs(horiz) + abs(vert);
imshow(magnitude);
title('approximation of magnitude');
```

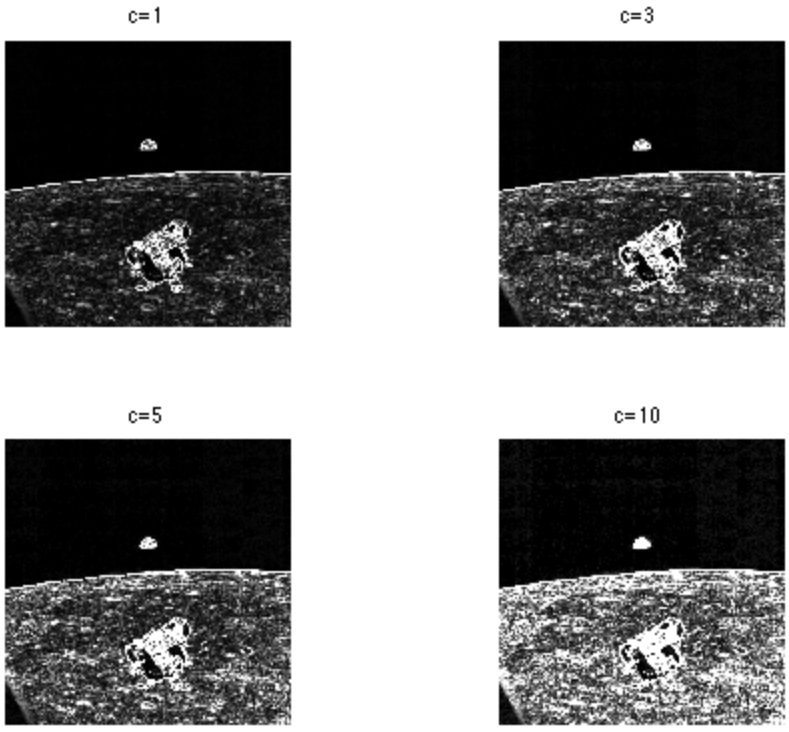


Part 4

Use different constant for c in sobel mask. try different values and see what happens.

```
figure('name', 'various values for c in sobel mask')
subplot(221)
c=1;
sobelX=double([-1 0 1; -c 0 c; -1 0 1]);
sobelY=double([1 c 1; 0 0 0; -1 -c -1]);
h=convolution(I,sobelX);
v=convolution(I,sobelY);
C=(abs(h)+abs(v));
imshow(C);
title('c=1')

subplot(222);
c=3;
sobelX=double([-1 0 1; -c 0 c; -1 0 1]);
sobelY=double([1 c 1; 0 0 0; -1 -c -1]);
h=convolution(I,sobelX);
v=convolution(I,sobelY);
C=(abs(h)+abs(v));
```

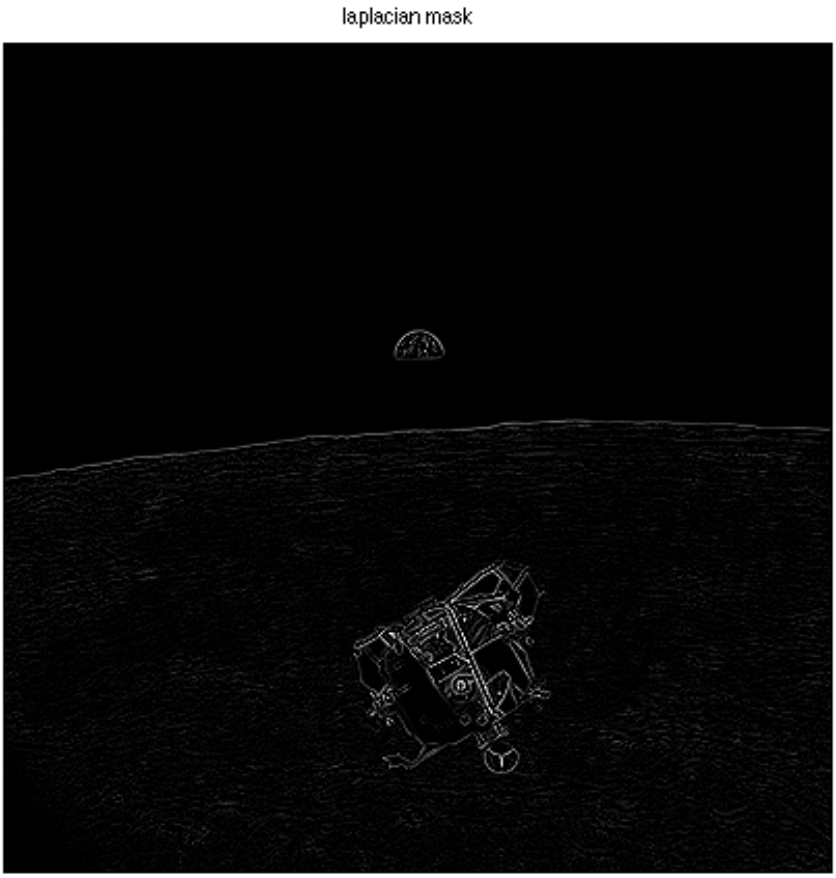
```
imshow(C);
title('c=3')

subplot(223)
c=5;
sobelX=double([-1 0 1; -c 0 c; -1 0 1]);
sobelY=double([1 c 1; 0 0 0; -1 -c -1]);
h=convolution(I,sobelX);
v=convolution(I,sobelY);
C=(abs(h)+abs(v));
imshow(C);
title('c=5')


subplot(224)
c=10;
sobelX=double([-1 0 1; -c 0 c; -1 0 1]);
sobelY=double([1 c 1; 0 0 0; -1 -c -1]);
startS=tic; %asked for part 6
h=convolution(I,sobelX);
v=convolution(I,sobelY);
C=(abs(h)+abs(v));
sobeltime= toc(startS);
imshow(C);
title('c=10')
```



Part 5

Use the Laplacian mask to edge detect same image, compare results.

```
figure('name', 'laplacian mask')
L=[0 1 0; 1 -4 1; 0 1 0];
startL = tic;
laplacedI=convolution(I,L);
laplacetime = toc(startL);
imshow(im2uint8(laplacedI)), title('laplacian mask');
rmpath ../commonFunctions
```



Part 6

compare computation times

```
sprintf('sobel approach: %s',sobeltime)
sprintf('laplacian approach: %s', laplacetime)
```

```
ans =

sobel approach: 1.483045e-01


ans =

laplacian approach: 7.407060e-02
```

Problem 2

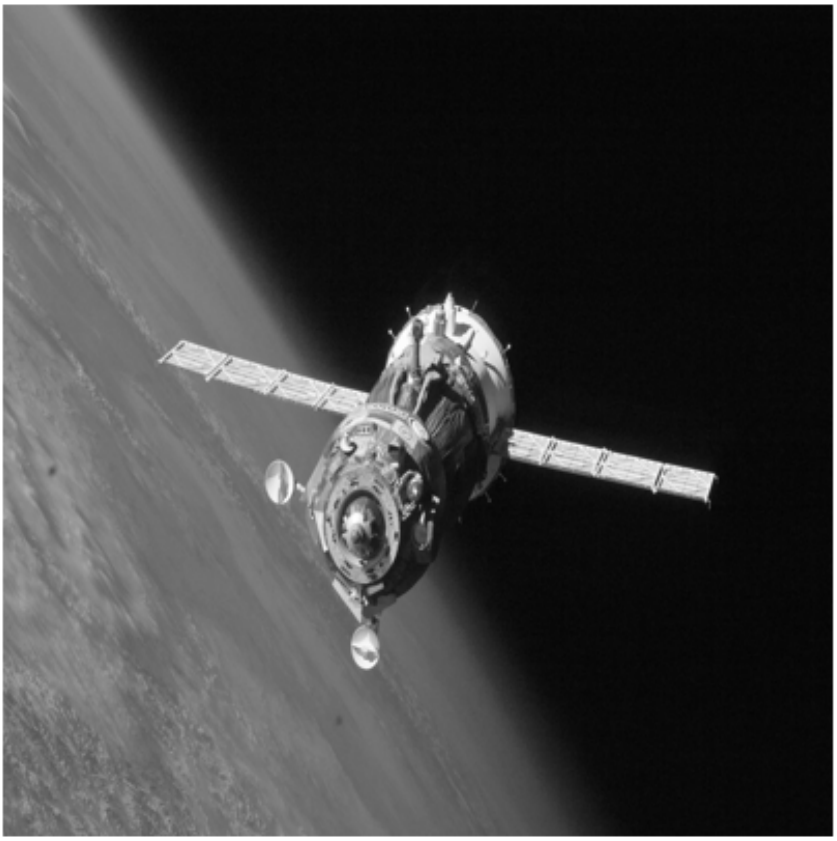
Edge detection in the frequency domain.

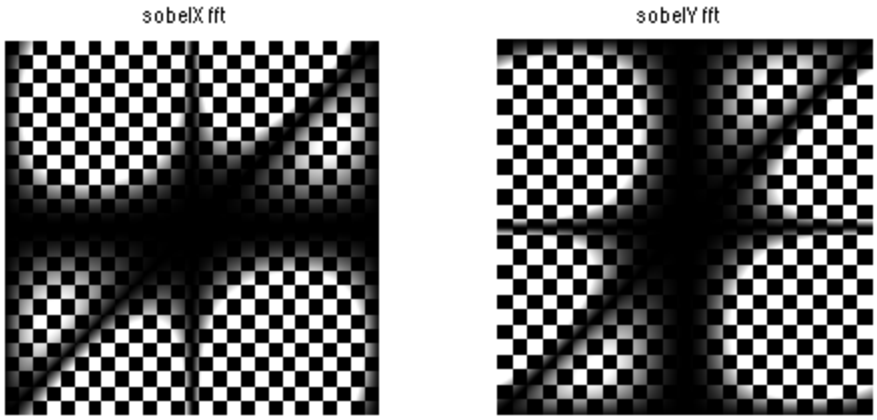
Part 1

Obtain the FFT of the horizontal and vertical Sobel masks. You need to perform zero padding.

```
addpath ../commonFunctions
I2 = getIMG('Soyuz_TMA-19_spacecraft_departs_the_ISS.jpg');
rmpath ../commonFunctions
I2 = im2double(rgb2gray(imresize(I2,[401,401])));
imshow(I2);

figure('name','ffts of sobel masks');
c=2;
sobelX=double([-1 0 1; -c 0 c; -1 0 1]);
sobelY=double([1 c 1; 0 0 0; -1 -c -1]);
%work on making this more general..........
sobelX1 = padarray(sobelX,[199,199]);
sobelFFTX=fft2(sobelX1);
%work on making this more general..........
sobelY1 = padarray(sobelY,[199,199]);
sobelFFTY=fft2(sobelY1);

subplot(121), imshow(sobelFFTX), title('sobelX fft');
subplot(122), imshow(sobelFFTY), title('sobelY fft');
```

```
Warning: Displaying real part of complex input.
Warning: Displaying real part of complex input.
```



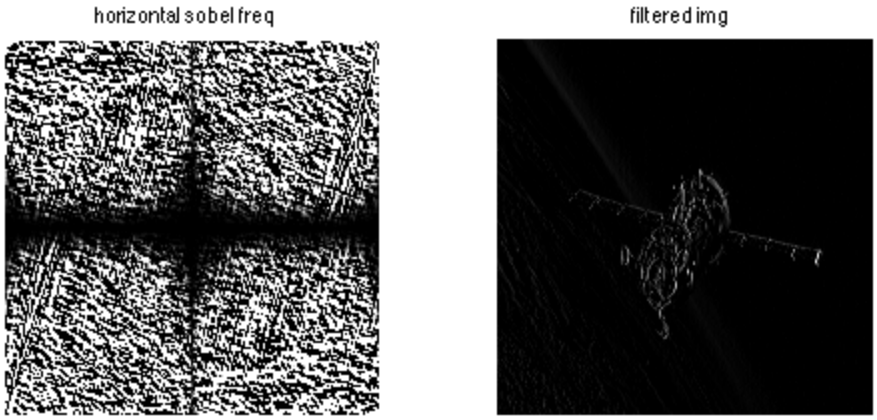


Part 2

Perform filtering in the frequency domain using the horizontal Sobel mask

```
figure('name', 'frequency horizontal filtering');
horizF=fft2(I2).*sobelFFTX;
horizS=ifft2(horizF);
dmin = min(min(abs(horizS)));
dmax = max(max(abs(horizS)));
subplot(121), imshow(horizF), title('horizontal sobel freq');
subplot(122), imshow(ifftshift(horizS),[dmin dmax]), title('filtered img');
```

Warning: Displaying real part of complex input.

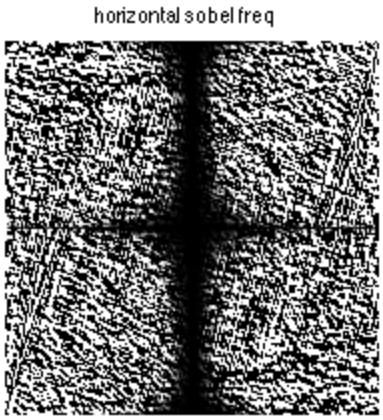


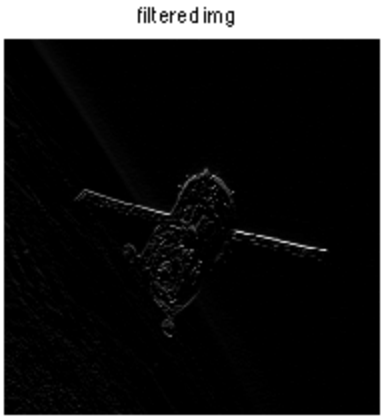
Part 3

Perform filtering in the frequency domain using the vertical Sobel mask

```
figure('name', 'frequency vertical filtering');
vertF=fft2(I2).*sobelFFTY;
vertS=ifft2(vertF);
dmin = min(min(abs(vertS)));
dmax = max(max(abs(vertS)));
subplot(121), imshow(vertF), title('horizontal sobel freq');
subplot(122), imshow(ifftshift(vertS),[dmin dmax]), title('filtered img');
```

```
Warning: Displaying real part of complex input.
```





Part 4

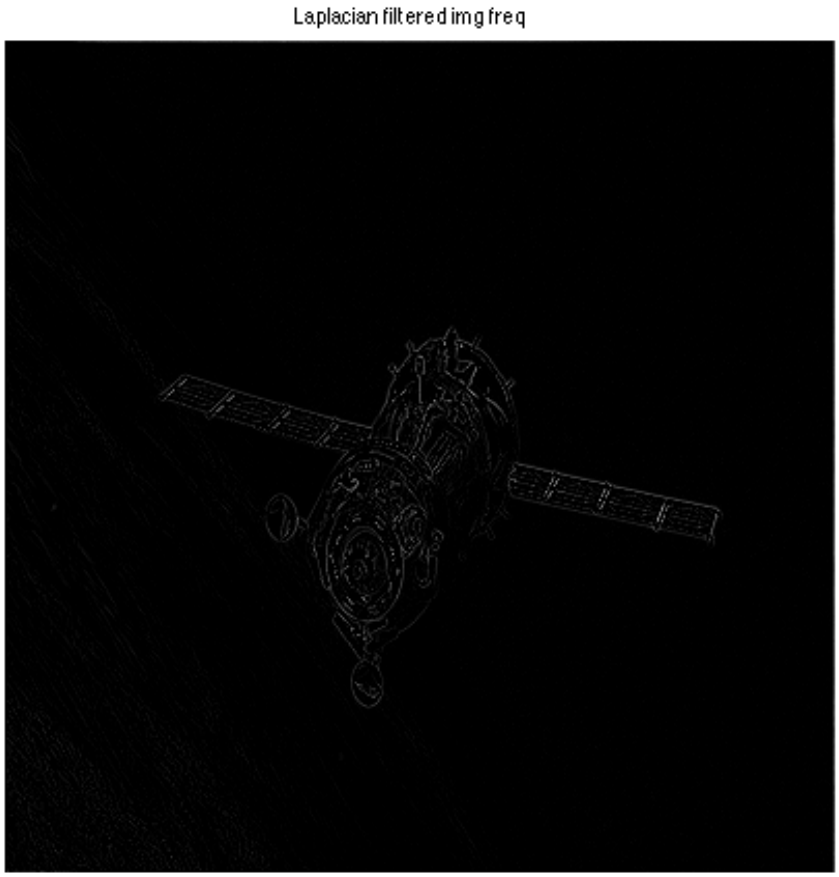
Obtain the FFT of the Laplacian mask. You need to perform zero padding.

```
L1 = fft2(padarray(L,[199,199]));
```

Part 5

Perform filtering in the frequency domain using Laplacian to detect the edges.

```
figure('name','laplacian frequency filtering');
lapF=fft2(I2).*L1;
lapS=ifft2(lapF);
dmin = min(min(abs(lapS)));
dmax = max(max(abs(lapS)));
imshow(ifftshift(lapS),[dmin dmax]), title('Laplacian filtered img freq');
```



Part 6

Construct a 3 by 3 mask for the Laplacian of Gaussian (LoG) and then obtain its FFT. You can use fspecial for this question

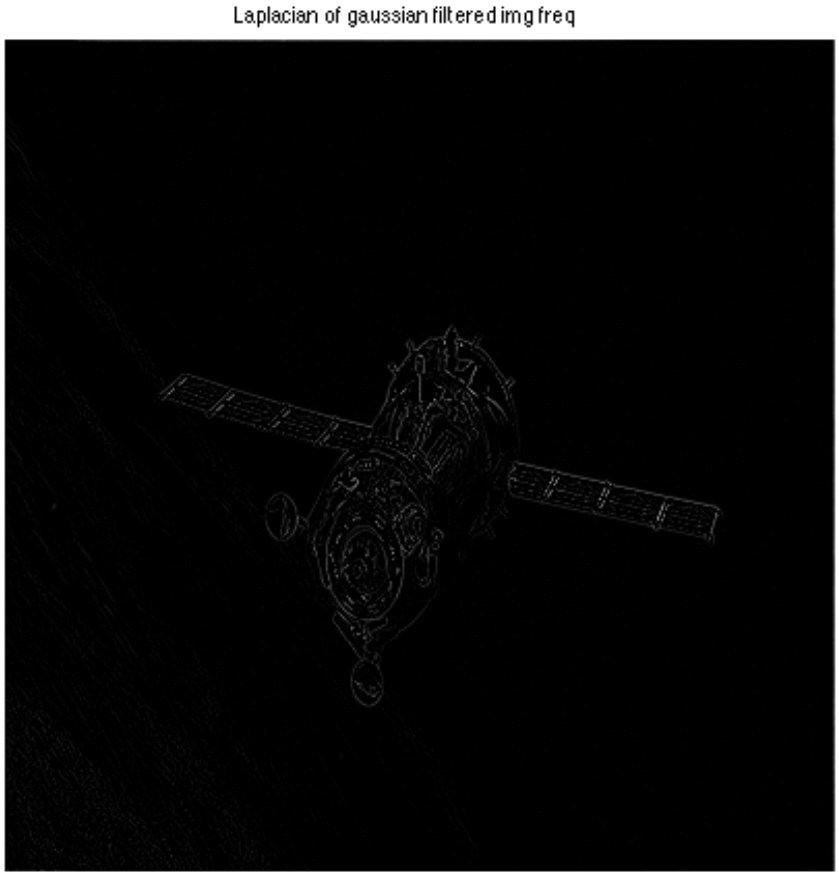
```
addpath ../commonFunctions
gauss=gaussfilter(3, 1/2);
LoG=L*gauss;
LoGF=fft2(LoG);
rmpath ../commonFunctions
%LoG=fspecial('log',[401,401]);
%LoGF=fft2(LoG);
```

Parts 7 and 8

Perform filtering in the frequency domain using LoG to detect the edges

```
figure('name','laplacian of gaussian freq filtering');
I3=fft2(I2).*L1;
lapS=ifft2(I3);
```

```
dmin = min(min(abs(lapS)));
dmax = max(max(abs(lapS)));
imshow(ifftshift(lapS),[dmin dmax]);
title('Laplacian of gaussian filtered img freq');
```

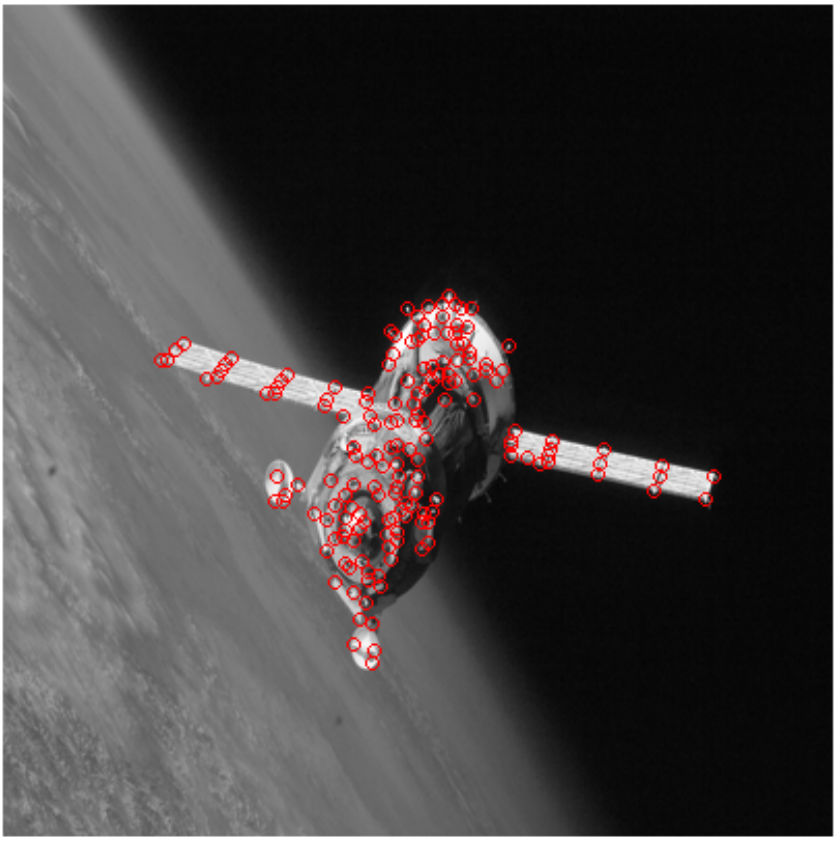


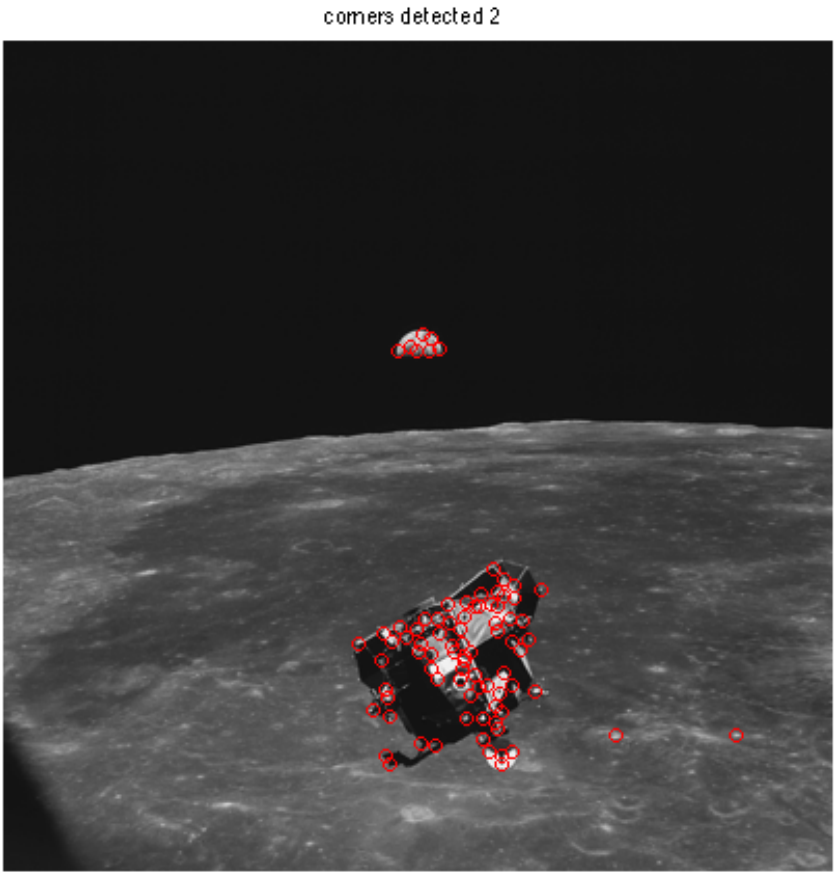
Problem 3

The Harris detector was introduced in 1988 for corner detection. An illustration is shown in figure 2. Apply the Harris detector to an image of your choice. Pick an image with visible corners. Mark all corners on the image in a similar way to figure 2

```
figure('name','corner detection');
C=corner(I2); %to find corner points
imshow(I2);hold on, plot(C(:,1),C(:,2),'ro')
title('corners detected')
figure('name','corner detection2')
C=corner(I); %to find corner points
imshow(I);hold on, plot(C(:,1),C(:,2),'ro')
title('corners detected 2');
```

corners detected





Published with MATLAB® R2013b