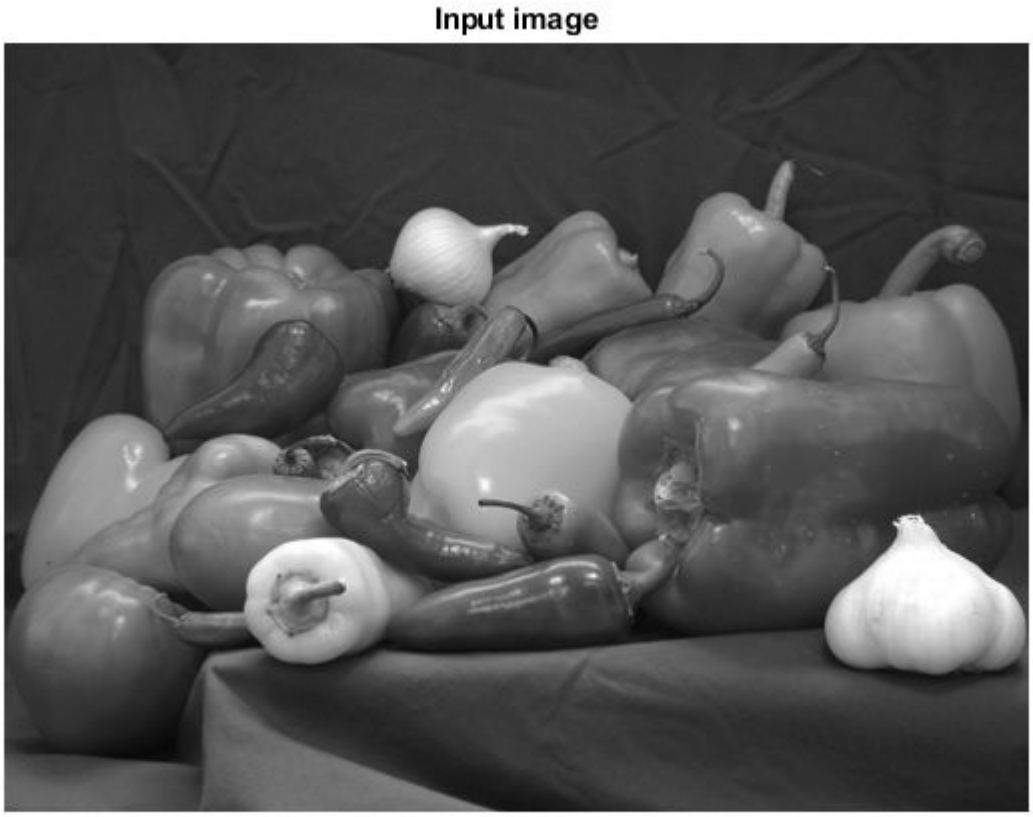
LAB 3 - Convolution and Filtering

Nick Snyder

Table of Contents

Part 1: Viewing Kernels with the Kernel Browser

```
Ii = rgb2gray(imread('peppers.png'));
imshow(Ii), title('Input image')
```



Part 2: Lowpass filtering and Neighborhood Averaging

1.

```
Kernel = Lowpass02;
```

2.

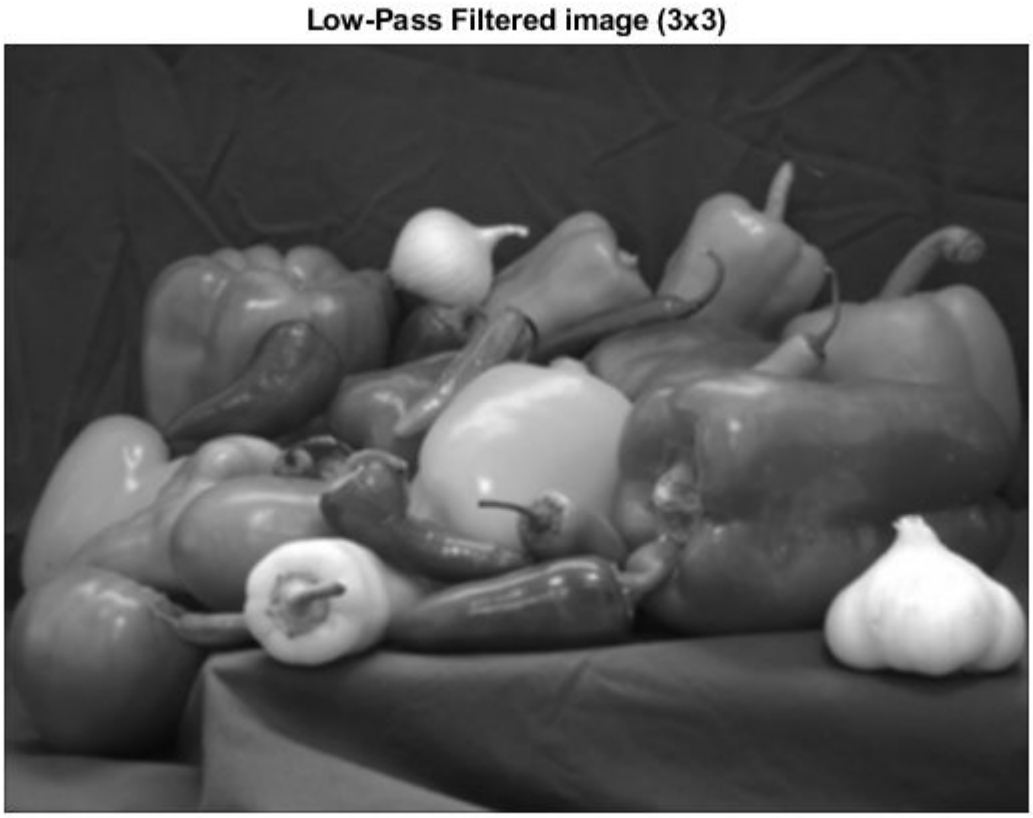
```
Lowpass02 = fspecial('average', 3);
```

3.

```
Lowpass5x5 = fspecial('average', 5);
Lowpass7x7 = fspecial('average', 7);
```

4.

```
Io = imfilter(Ii, Kernel, 'conv');
imshow(Io), title('Low-Pass Filtered image (3x3)')
```

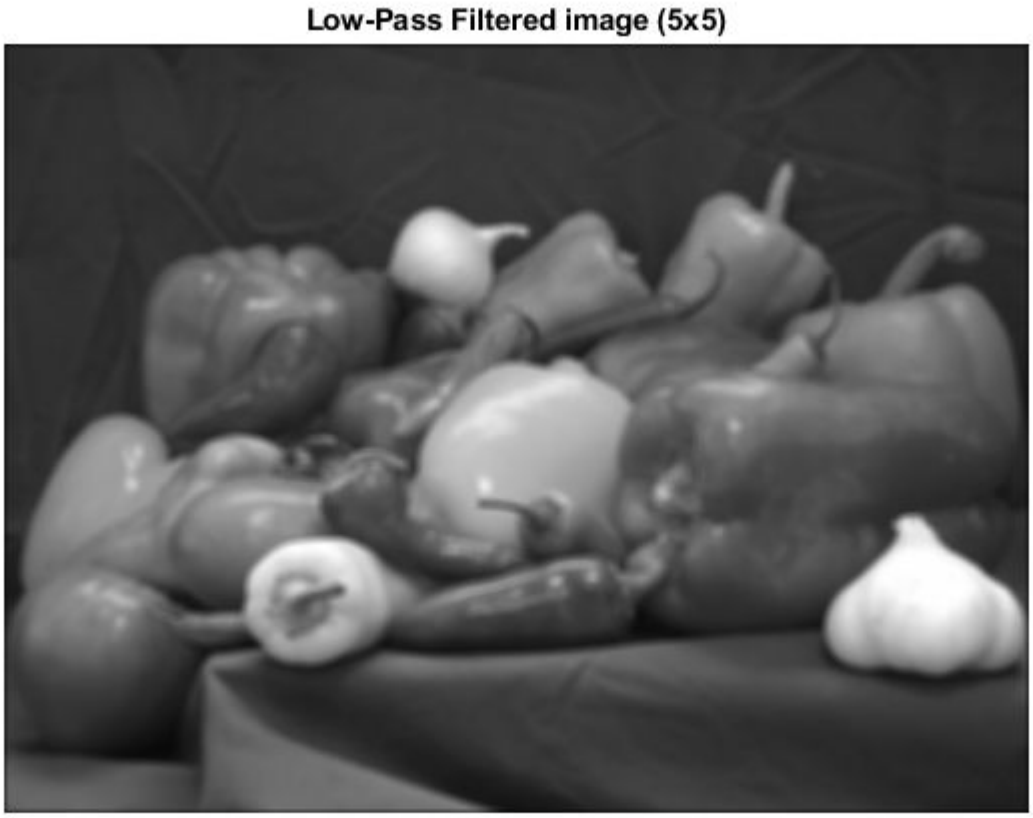


The image appears to be blurred a little bit

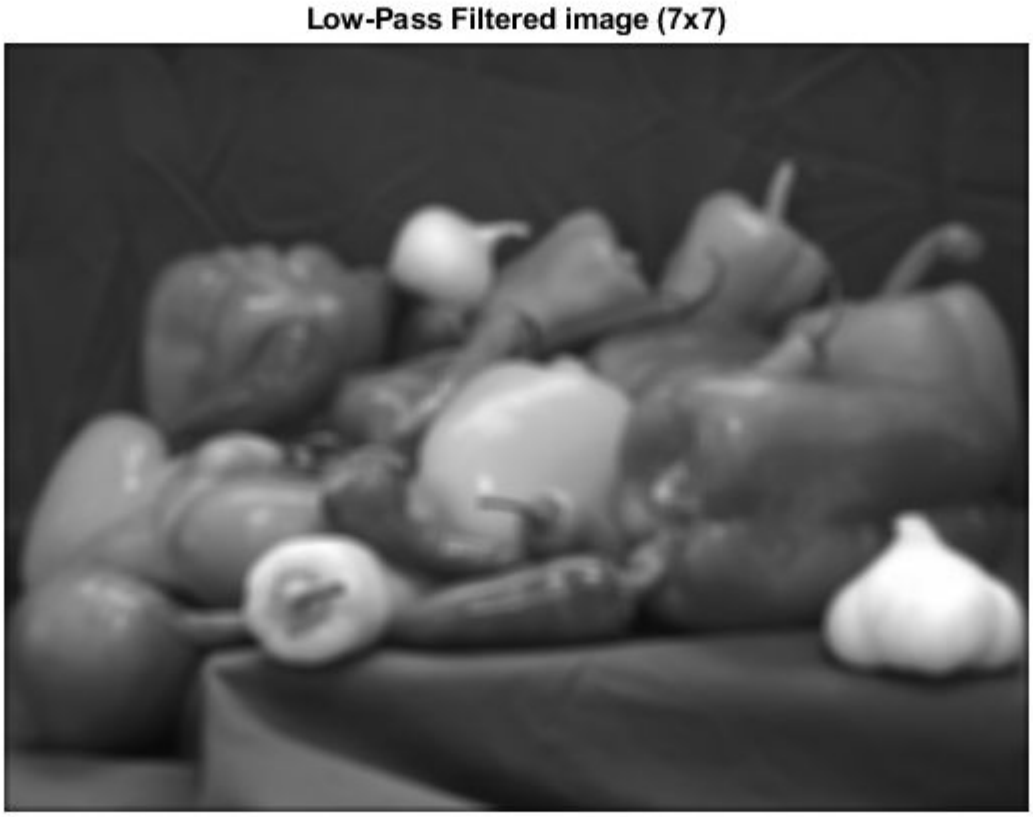
5.

```
Kernel = Lowpass5x5;
Io = imfilter(Ii, Kernel, 'conv');
```

```
imshow(Io), title('Low-Pass Filtered image (5x5)')
```



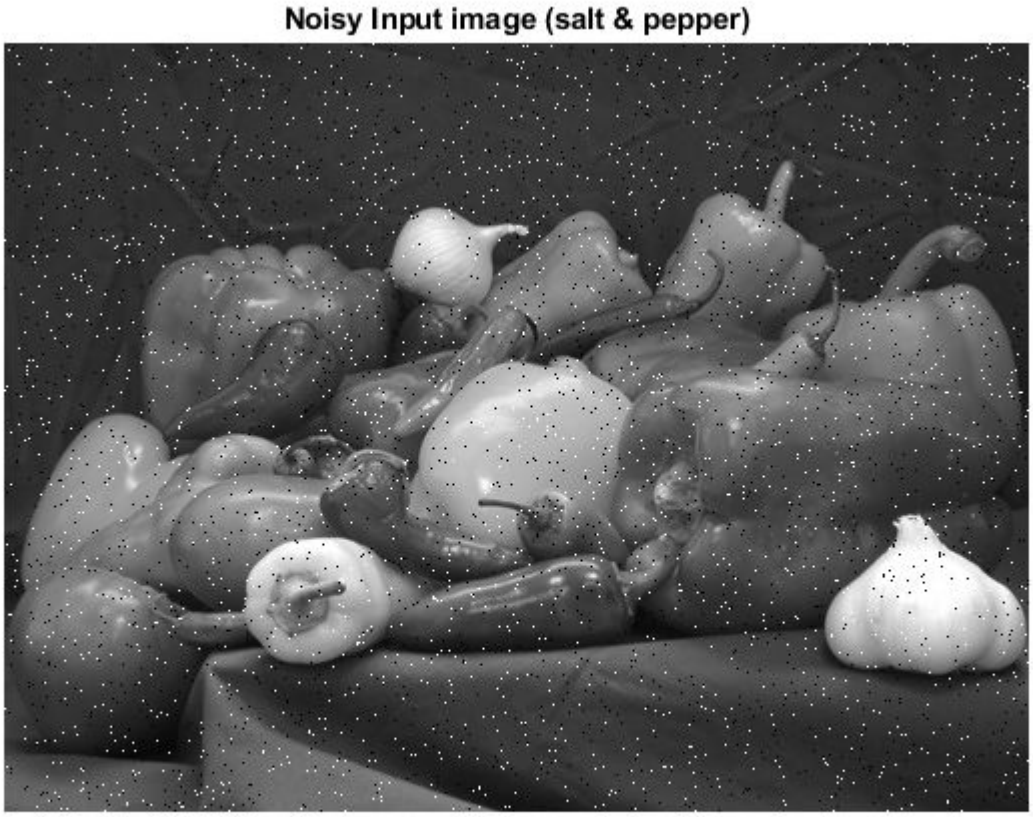

```
Kernel = Lowpass7x7;
Io = imfilter(Ii, Kernel, 'conv');
imshow(Io), title('Low-Pass Filtered image (7x7)')
```



A larger kernel seems to blur the image more. The background looks largely unaffected.

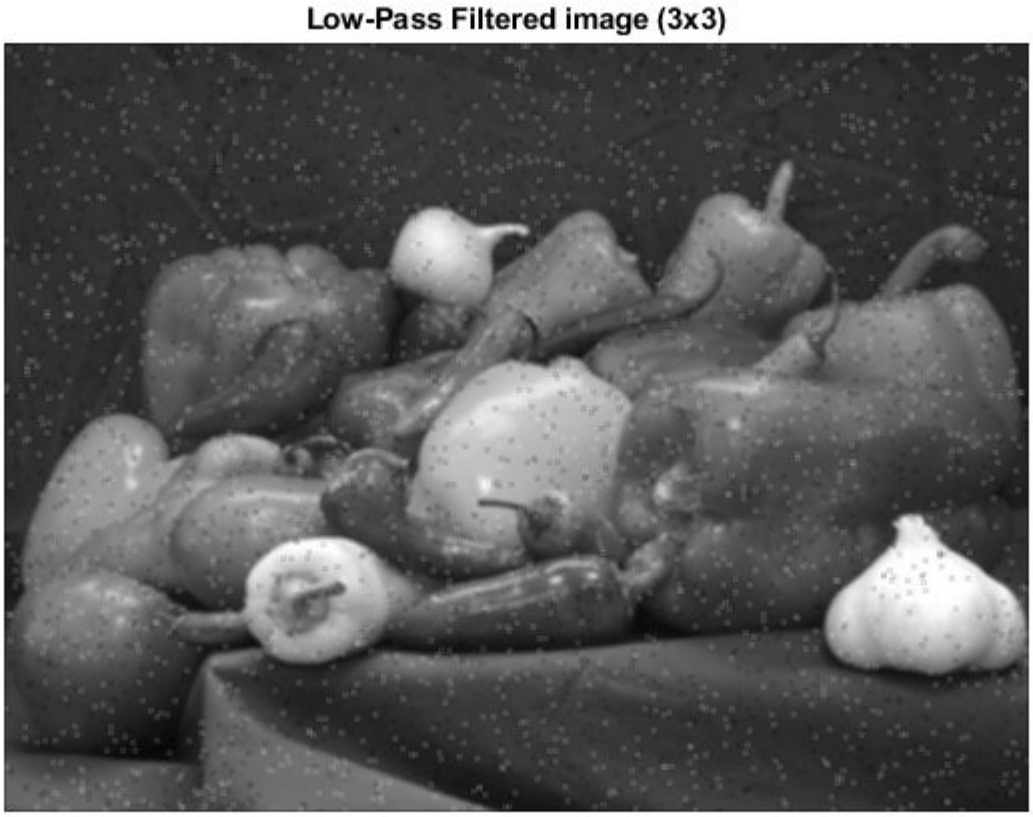
6.

```
Iin = imnoise(Ii, 'salt & pepper', 0.02);
imshow(Iin), title('Noisy Input image (salt & pepper)')
```

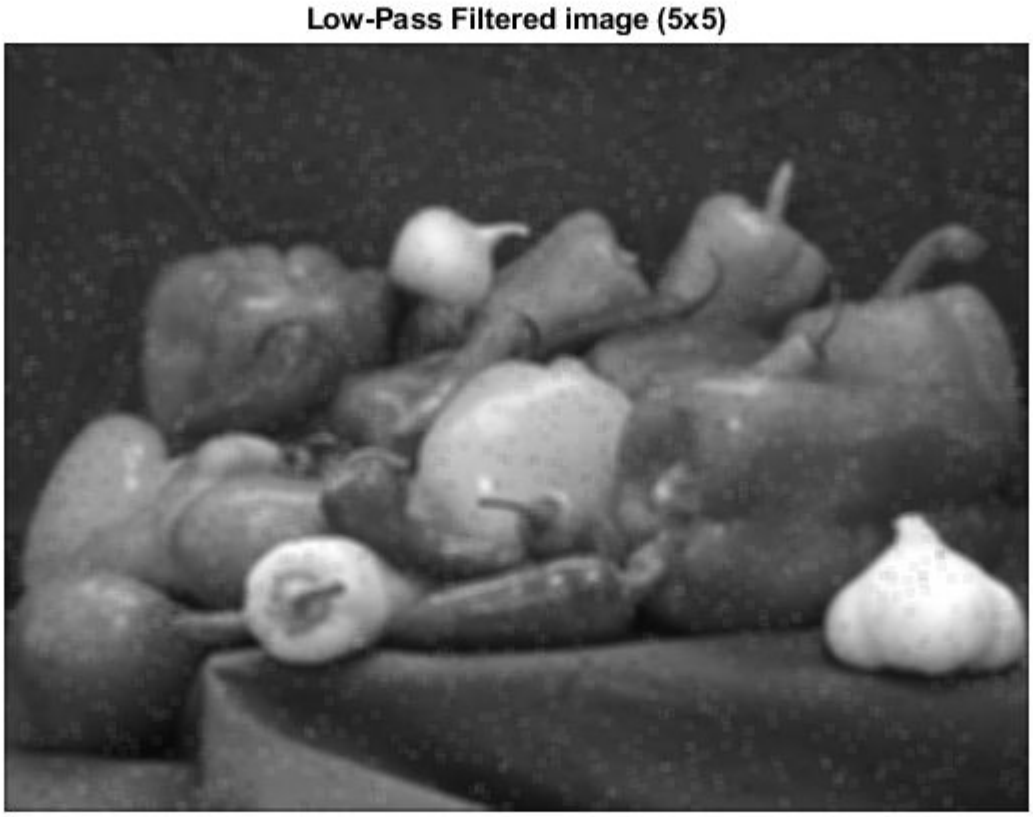


7.

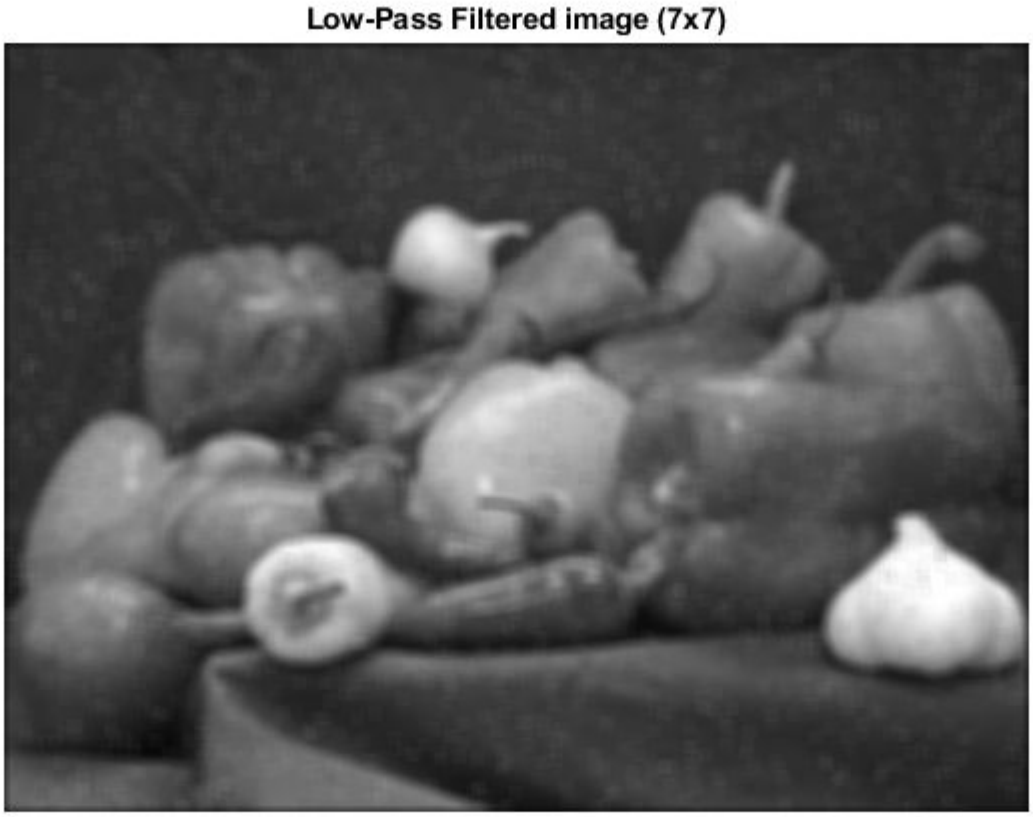
```
Kernel = Lowpass02;
Io = imfilter(Iin, Kernel, 'conv');
imshow(Io), title('Low-Pass Filtered image (3x3)')
```




```
Kernel = Lowpass5x5;
Io = imfilter(Iin, Kernel, 'conv');
imshow(Io), title('Low-Pass Filtered image (5x5)')
```



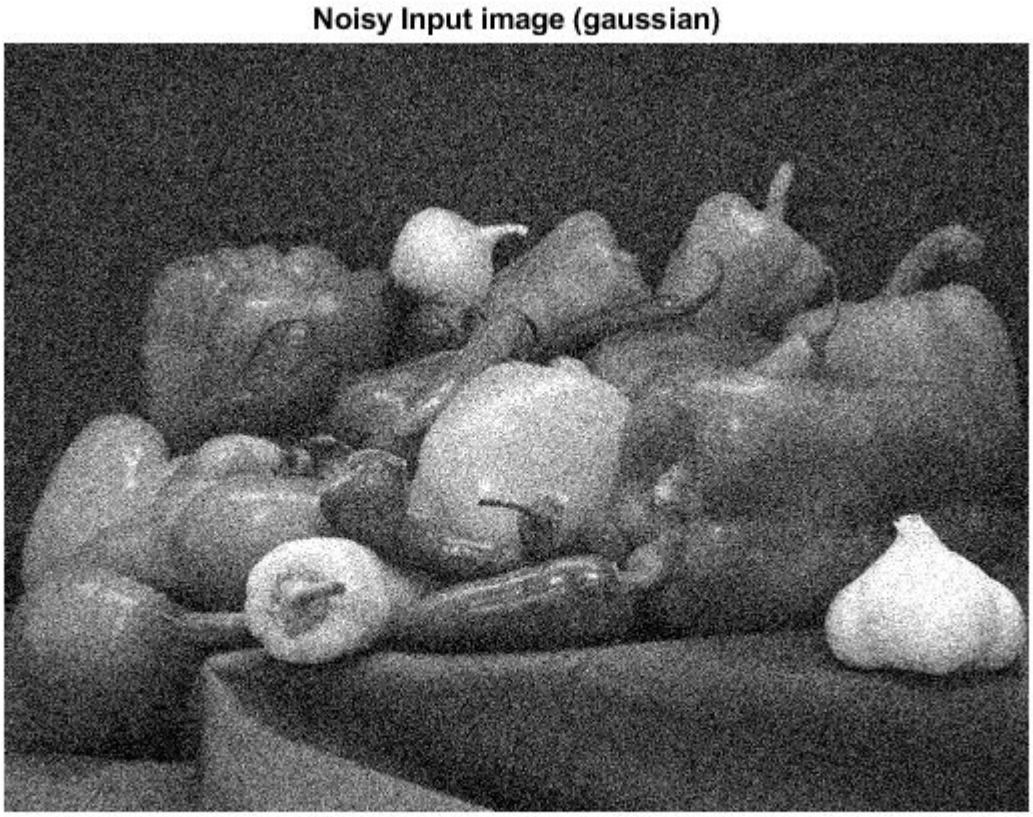

```
Kernel = Lowpass7x7;
Io = imfilter(Iin, Kernel, 'conv');
imshow(Io), title('Low-Pass Filtered image (7x7)')
```



The averaging blurrs both the image and the noise while the noise is still visible.

8.

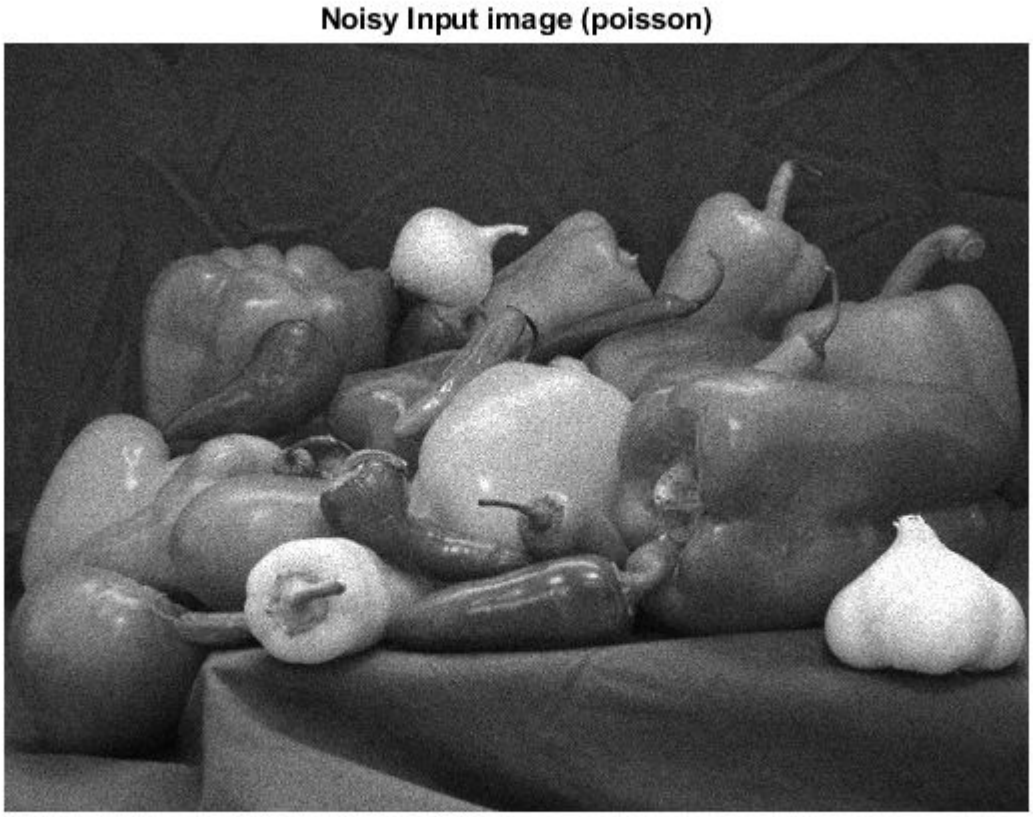
```
Kernel = Lowpass02;
Iin = imnoise(Ii, 'gaussian', 0.02);
Io = imfilter(Iin, Kernel, 'conv');
imshow(Iin), title('Noisy Input image (gaussian)')
```




```
imshow(Io), title('Low-Pass Filtered image (3x3)')
```

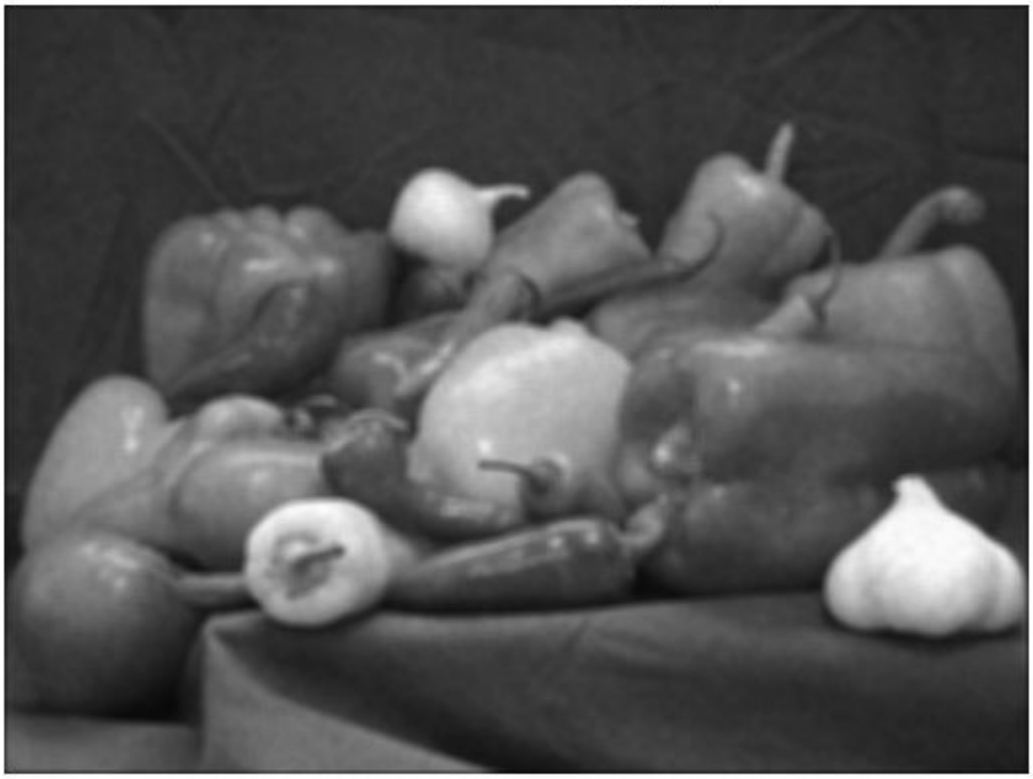
Low-Pass Filtered image (3x3)

```
Kernel = Lowpass5x5;
Iin = imnoise(Ii, 'poisson');
Io = imfilter(Iin, Kernel, 'conv');
imshow(Iin), title('Noisy Input image (poisson)')
```

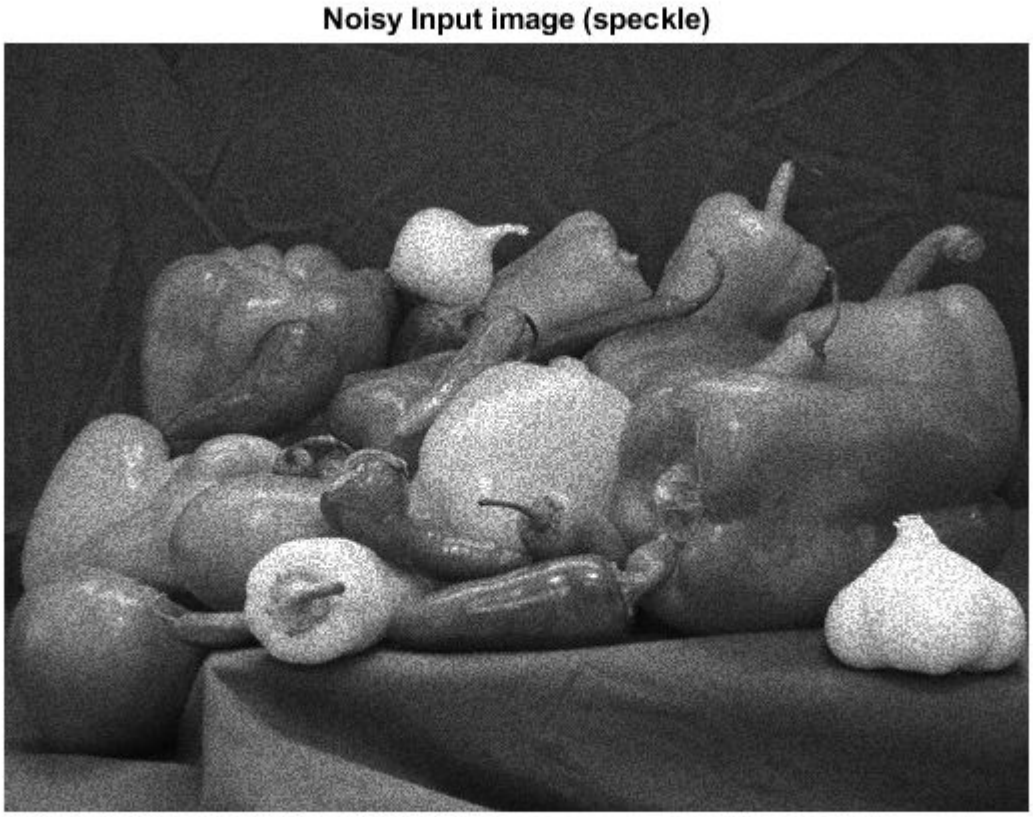



```
imshow(Io), title('Low-Pass Filtered image (5x5)')
```

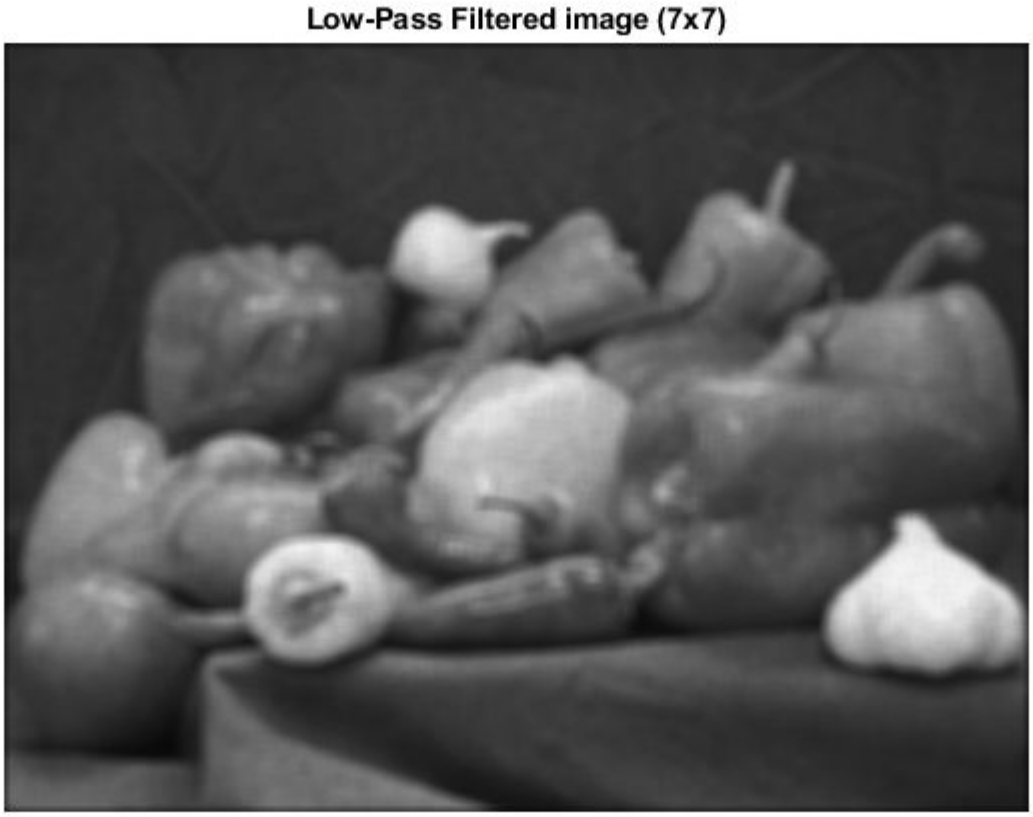
Low-Pass Filtered image (5x5)



```
Kernel = Lowpass7x7;
Iin = imnoise(Ii, 'speckle', 0.02);
Io = imfilter(Iin, Kernel, 'conv');
imshow(Iin), title('Noisy Input image (speckle)')
```




```
imshow(Io), title('Low-Pass Filtered image (7x7)')
```

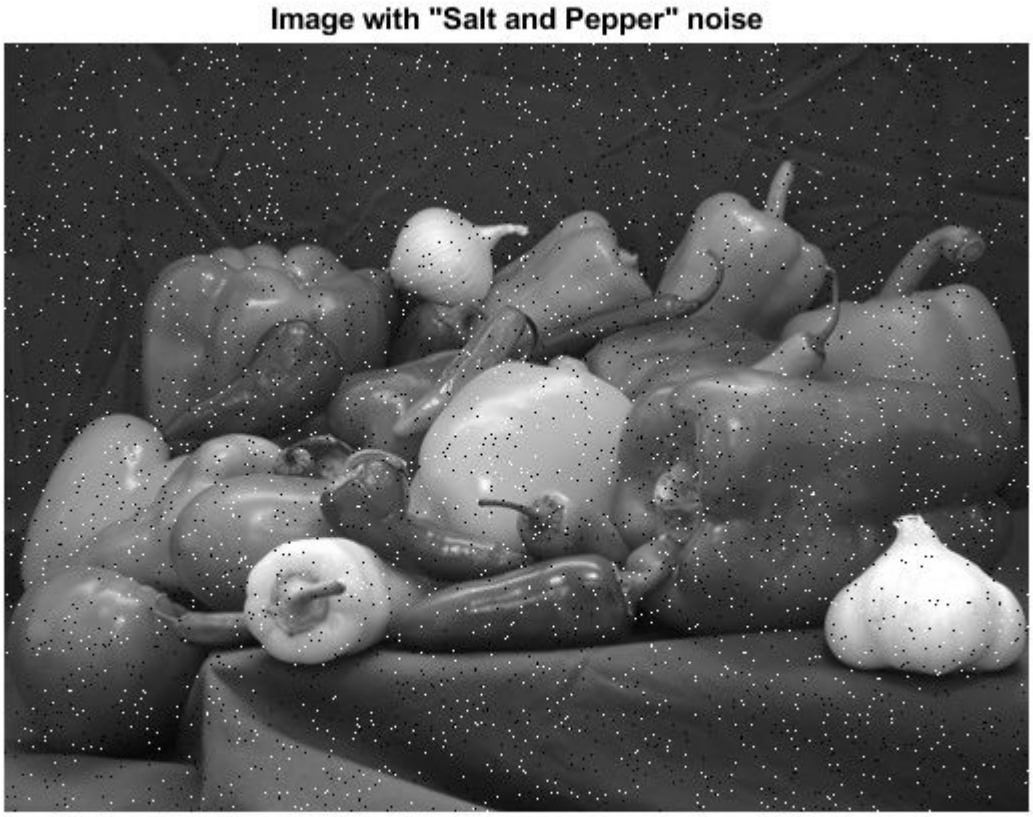


The other noise functions react similarly to the filter kernel

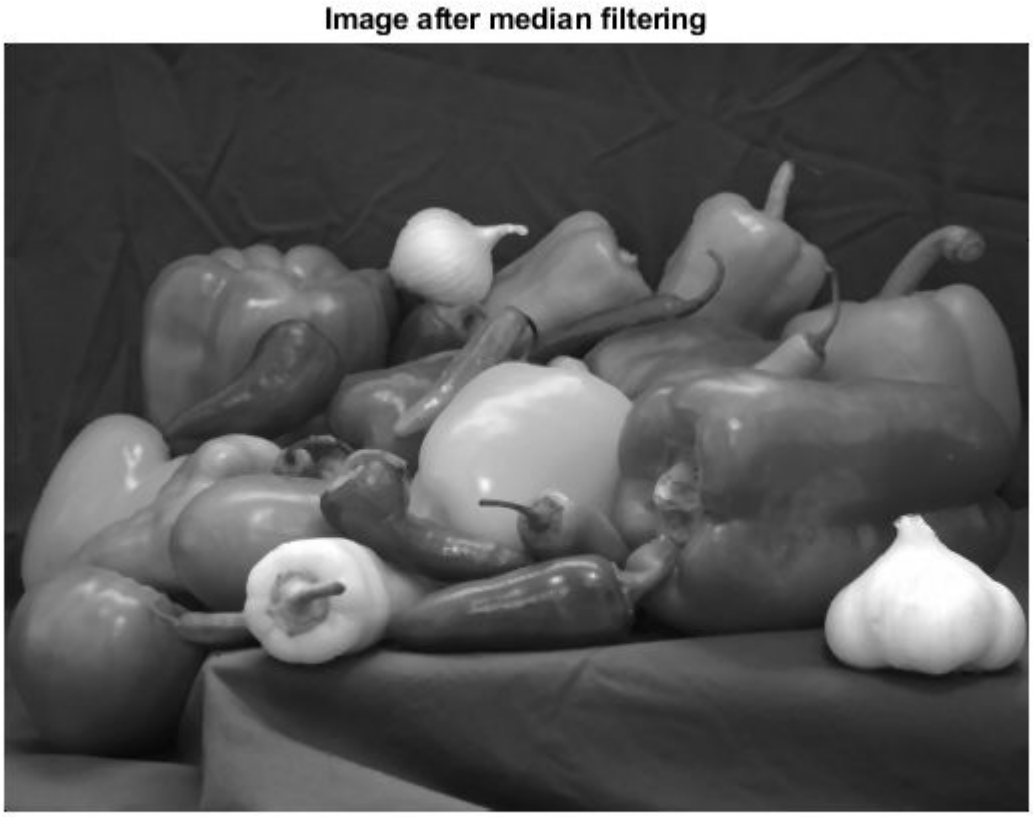
Part 3: Noise Removal using a Median Filter

2.

```
Iin = imnoise(Ii, 'salt & pepper', 0.02);
imshow(Iin), title('Image with "Salt and Pepper" noise')
```



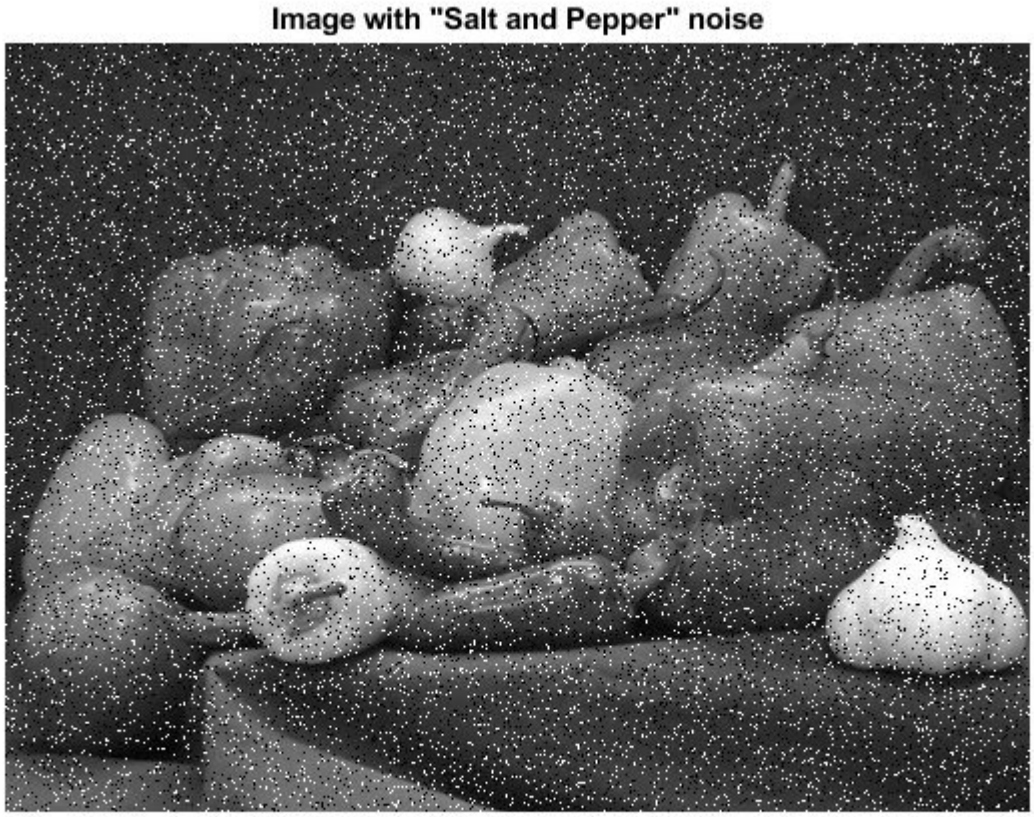

```
Io = medfilt2(Iin, [3 3]);
imshow(Io), title('Image after median filtering')
```



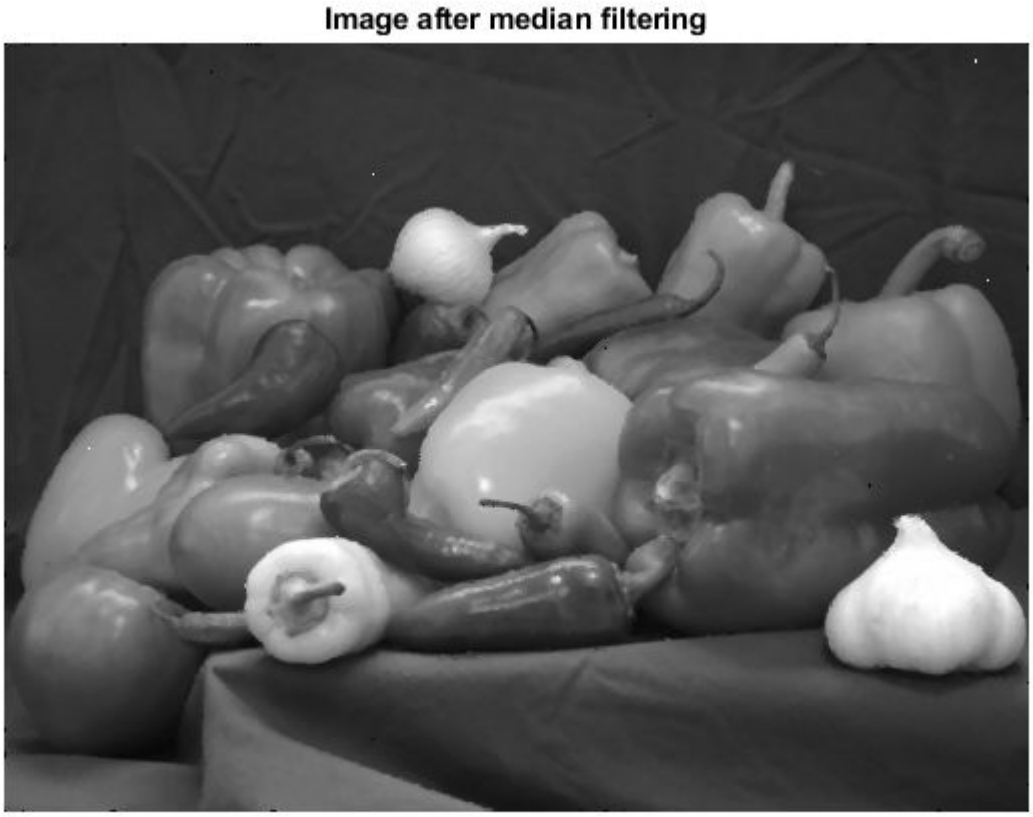
The median filter works very well and removes all the noise. It is much better than a lowpass filter. There is a slight side-effect of the median filter which distorts the edges of some objects. This could be the filter interpreting the edges as noise and trying to filter it.

3.

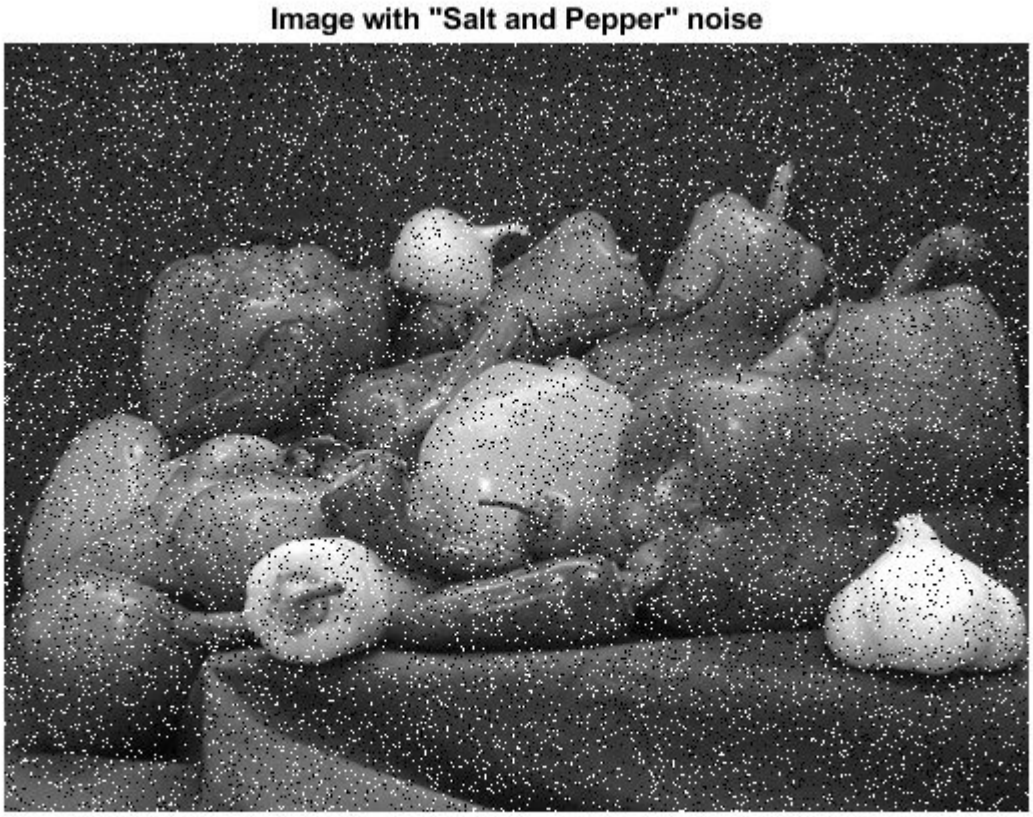
```
Iin = imnoise(Ii, 'salt & pepper', 0.1);
imshow(Iin), title('Image with "Salt and Pepper" noise')
```



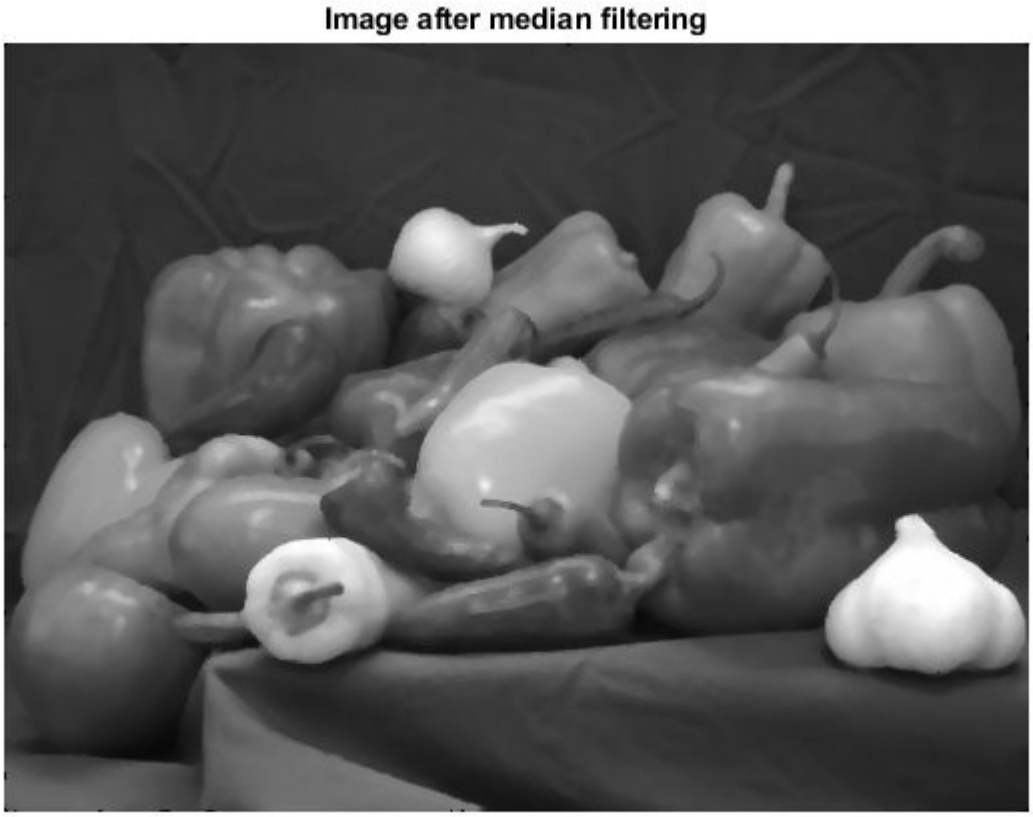

```
Io = medfilt2(Iin, [3 3]);
imshow(Io), title('Image after median filtering')
```



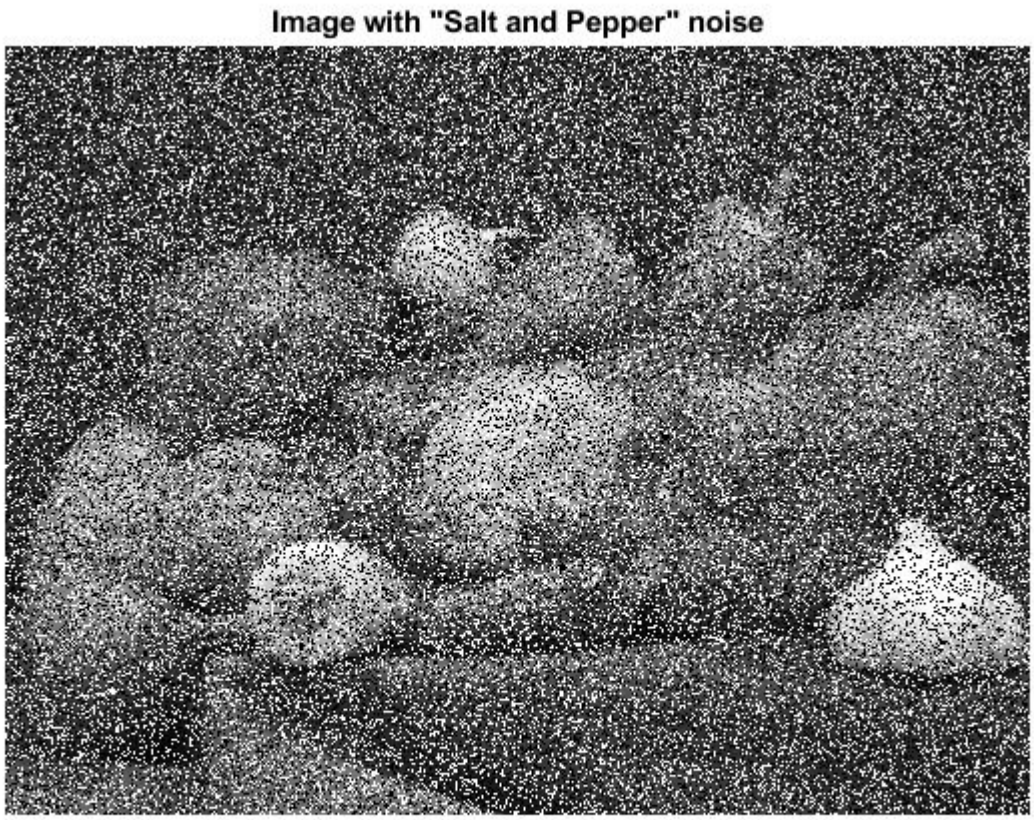

```
Iin = imnoise(Ii, 'salt & pepper', 0.1);
imshow(Iin), title('Image with "Salt and Pepper" noise')
```



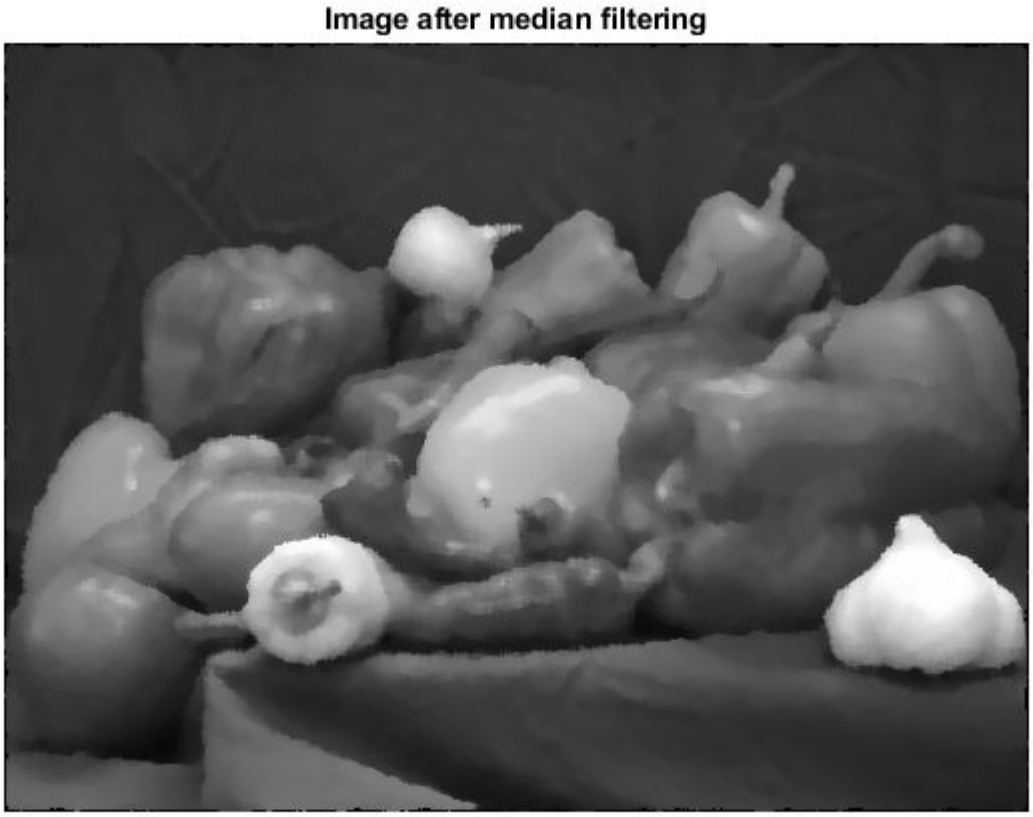

```
Io = medfilt2(Iin, [5 5]);
imshow(Io), title('Image after median filtering')
```




```
Iin = imnoise(Ii, 'salt & pepper', 0.4);
imshow(Iin), title('Image with "Salt and Pepper" noise')
```




```
Io = medfilt2(Iin, [7 7]);
imshow(Io), title('Image after median filtering')
```

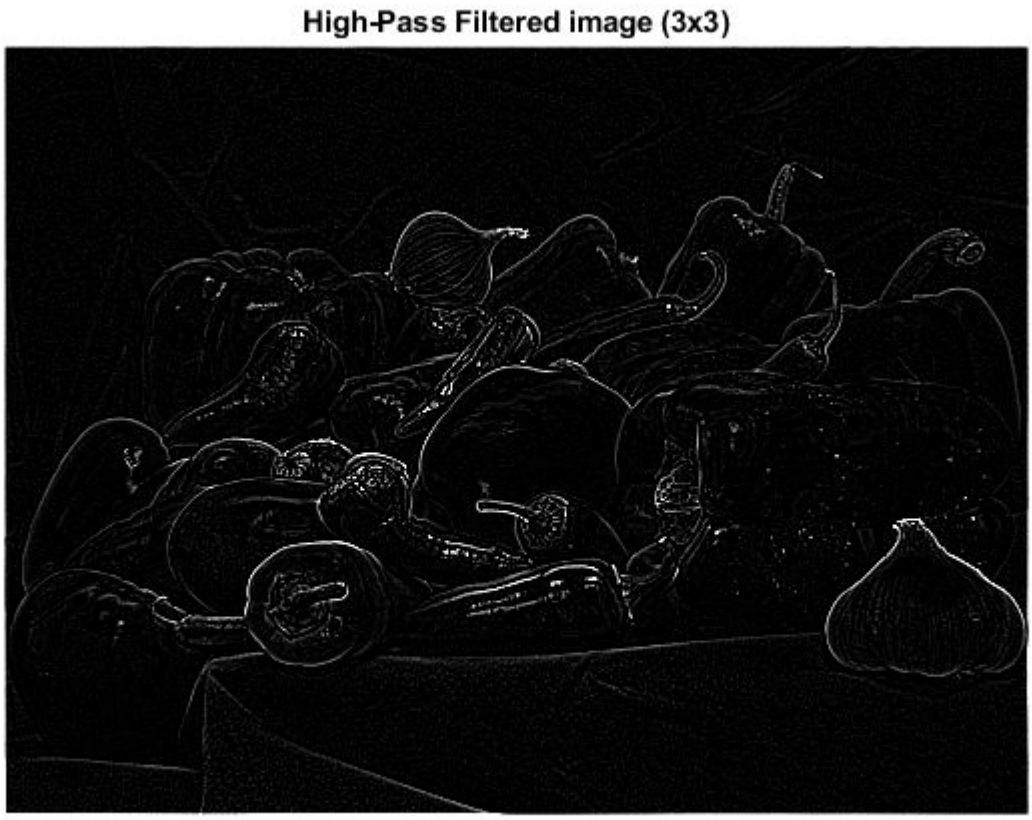


An image with too much noise cannot be completely filtered by filter that is too small. Increasing the median filter size can combat the noisiest of images but will also distort the original edges.

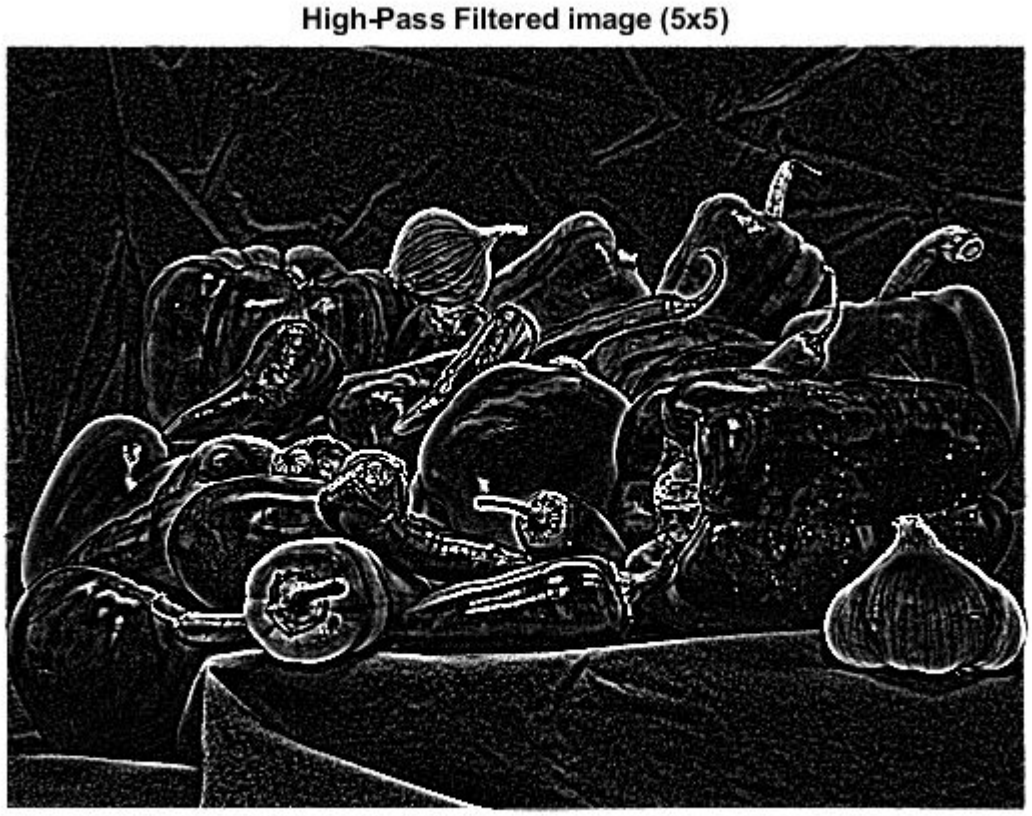
Part 4: Highpass Filtering

2.

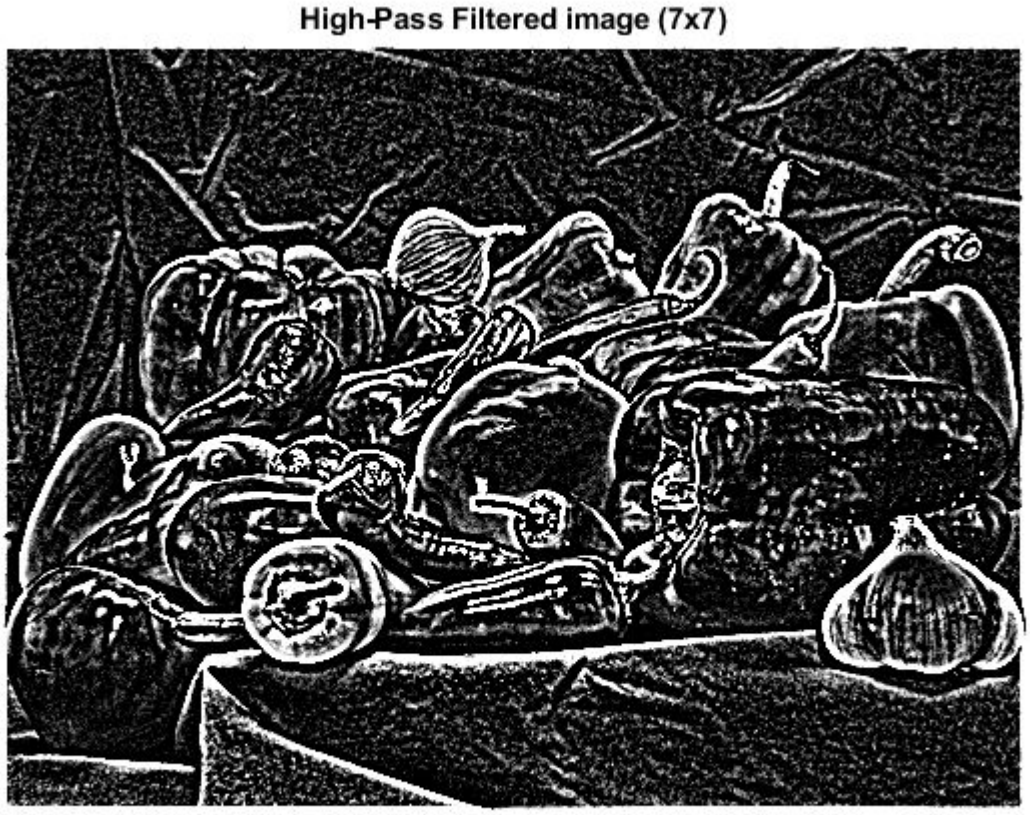
```
Highpass3x3 = [ -1, -1, -1; -1, 8, -1; -1, -1, -1 ];
Kernel = Highpass3x3;
Io = imfilter(Ii, Kernel, 'conv');
imshow(Io), title('High-Pass Filtered image (3x3)')
```




```
Highpass5x5 = [ -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1;
                -1, -1, 24, -1, -1;
                -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1 ];
Kernel = Highpass5x5;
Io = imfilter(Ii, Kernel, 'conv');
imshow(Io), title('High-Pass Filtered image (5x5)')
```




```
Highpass7x7 = [ -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, 48, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1 ];
Kernel = Highpass7x7;
Io = imfilter(Ii, Kernel, 'conv');
imshow(Io), title('High-Pass Filtered image (7x7)')
```



The high-pass filter accentuates the edges of the image, making them brighter and darkening the rest of the image. Increasing the kernel size looks to shift the cutoff frequency of the filter and accentuates softer edges.

3.

```
Laplacian01 = fspecial('laplacian');
Kernel = Laplacian01;
Io = imfilter(Ii, Kernel, 'conv');
imshow(Io), title('Laplacian Filtered image (3x3)')
```


```
S = rgb2gray(imread('cam.jpg'));
imshow(S), title('Input image')
```


```
Io = imfilter(S, Kernel, 'conv');
imshow(Io), title('Laplacian Filtered image (3x3)')
```

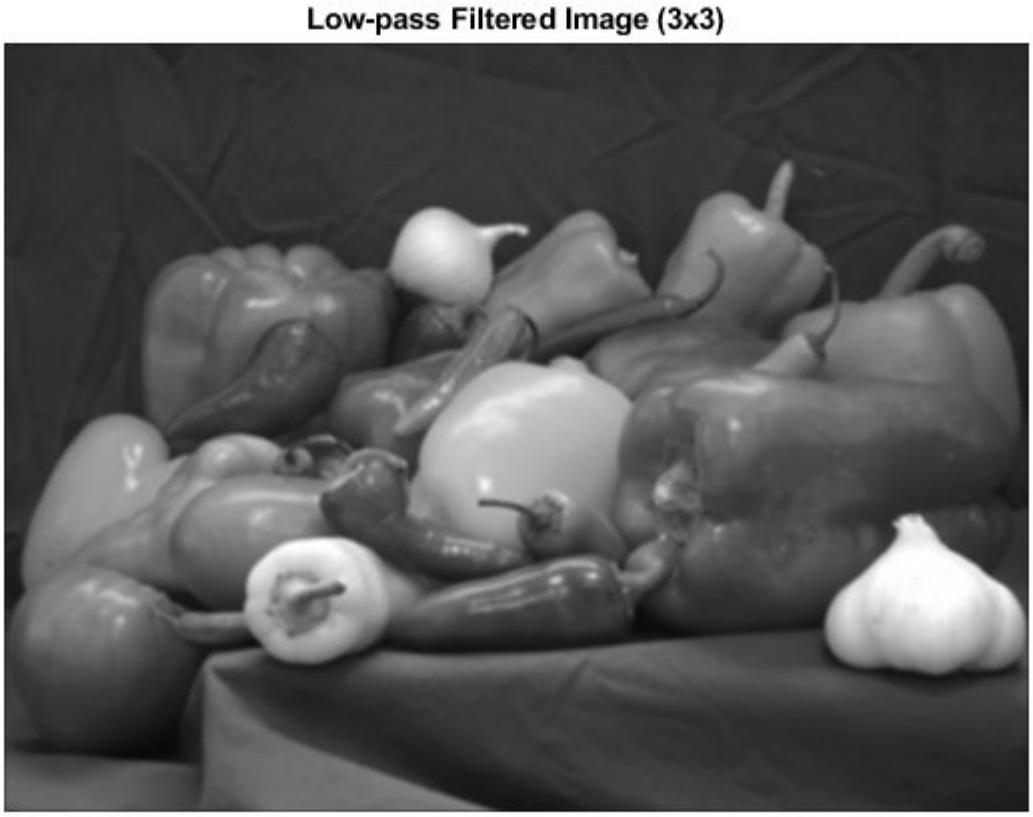
The laplacian kernel seems to be a high-pass filter with a higher cutoff frequency than normal. The image of security camera footage does not filter very well as it picks up the aliased pixels from the screen the image is taken from rather than the people in the image.

Part 5: Unsharp Masking

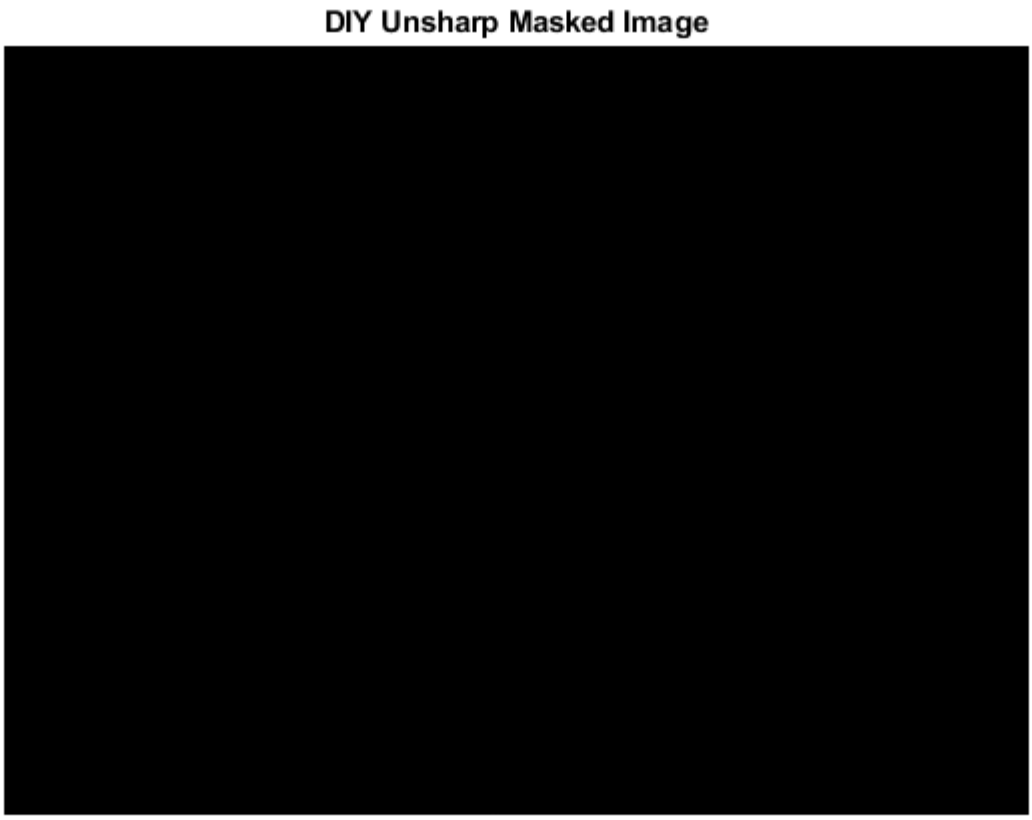
```
UnsharpKernel = fspecial('unsharp');
Kernel = UnsharpKernel;
Io = imfilter(Ii, Kernel, 'conv');
imshow(Io), title('Unsharp Masked image')
```

The edges of this image seem to be more defined while not distorting the remainder of the image. It is a solid middle-ground between high and low-pass filters.

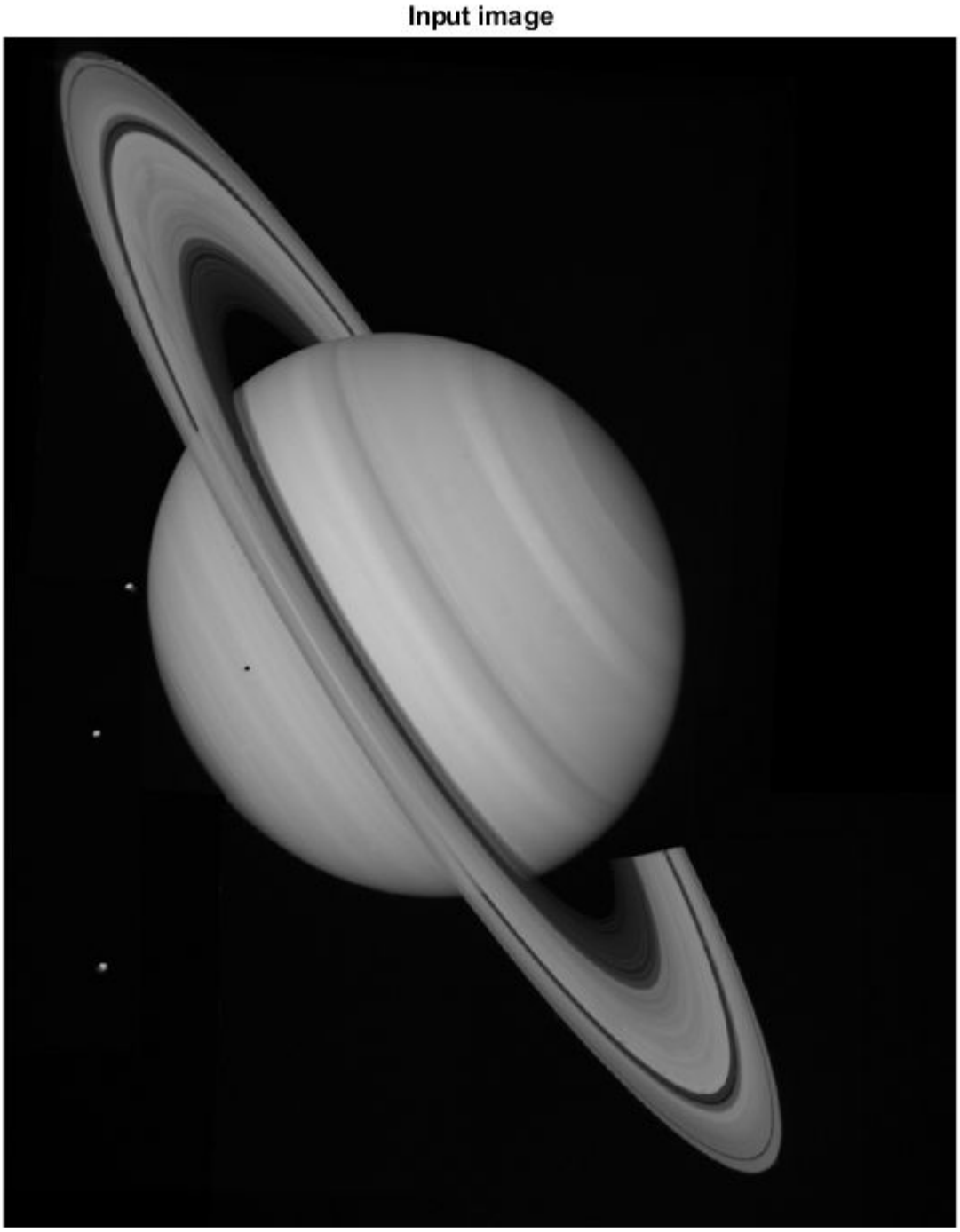
```
Kernel = Lowpass02;
low = imfilter(Ii, Kernel, 'conv');
imshow(low), title('Low-pass Filtered Image (3x3)')
```



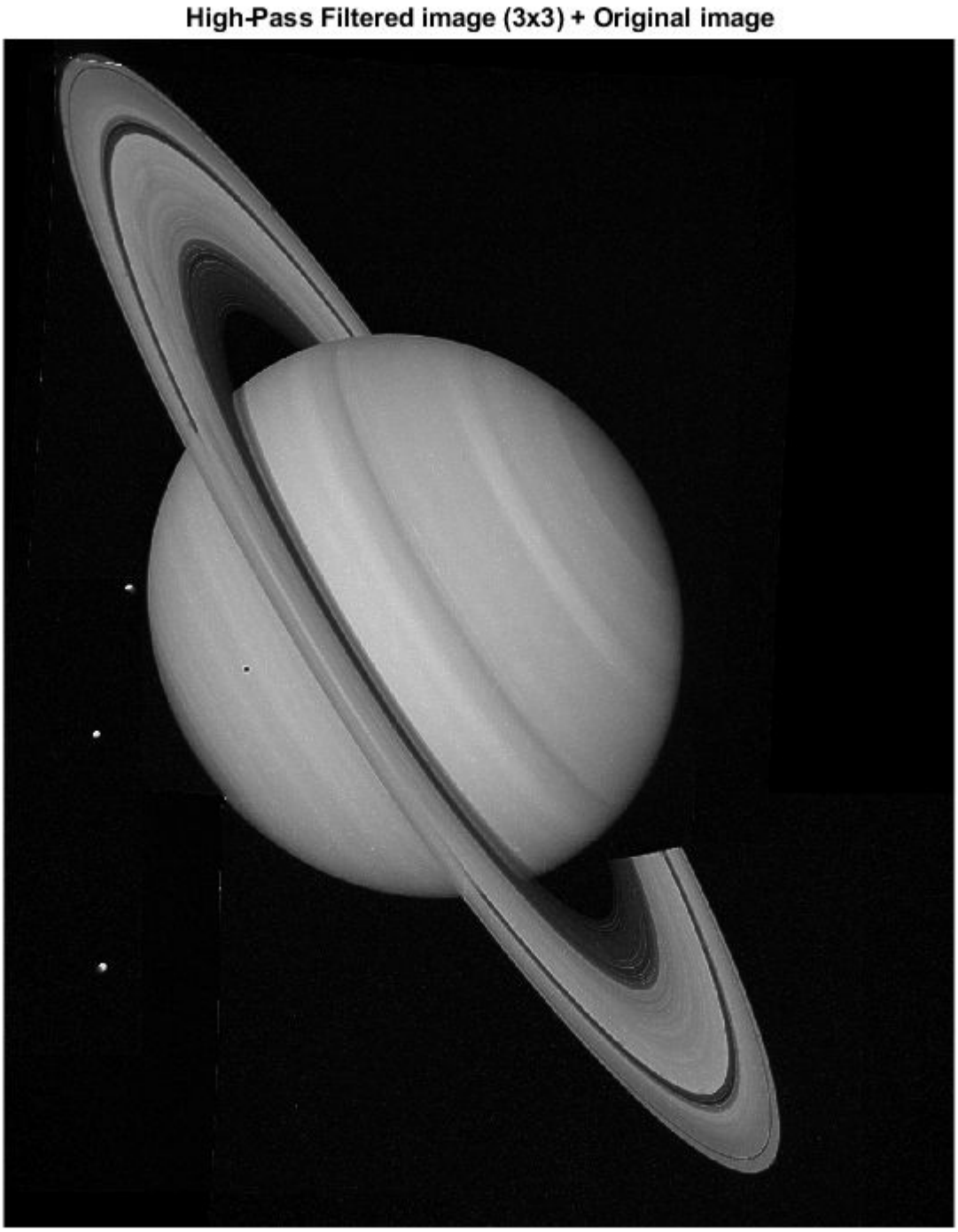

```
boost = (0.5 * Ii) - low;
imshow(boost), title('DIY Unsharp Masked Image')
```




```
Iis = rgb2gray(imread('saturn.png'));
imshow(Iis), title('Input image')
```




```
Kernel = Highpass3x3;
Io = imfilter(Iis, Kernel, 'conv');
imshow(Io + Iis), title('High-Pass Filtered image (3x3) + Original image')
```



I can't see any more moons but I can see a few rings more clearly than before. The high-pass filter found the darker rings and superimposed that on top of the original image to boost the result.

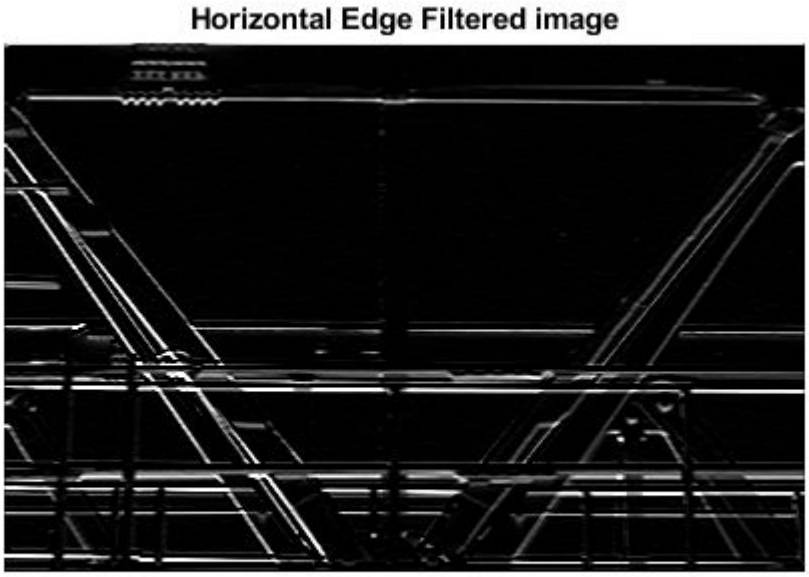
Part 6: Edge Detection

1.

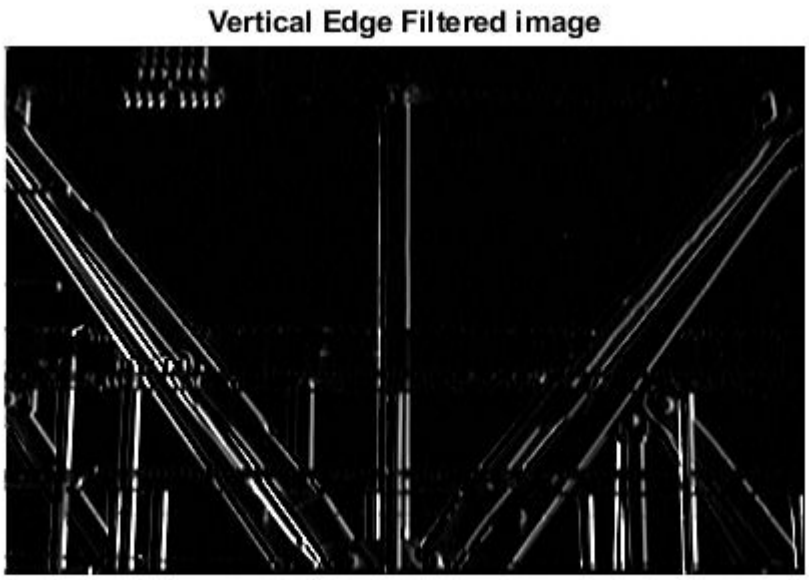
```
Iig = rgb2gray(imread('gantrycrane.png'));
```

2.

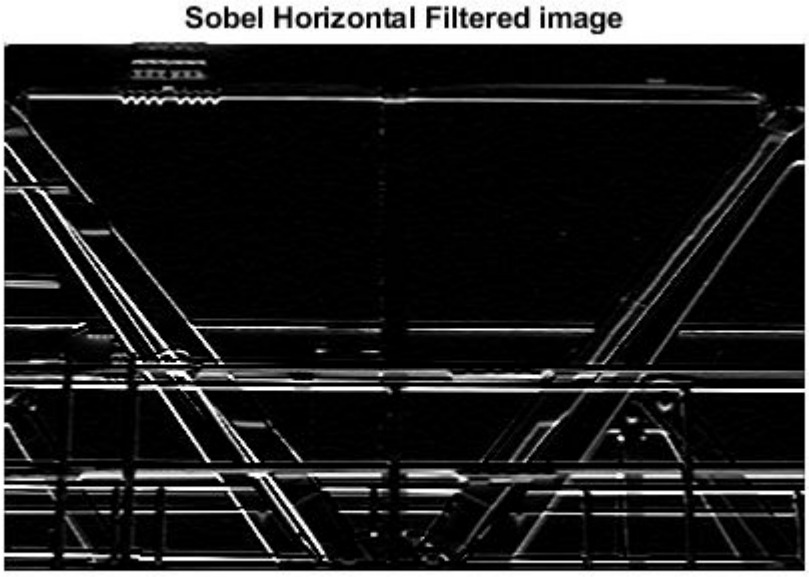
```
EdgeHorizontal01 = [ 1 1 1; 0 0 0; -1 -1 -1];
Kernel = EdgeHorizontal01;
Io = imfilter(Iig, Kernel, 'conv');
imshow(Io), title('Horizontal Edge Filtered image')
```



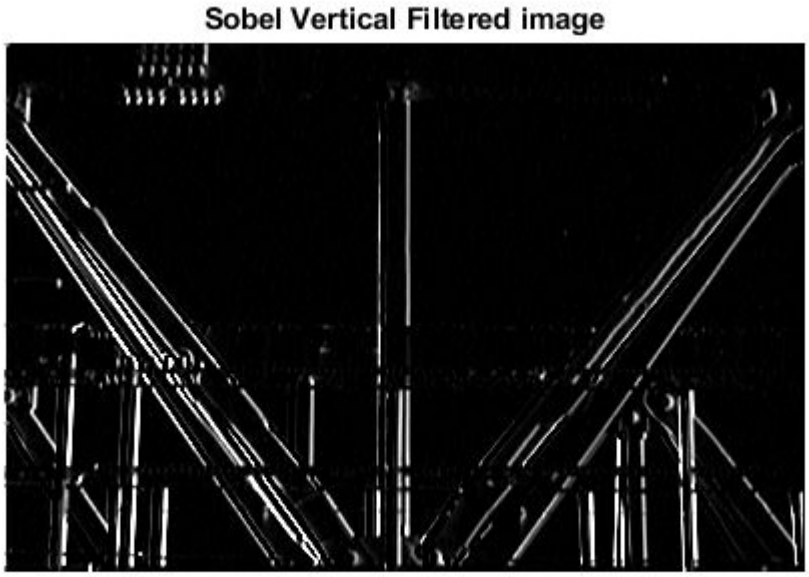

```
EdgeVertical01 = EdgeHorizontal01';
Kernel = EdgeVertical01;
Io = imfilter(Iig, Kernel, 'conv');
imshow(Io), title('Vertical Edge Filtered image')
```




```
SobelHorizontal01 = fspecial('sobel');
Kernel = SobelHorizontal01;
Io = imfilter(Iig, Kernel, 'conv');
imshow(Io), title('Sobel Horizontal Filtered image')
```




```
SobelVertical01 = SobelHorizontal01';
Kernel = SobelVertical01;
Io = imfilter(Iig, Kernel, 'conv');
imshow(Io), title('Sobel Vertical Filtered image')
```

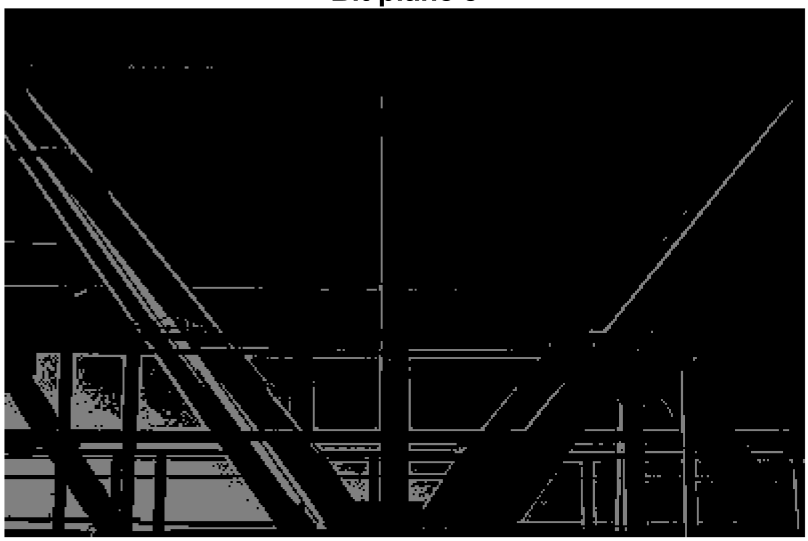


The vertical and horizontal edge filters only show verticacl and horizontal edges. The Sobel filters do the same but pick up more edges.

Part 7: Bit Planes

```
Ii = rgb2gray(imread('gantrycrane.png'));
bitplane8 = Ii - mod(Ii, 128);
bitplane7 = Ii - mod(Ii, 64) - bitplane8;
bitplane6 = Ii - mod(Ii, 32) - bitplane8 - bitplane7;
bitplane5 = Ii - mod(Ii, 16) - bitplane8 - bitplane7 - bitplane6;
bitplane4 = Ii - mod(Ii, 8) - bitplane8 - bitplane7 - bitplane6 - bitplane5;
bitplane3 = Ii - mod(Ii, 4) - bitplane8 - bitplane7 - bitplane6 - bitplane5 - bitplane4;
bitplane2 = Ii - mod(Ii, 2) - bitplane8 - bitplane7 - bitplane6 - bitplane5 - bitplane4 - bitpl
bitplane1 = Ii - mod(Ii, 1) - bitplane8 - bitplane7 - bitplane6 - bitplane5 - bitplane4 - bitpl
imshow(bitplane8), title('Bit plane 8')
```

Bit plane 8



```
imshow(bitplane7), title('Bit plane 7')
```

Bit plane 7

```
imshow(bitplane6), title('Bit plane 6')
```

Bit plane 6

```
imshow(bitplane5), title('Bit plane 5')
```

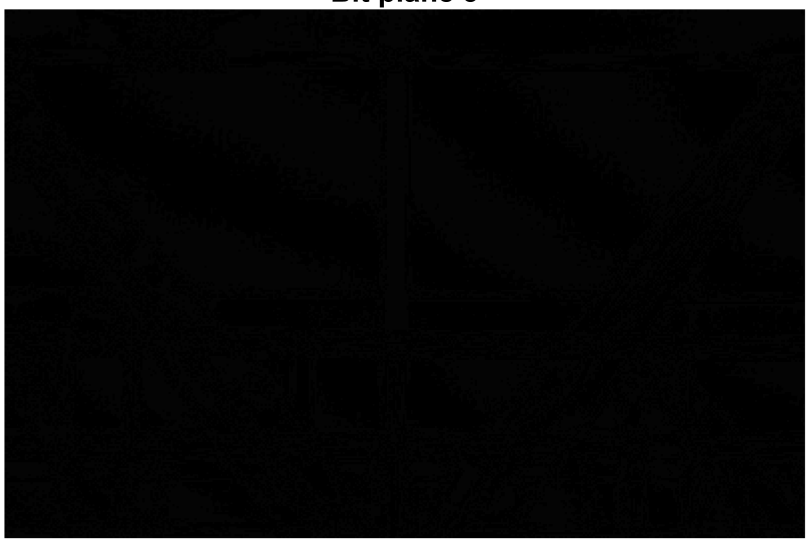
Bit plane 5

```
imshow(bitplane4), title('Bit plane 4')
```

Bit plane 4

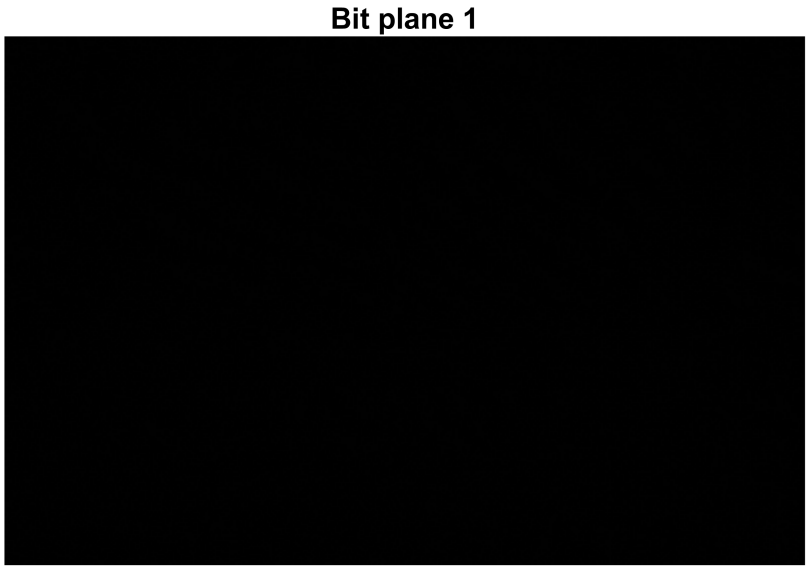
```
imshow(bitplane3), title('Bit plane 3')
```

Bit plane 3



```
imshow(bitplane2), title('Bit plane 2')
```


```
imshow(bitplane1), title('Bit plane 1')
```




```
recombine = bitplane8 + bitplane7 + bitplane6 + bitplane5 + bitplane4 + bitplane3 + bitplane2 +
a = Ii - recombine
```

```
a = 264×400 uint8 matrix
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 ···
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

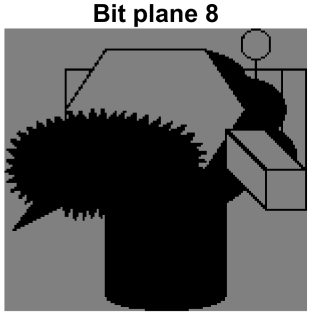
```
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  ⋮
```

```
imshow(recombine)
```

There is no difference after recombining the bit planes

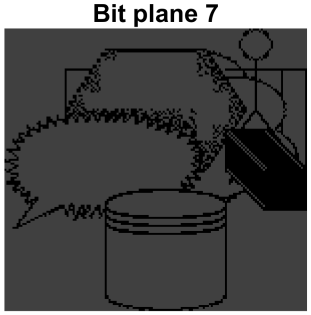
```
Ii = rgb2gray(imread('syn.jpg'));
bitplane8 = Ii - mod(Ii, 128);
bitplane7 = Ii - mod(Ii, 64) - bitplane8;
bitplane6 = Ii - mod(Ii, 32) - bitplane8 - bitplane7;
bitplane5 = Ii - mod(Ii, 16) - bitplane8 - bitplane7 - bitplane6;
bitplane4 = Ii - mod(Ii, 8) - bitplane8 - bitplane7 - bitplane6 - bitplane5;
bitplane3 = Ii - mod(Ii, 4) - bitplane8 - bitplane7 - bitplane6 - bitplane5 - bitplane4;
bitplane2 = Ii - mod(Ii, 2) - bitplane8 - bitplane7 - bitplane6 - bitplane5 - bitplane4 - bitpl
bitplane1 = Ii - mod(Ii, 1) - bitplane8 - bitplane7 - bitplane6 - bitplane5 - bitplane4 - bitpl
imshow(bitplane8), title('Bit plane 8')
```

Bit plane 8



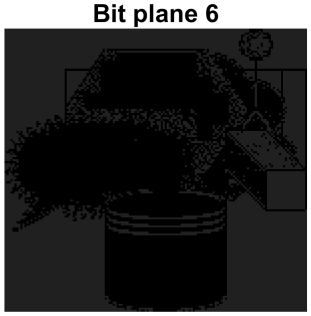
```
imshow(bitplane7), title('Bit plane 7')
```

Bit plane 7

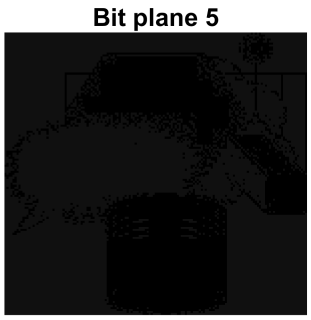


```
imshow(bitplane6), title('Bit plane 6')
```

Bit plane 6

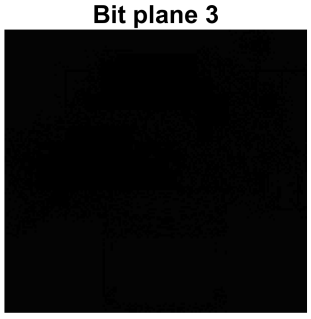


```
imshow(bitplane5), title('Bit plane 5')
```



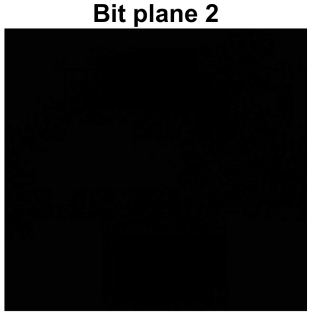

```
imshow(bitplane4), title('Bit plane 4')
```


```
imshow(bitplane3), title('Bit plane 3')
```



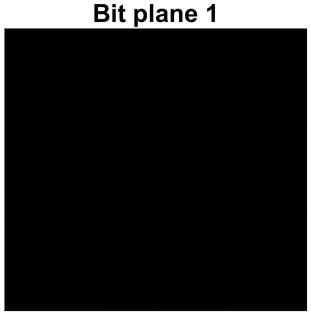

```
imshow(bitplane2), title('Bit plane 2')
```

Bit plane 2



```
imshow(bitplane1), title('Bit plane 1')
```

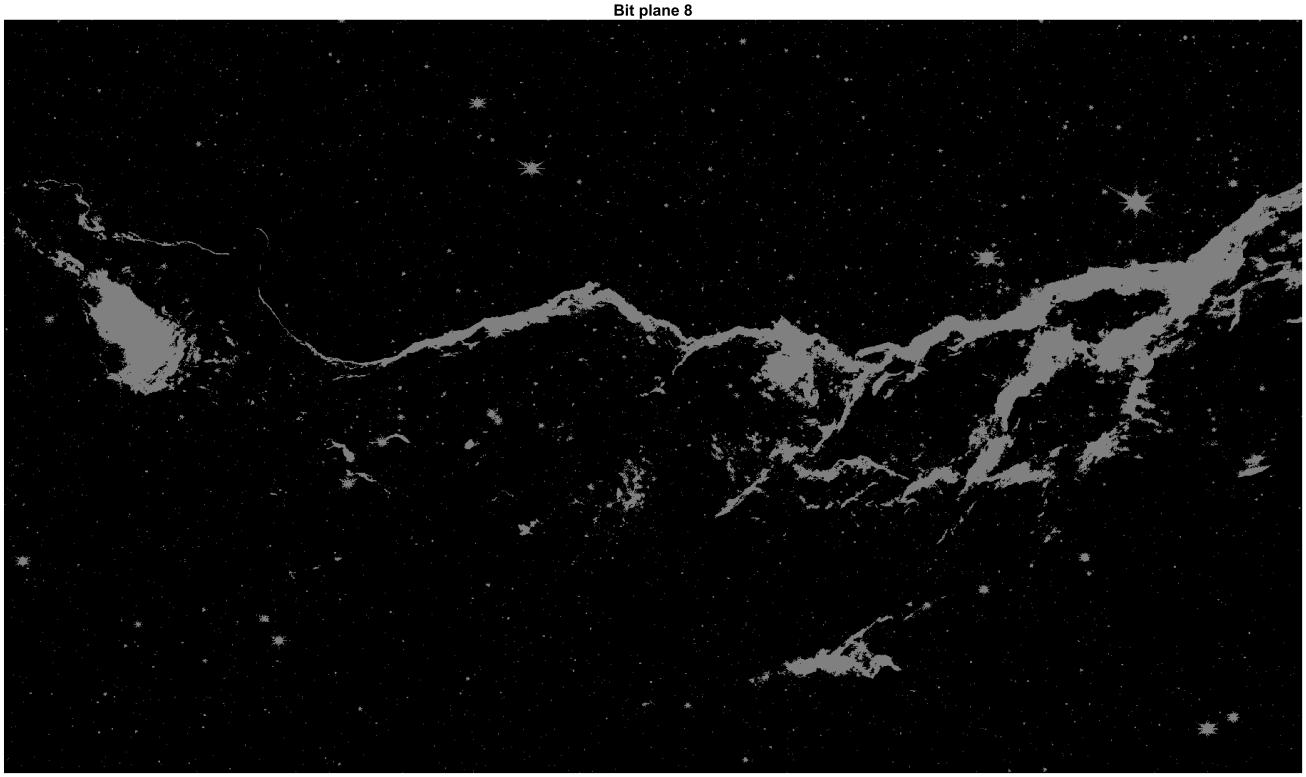
Bit plane 1



The images for each bit plane are what I expect. The image gets darker as the bit plane decreases and it reaches the least significand bits.

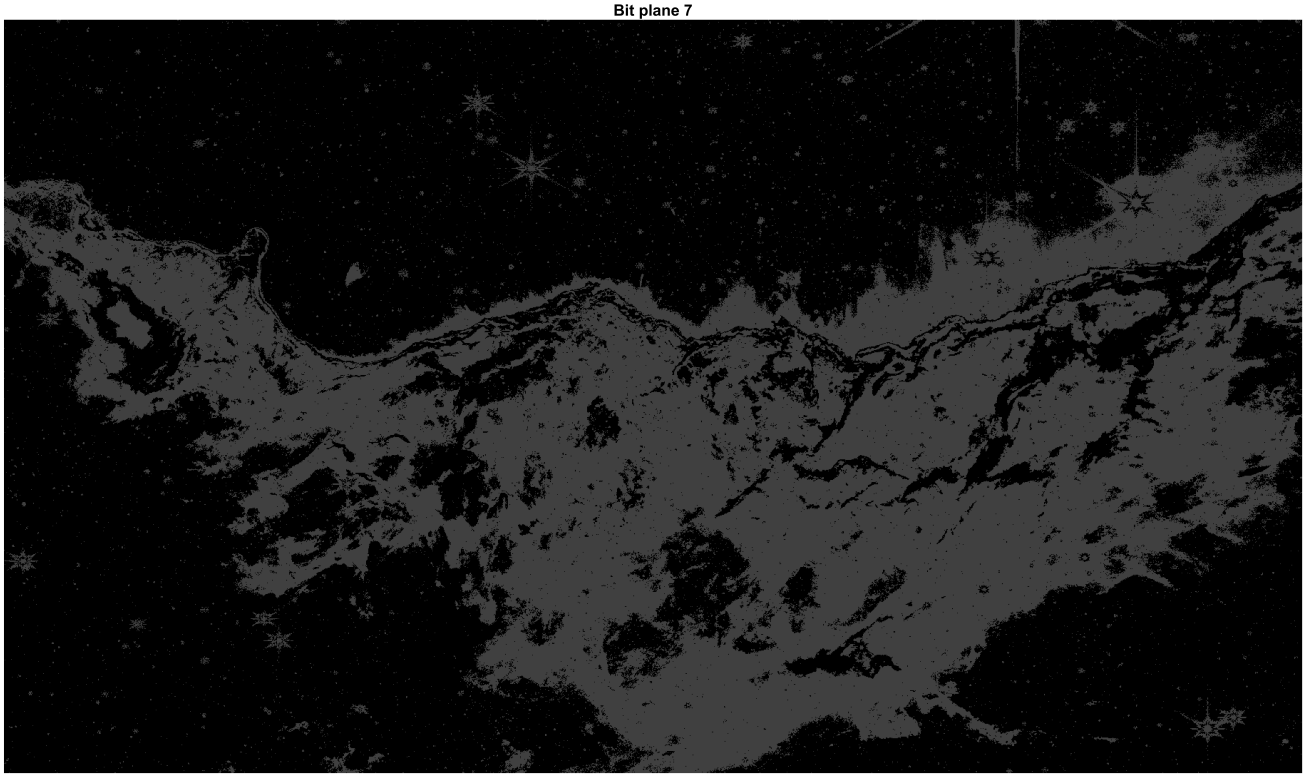
```
Ii = rgb2gray(imread('space.png'));
bitplane8 = Ii - mod(Ii, 128);
bitplane7 = Ii - mod(Ii, 64) - bitplane8;
bitplane6 = Ii - mod(Ii, 32) - bitplane8 - bitplane7;
bitplane5 = Ii - mod(Ii, 16) - bitplane8 - bitplane7 - bitplane6;
bitplane4 = Ii - mod(Ii, 8) - bitplane8 - bitplane7 - bitplane6 - bitplane5;
bitplane3 = Ii - mod(Ii, 4) - bitplane8 - bitplane7 - bitplane6 - bitplane5 - bitplane4;
bitplane2 = Ii - mod(Ii, 2) - bitplane8 - bitplane7 - bitplane6 - bitplane5 - bitplane4 - bitpl
bitplane1 = Ii - mod(Ii, 1) - bitplane8 - bitplane7 - bitplane6 - bitplane5 - bitplane4 - bitpl
imshow(bitplane8), title('Bit plane 8')
```

Bit plane 8

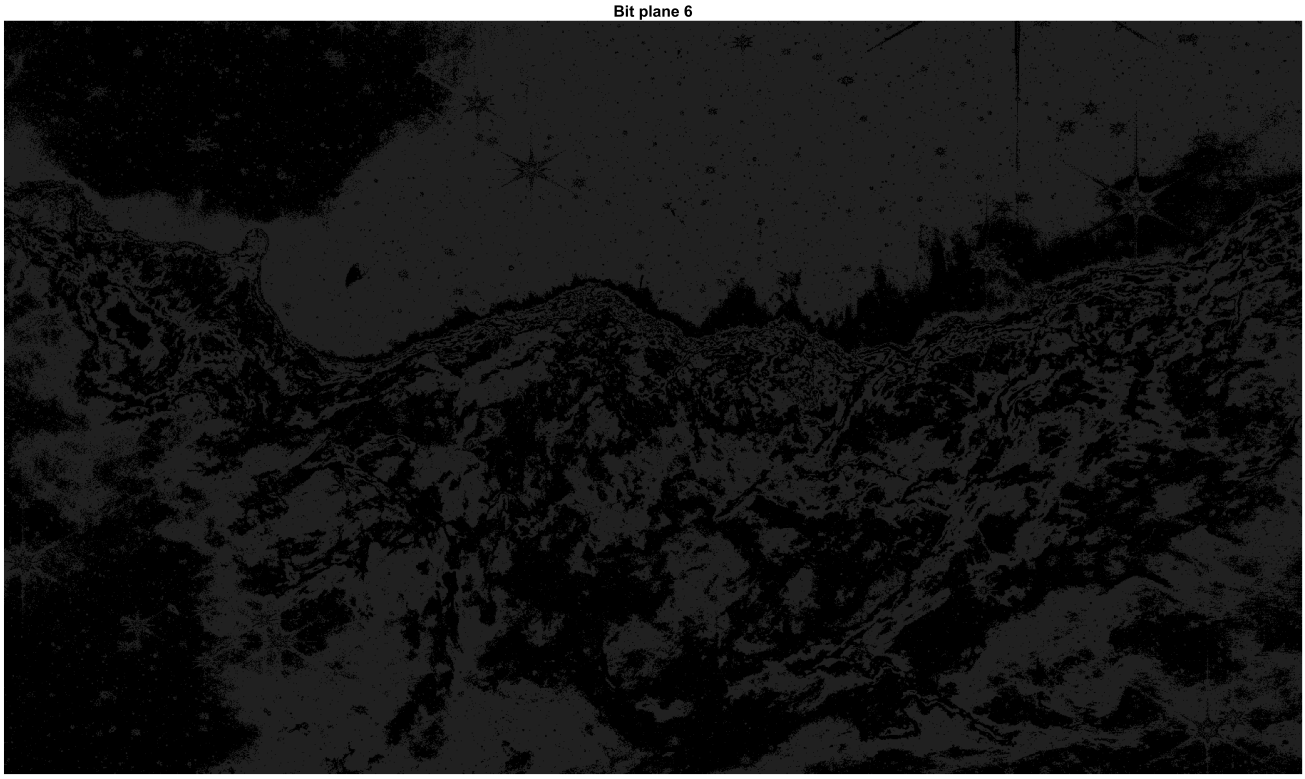


```
imshow(bitplane7), title('Bit plane 7')
```

Bit plane 7



```
imshow(bitplane6), title('Bit plane 6')
```

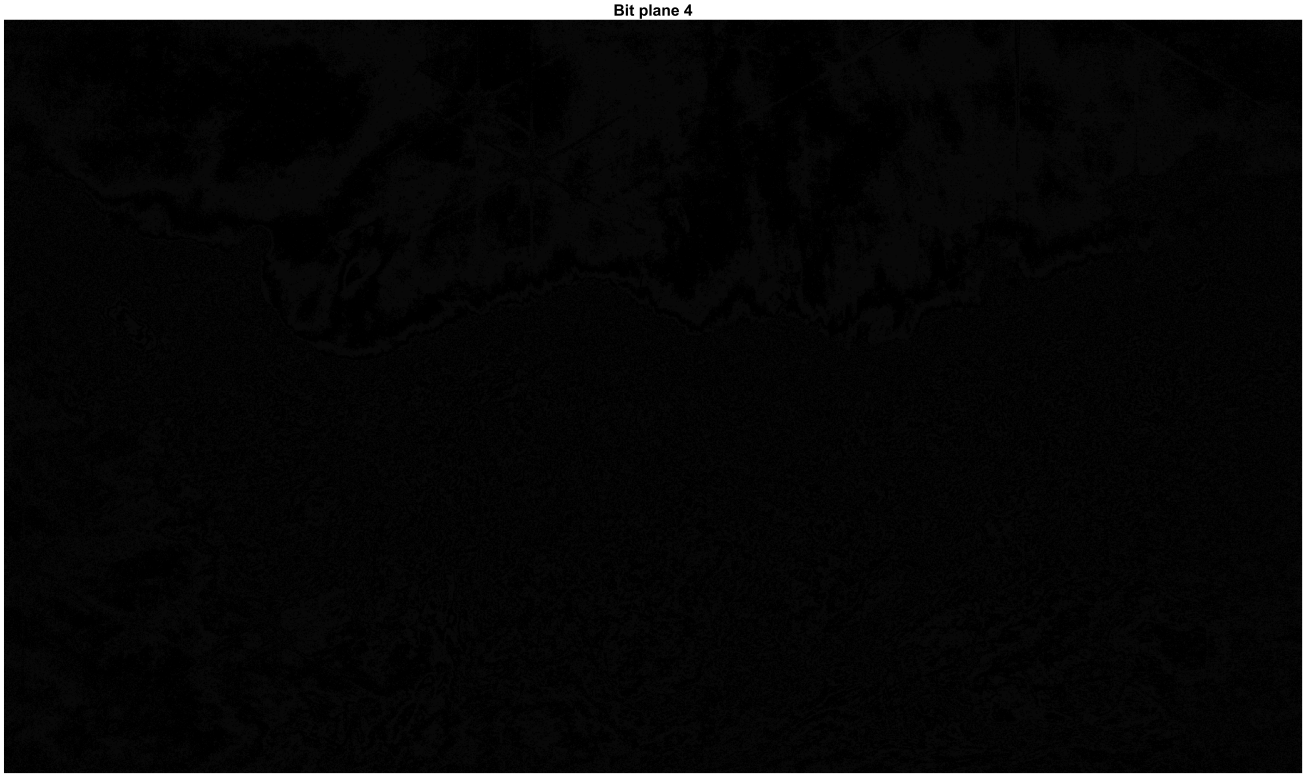



```
imshow(bitplane5), title('Bit plane 5')
```

Bit plane 5

```
imshow(bitplane4), title('Bit plane 4')
```

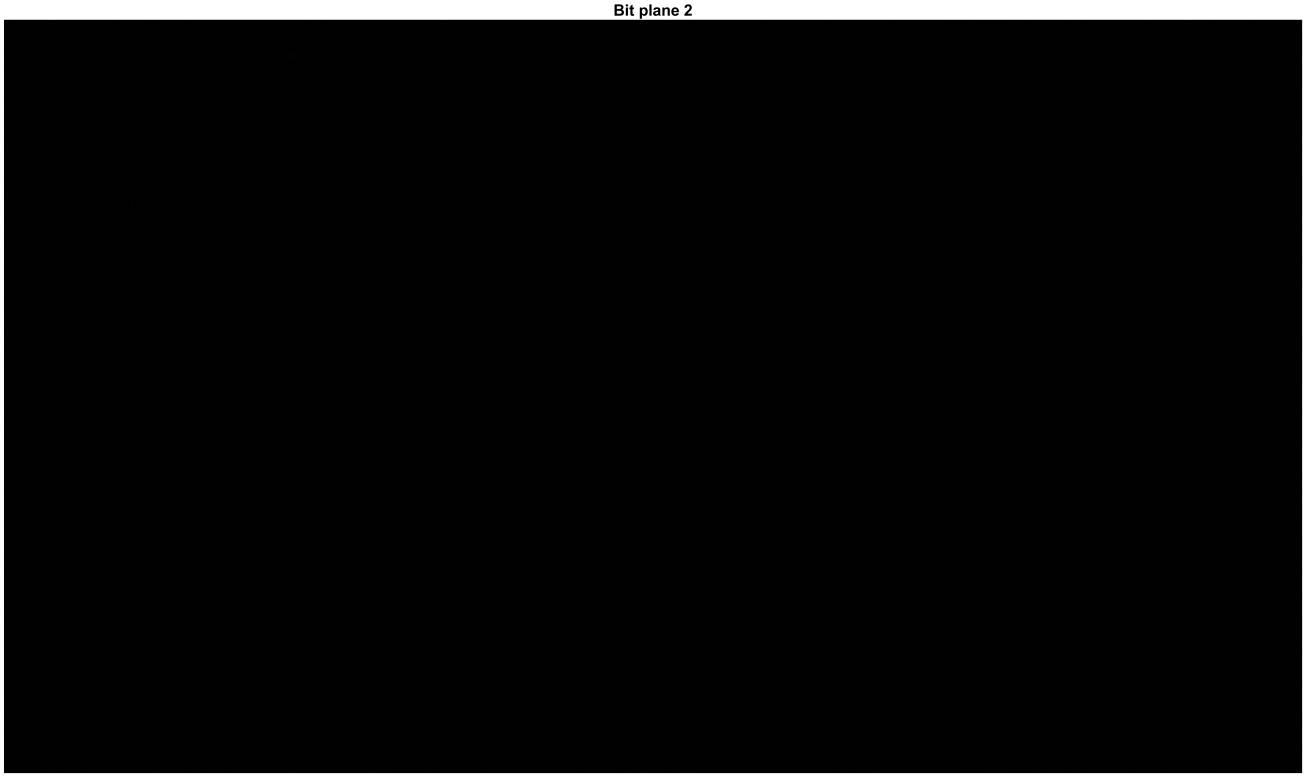
Bit plane 4



```
imshow(bitplane3), title('Bit plane 3')
```

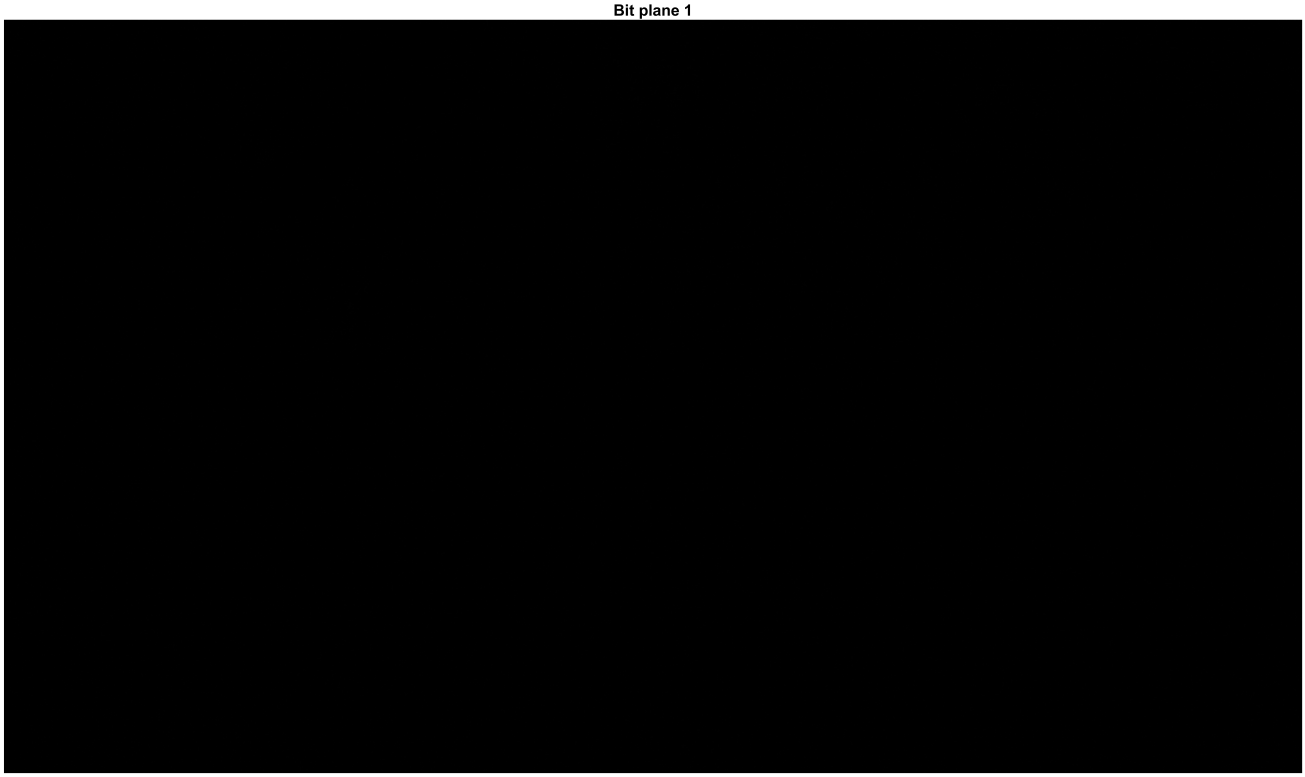

```
imshow(bitplane2), title('Bit plane 2')
```

Bit plane 2



```
imshow(bitplane1), title('Bit plane 1')
```

Bit plane 1



The images for each bit plane are what I expect

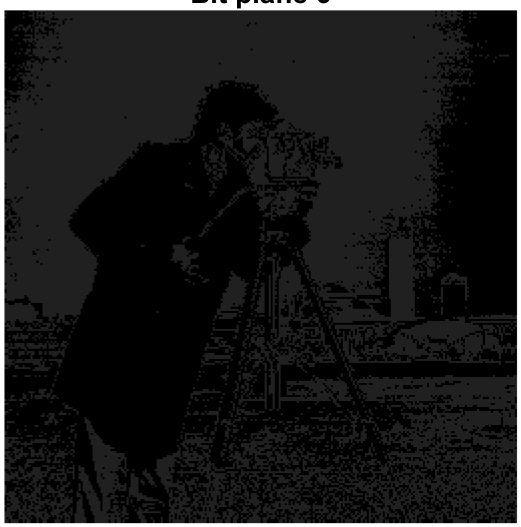
```
Ii = (imread('mycameraman2.png'));
bitplane8 = Ii - mod(Ii, 128);
bitplane7 = Ii - mod(Ii, 64) - bitplane8;
bitplane6 = Ii - mod(Ii, 32) - bitplane8 - bitplane7;
bitplane5 = Ii - mod(Ii, 16) - bitplane8 - bitplane7 - bitplane6;
bitplane4 = Ii - mod(Ii, 8) - bitplane8 - bitplane7 - bitplane6 - bitplane5;
bitplane3 = Ii - mod(Ii, 4) - bitplane8 - bitplane7 - bitplane6 - bitplane5 - bitplane4;
bitplane2 = Ii - mod(Ii, 2) - bitplane8 - bitplane7 - bitplane6 - bitplane5 - bitplane4 - bitpl
bitplane1 = Ii - mod(Ii, 1) - bitplane8 - bitplane7 - bitplane6 - bitplane5 - bitplane4 - bitpl
imshow(bitplane8), title('Bit plane 8')
```


```
imshow(bitplane7), title('Bit plane 7')
```

Bit plane 7

```
imshow(bitplane6), title('Bit plane 6')
```

Bit plane 6



```
imshow(bitplane5), title('Bit plane 5')
```

Bit plane 5

```
imshow(bitplane4), title('Bit plane 4')
```

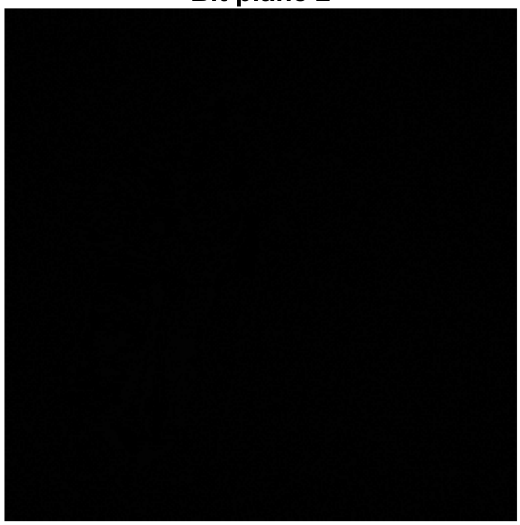
Bit plane 4

```
imshow(bitplane3), title('Bit plane 3')
```

Bit plane 3

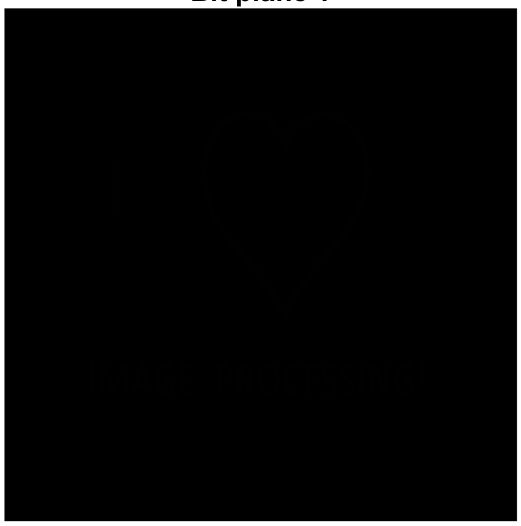
```
imshow(bitplane2), title('Bit plane 2')
```

Bit plane 2



```
imshow(bitplane1), title('Bit plane 1')
```

Bit plane 1



The images for each bit plane are what I expect. The image is of someone looking into a camera.