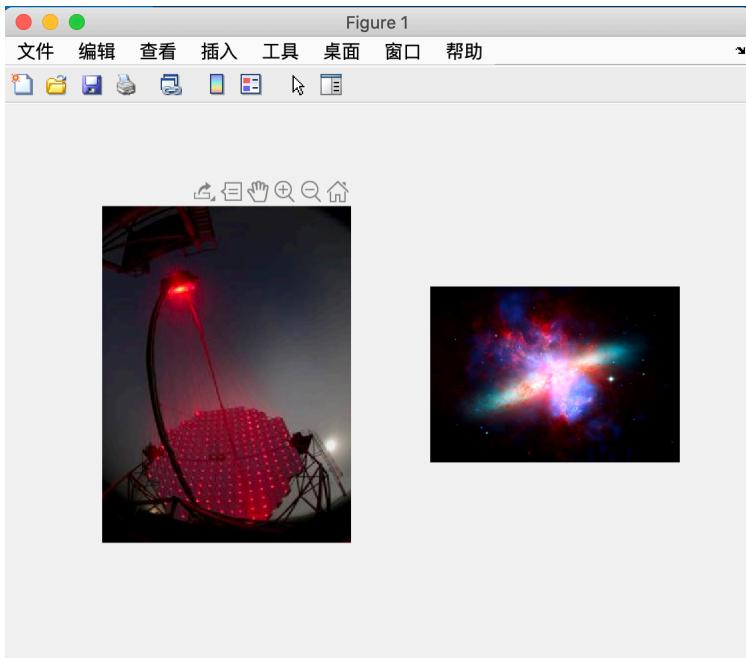


1.

```
im1 = imread('images/image1.png');
im2 = imread('images/image2.png');
figure();
subplot(1, 2, 1); imshow(im1);
subplot(1, 2, 2); imshow(im2);
```



2.

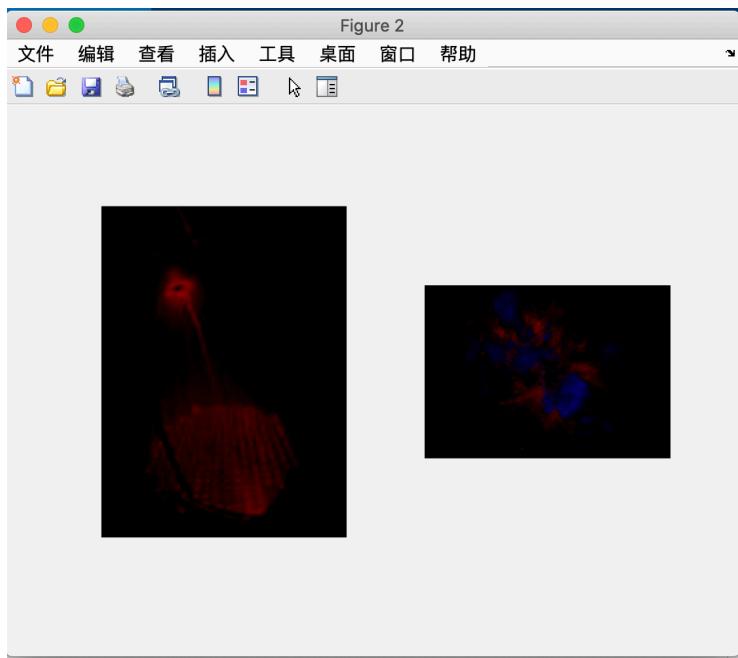
```
Gauss_filter = fspecial('gaussian',5)/255;
```

3.

```
im1ycbcr = rgb2ycbcr(im1);
im2ycbcr = rgb2ycbcr(im2);
y1 = im1ycbcr(:, :, 1);
y2 = im2ycbcr(:, :, 1);
```

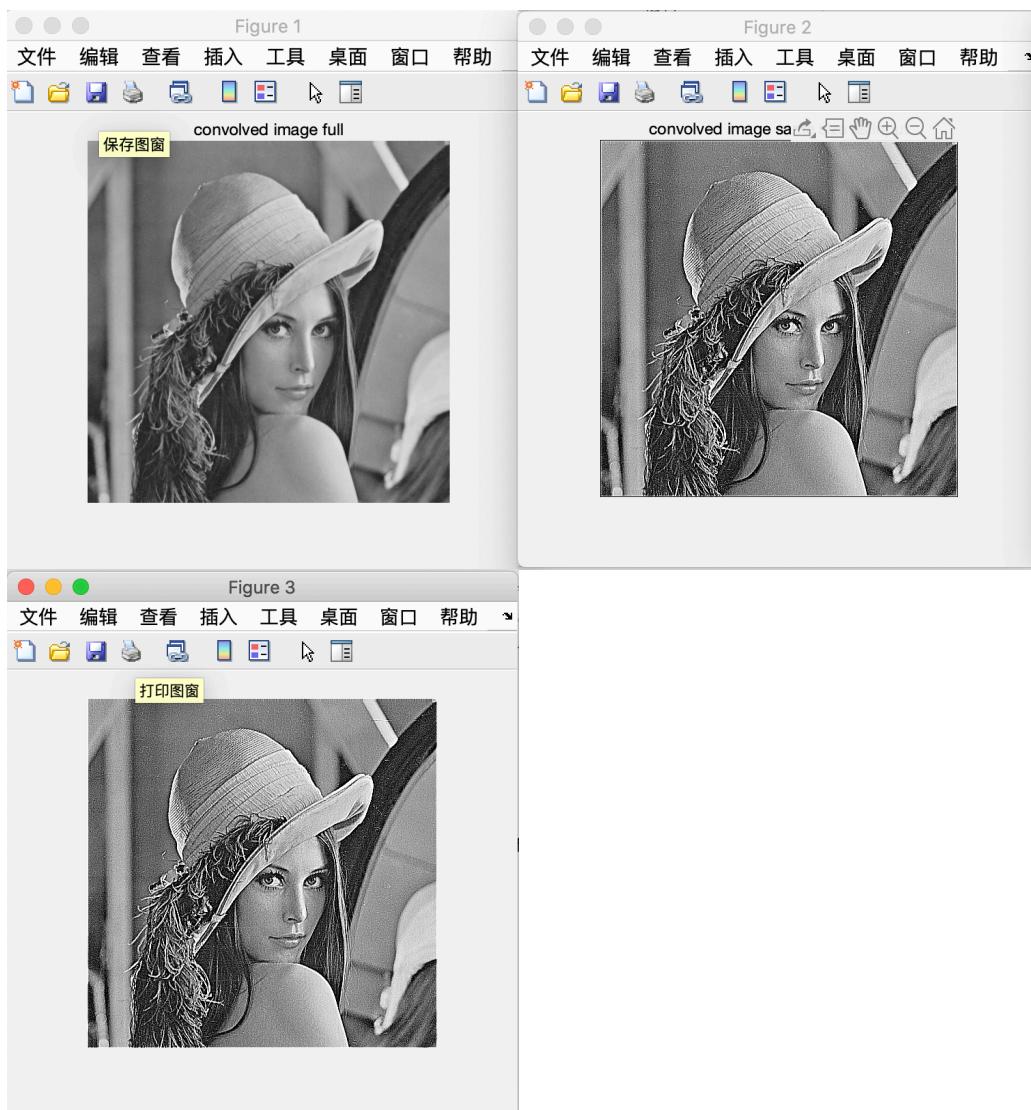
4.

```
fY1 = filter2(Gauss_filter, y1);
fY2 = filter2(Gauss_filter, y2);
im1ycbcr(:, :, 1) = fY1;
im2ycbcr(:, :, 1) = fY2;
im1f = ycbcr2rgb(im1ycbcr);
im2f = ycbcr2rgb(im2ycbcr);
figure();
subplot(1, 2, 1); imshow(im1f);
subplot(1, 2, 2); imshow(im2f);
```



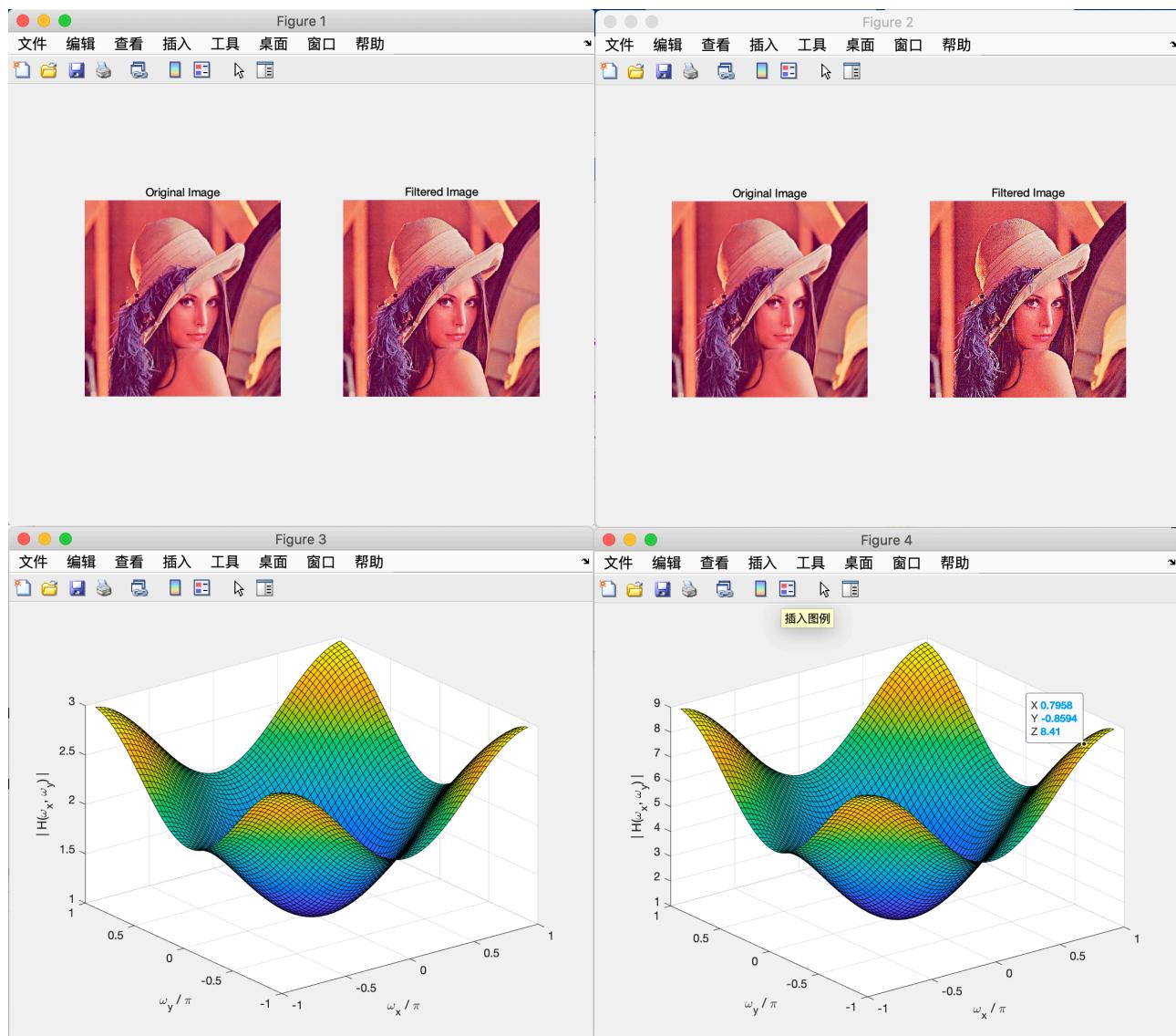
5.

EquivalenceOfSpatialAndFrequencyDomainConv.m

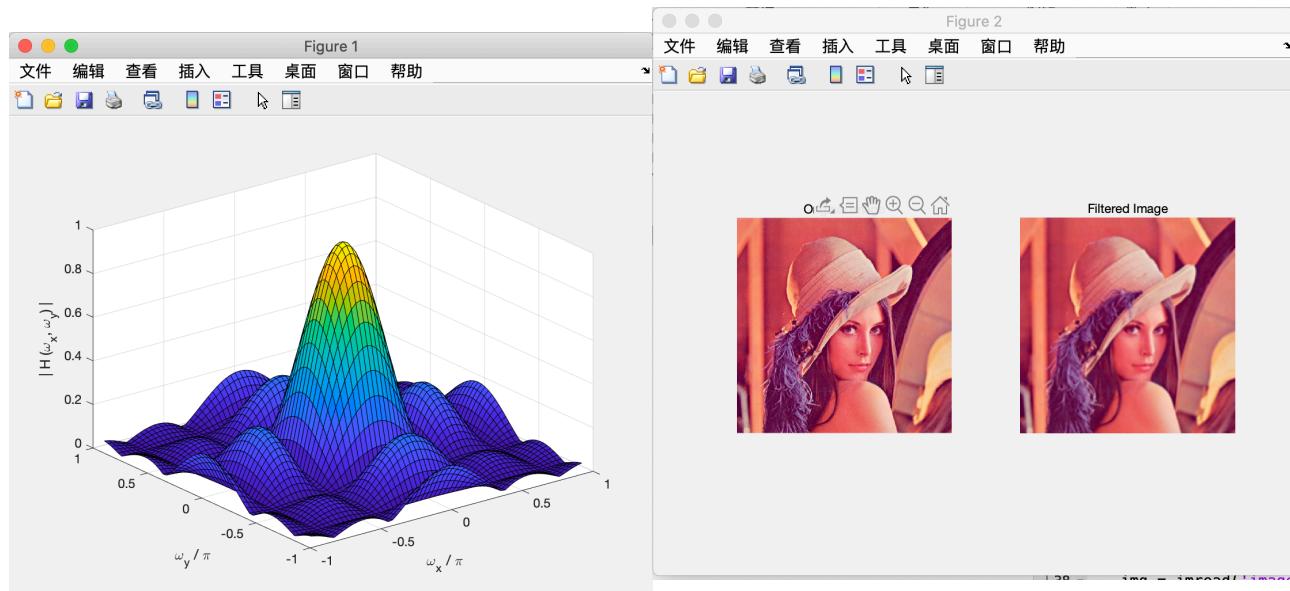


6.

Sharpening_Filter.m

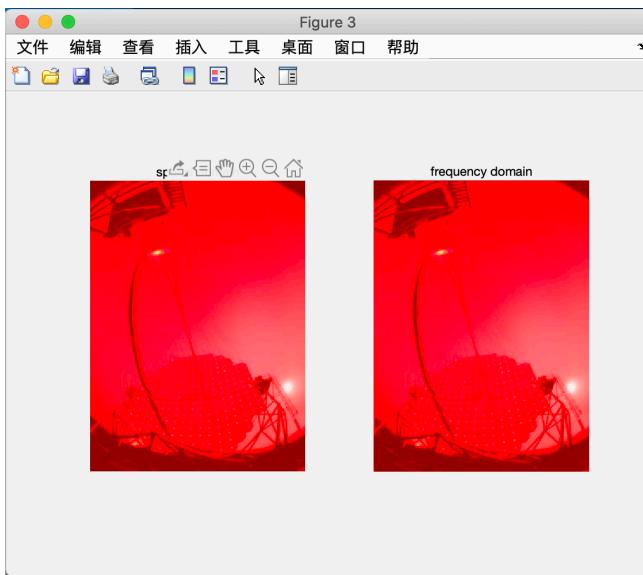


Lowpass_filter.m



7.

```
im1ycbcr = rgb2ycbcr(im1);
Y = im1ycbcr(:,:,1);
FY = fft2(Y);
h = [0 1 0;1 4 1;0 1 0];
Ik = uint8(conv2(Y,h,'same'));
h1 = zeros(338,250);
h1(1:3,1:3) = h;
H = fft2(h1);
Iff = FY .* H;
Iffi = uint8(ifft2(Iff));
spatial = im1;
frequency = im1;
spatial(:,:,1) = Ik;
frequency(:,:,1) = Iffi;
figure();
subplot(1,2,1);imshow(spatial);title('spatial domain');
subplot(1,2,2);imshow(frequency);title('frequency domain');
```



8.

```
im = imread('images/Lena.png');
im1g = rgb2gray(im); [r, c] = size(im1g); ft = dct2(im1g);
ift = idct2(ft);
nx = round(1.75 * r);
ny = round(1.75 * c);
ft_upsampled = dct2(im1g)*1.75; ift_upsampled = idct2(ft_upsampled, nx, ny);
nx = round(0.6 * r);
ny = round(0.6 * c);
ft_dsmpled = dct2(im1g)*0.6;
ift_dsmpled = idct2(ft_dsmpled, nx, ny);
subplot(1,3,1);imshow(im1g);title('original');
```

```
subplot(1,3,2);imshow(ift_upsampled,[0 ,255]);title('factors 1.75');
subplot(1,3,3);imshow(ift_dsampled,[0 ,255]);title('factors 0.6');
```

