Devon Quaternik

Table of Contents

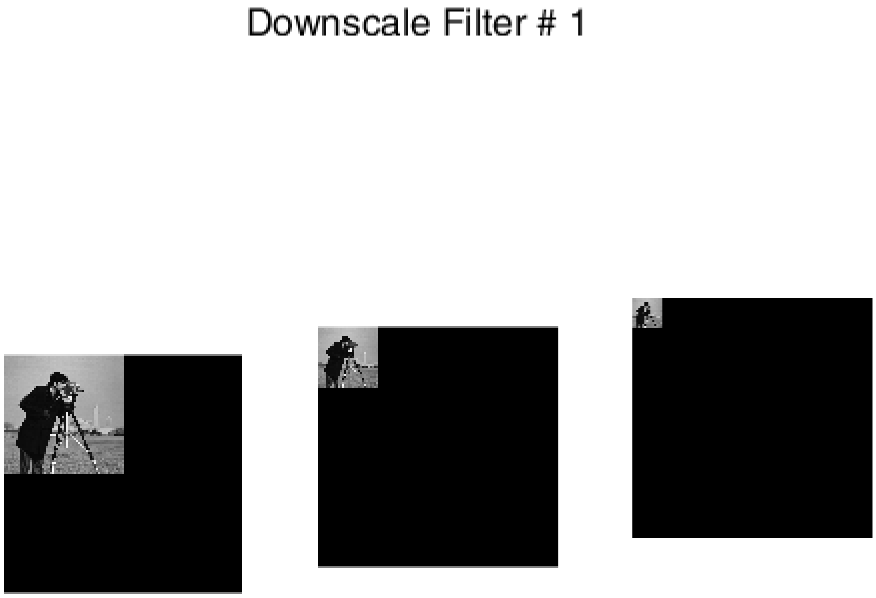
ELEN 644 HW 4

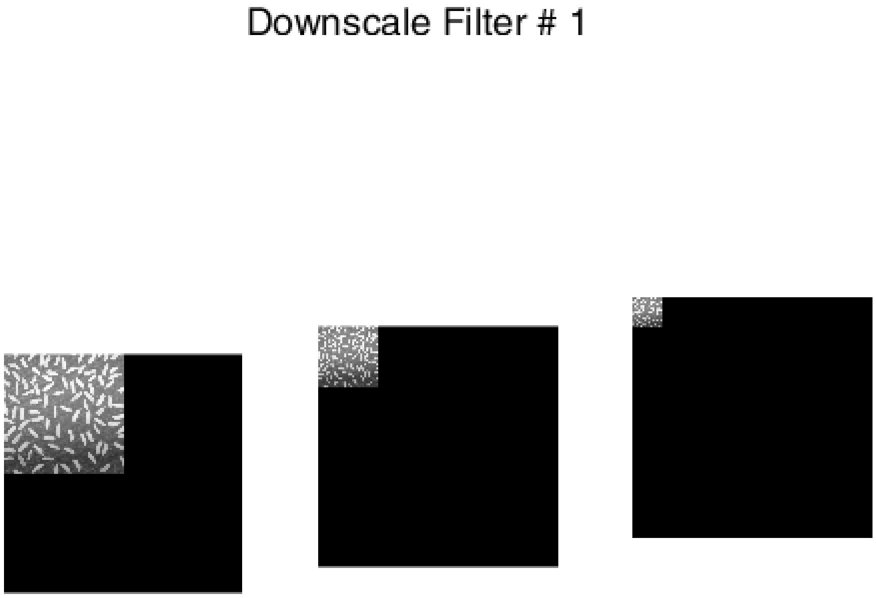
Problem 2

```
% Initialize images
im(:,:,1) = imread('cameraman.tif');
im(:,:,2) = imread('rice.png');  % Could not find lighthouse.jpg, used
 rice.png as a substitute.
im(:,:,3) = imread('5.1.09.tiff');
[n, m, nim] = size(im);

% Nubmer of levels to use
lev = 3;

% Initialize filters
h1 = 1; % Effectively the same as not filtering. Average over 1 point.
h2 = .25*[1 2 1];
h3 = (1/16)*[1 4 6 4 1];
h4 = fspecial('gaussian',7,1.0);
h5 = firpm(8,[0.0, 0.4, 0.6, 1.0],[1.0, 1.0, 0.0, 0.0]);
nf = 5;

dow1(:,:,:,1) = dscale2(im(:,:,1),h1,lev);
dow1(:,:,:,2) = dscale2(im(:,:,1),h2,lev);
dow1(:,:,:,3) = dscale2(im(:,:,1),h3,lev);
dow1(:,:,:,4) = dscale2(im(:,:,1),h4,lev);
dow1(:,:,:,5) = dscale2(im(:,:,1),h5,lev);

dow2(:,:,:,1) = dscale2(im(:,:,2),h1,lev);
dow2(:,:,:,2) = dscale2(im(:,:,2),h2,lev);
dow2(:,:,:,3) = dscale2(im(:,:,2),h3,lev);
dow2(:,:,:,4) = dscale2(im(:,:,2),h4,lev);
dow2(:,:,:,5) = dscale2(im(:,:,2),h5,lev);

dow3(:,:,:,1) = dscale2(im(:,:,3),h1,lev);
dow3(:,:,:,2) = dscale2(im(:,:,3),h2,lev);
dow3(:,:,:,3) = dscale2(im(:,:,3),h3,lev);
dow3(:,:,:,4) = dscale2(im(:,:,3),h4,lev);
dow3(:,:,:,5) = dscale2(im(:,:,3),h5,lev);

[m, n, p, q] = size(dow1);
```

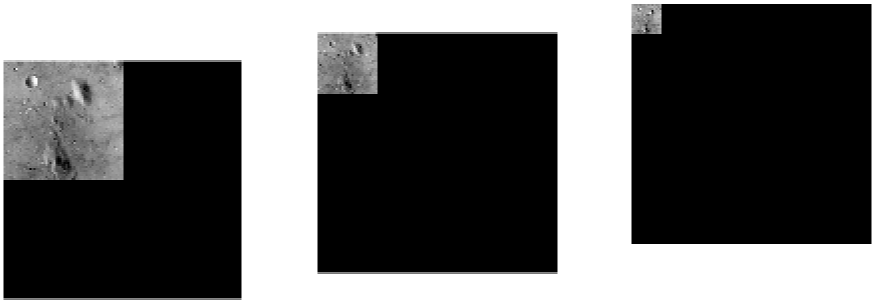
```
% Display levels 1, 2, 3
count = 0;
for j = 0:nf-1
    for i = 1:lev
        str2 = sprintf('Downscale Filter # %d',j+1);
        figure(3*j+1);
        subplot(1,lev,i),imshow(dow1(:,:,i,j+1),[]);
        suptitle(str2);
        figure(3*j+2);
        subplot(1,lev,i),imshow(dow2(:,:,i,j+1),[]);
        suptitle(str2);
        figure(3*j+3);
        subplot(1,lev,i),imshow(dow3(:,:,i,j+1),[]);
        suptitle(str2);
    end
    count = count+1;
end

disp('Zero padding used to unify image size and allow for certain data
 structures.');
disp('Downscaling, it is difficult to tell eexactly which filters work
 best. Although 2 and 3 appear slightly more clear than the rest, none
 are particularly bad.');
```

```
Zero padding used to unify image size and allow for certain data
 structures.
Downscaling, it is difficult to tell eexactly which filters work best.
 Although 2 and 3 appear slightly more clear than the rest, none are
 particularly bad.
```

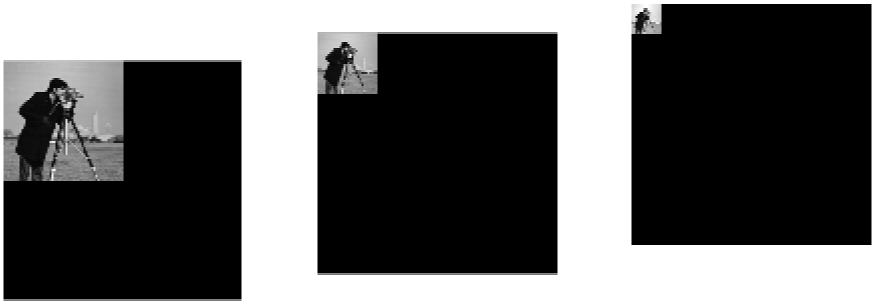

Downscale Filter # 1

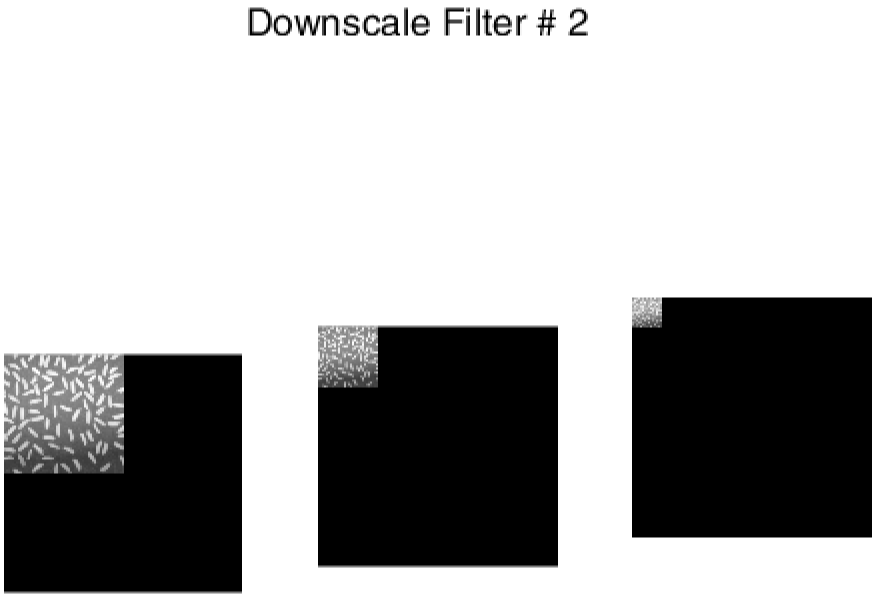

Downscale Filter # 1

Downscale Filter # 1

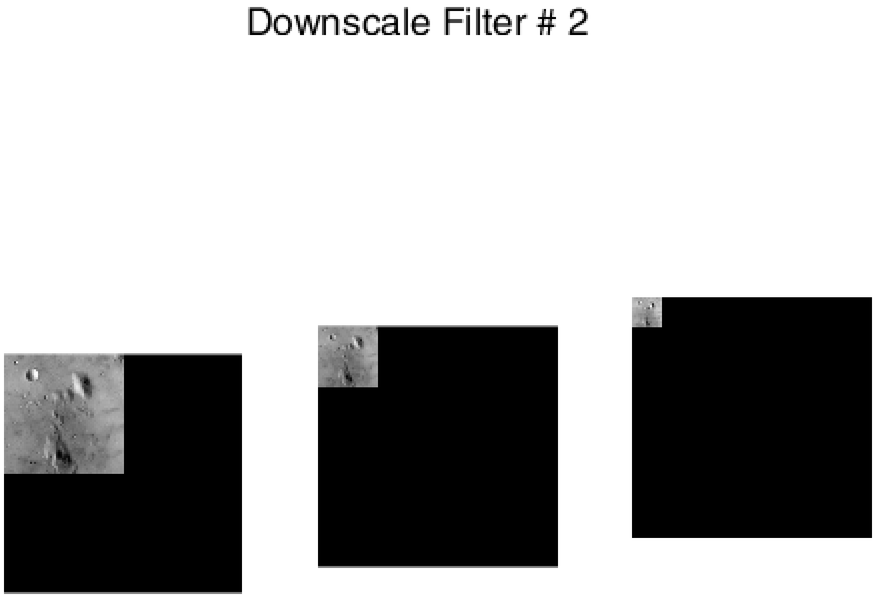


Downscale Filter # 2

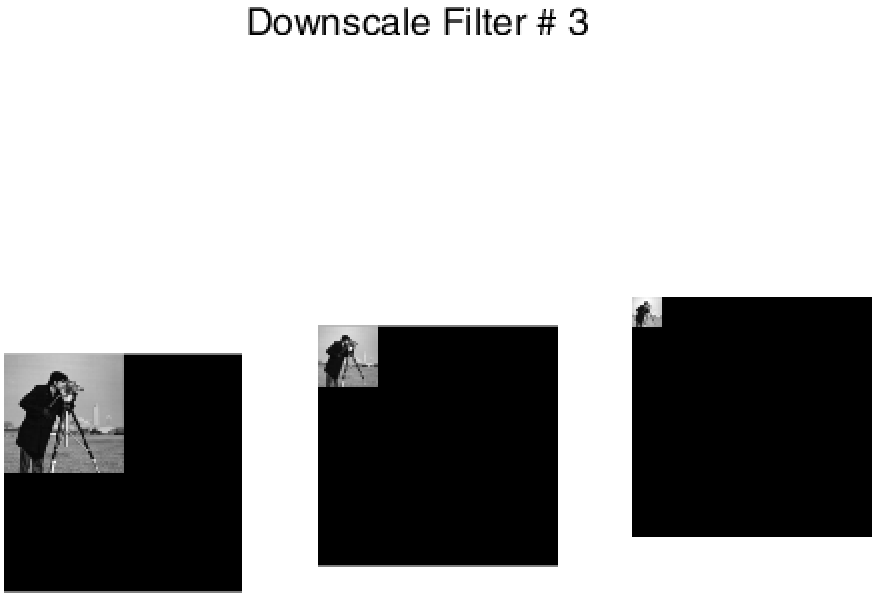


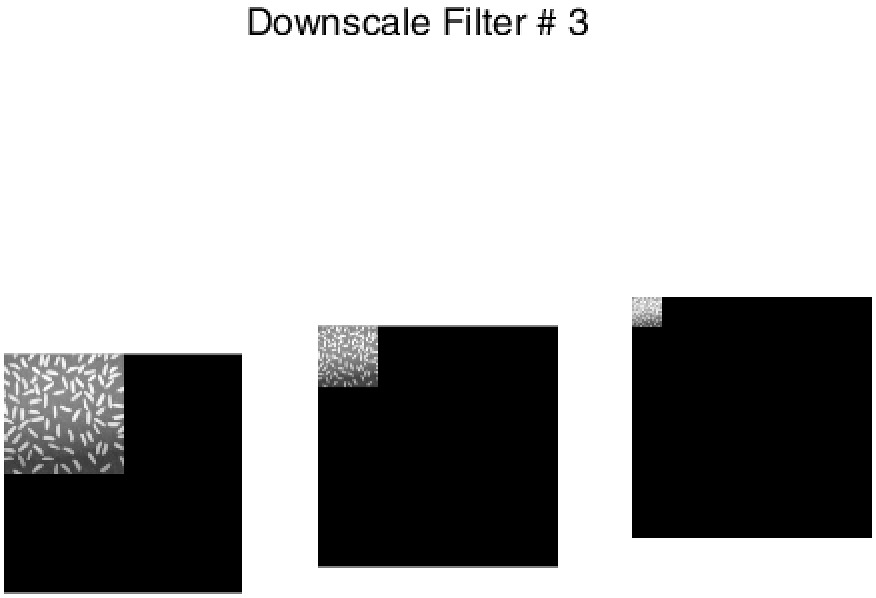


Downscale Filter # 2

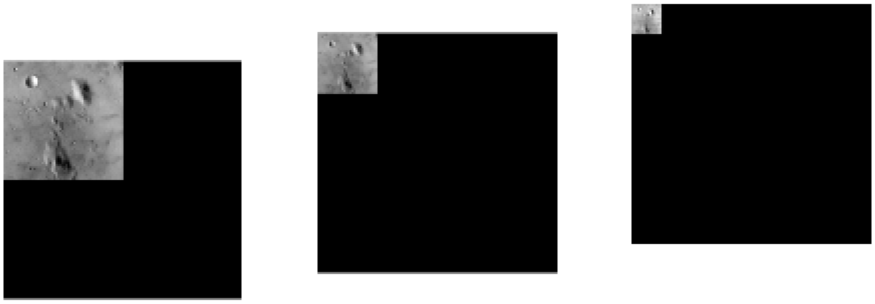


Downscale Filter # 2

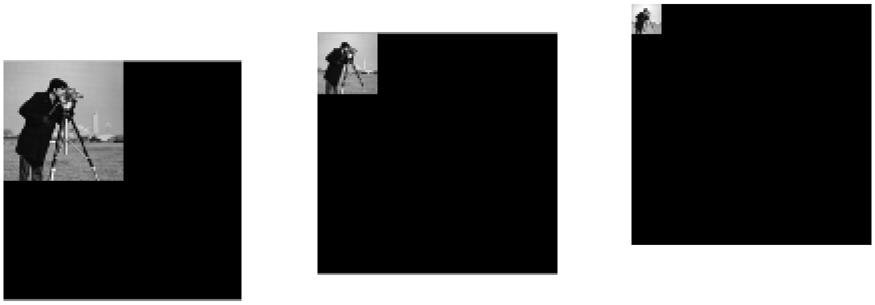

Downscale Filter # 3

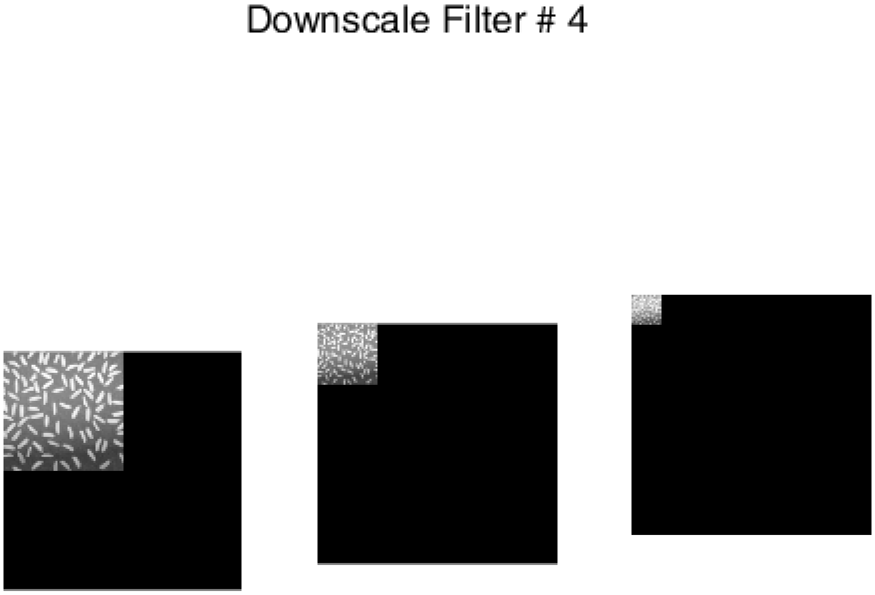

Downscale Filter # 3

Downscale Filter # 3

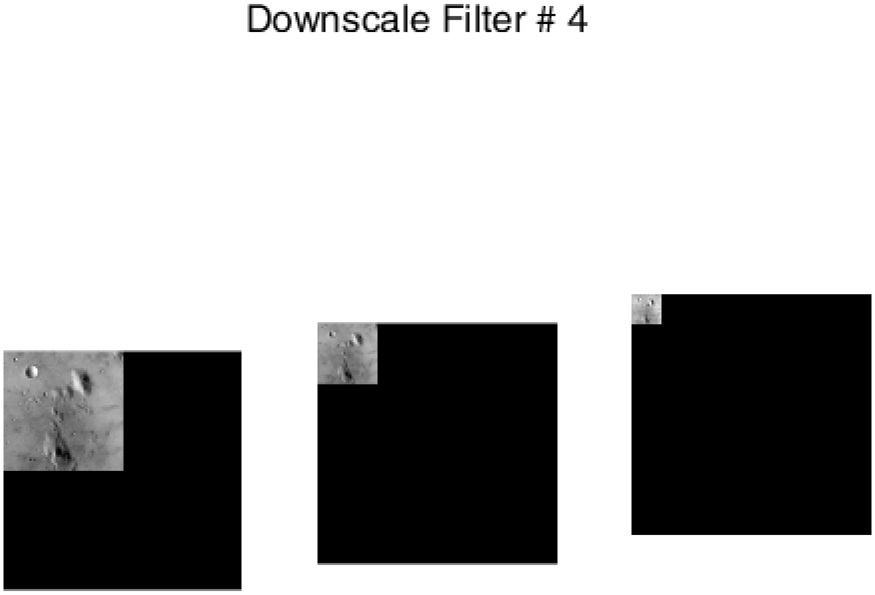


Downscale Filter # 4

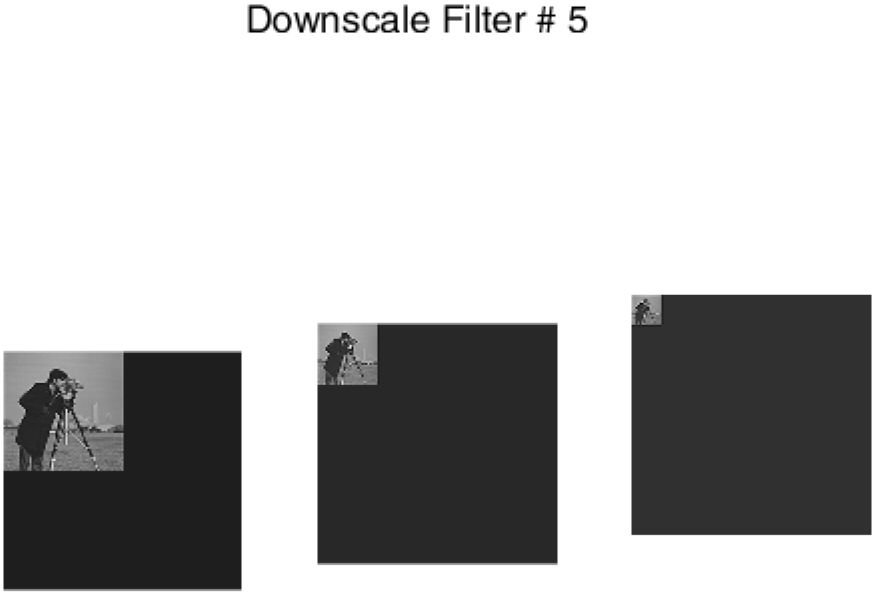




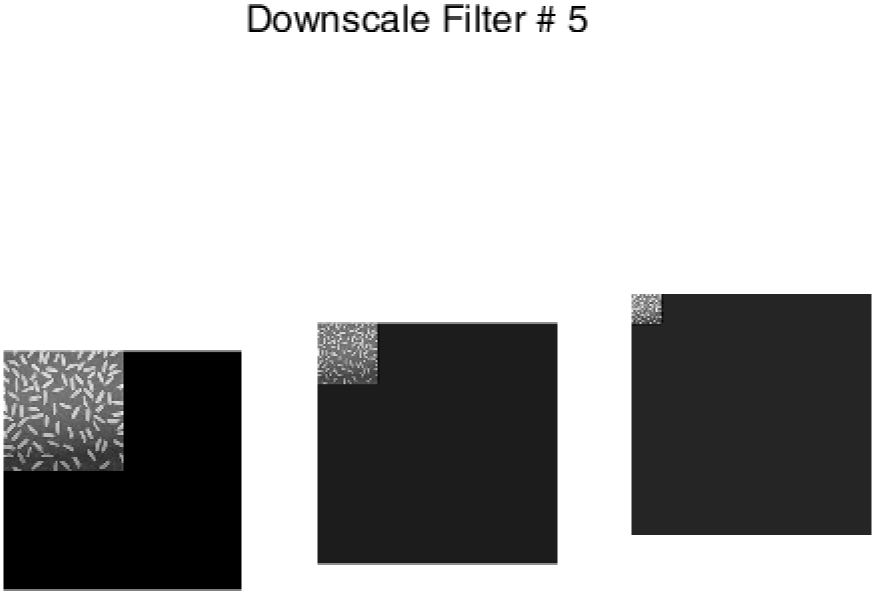
Downscale Filter # 4



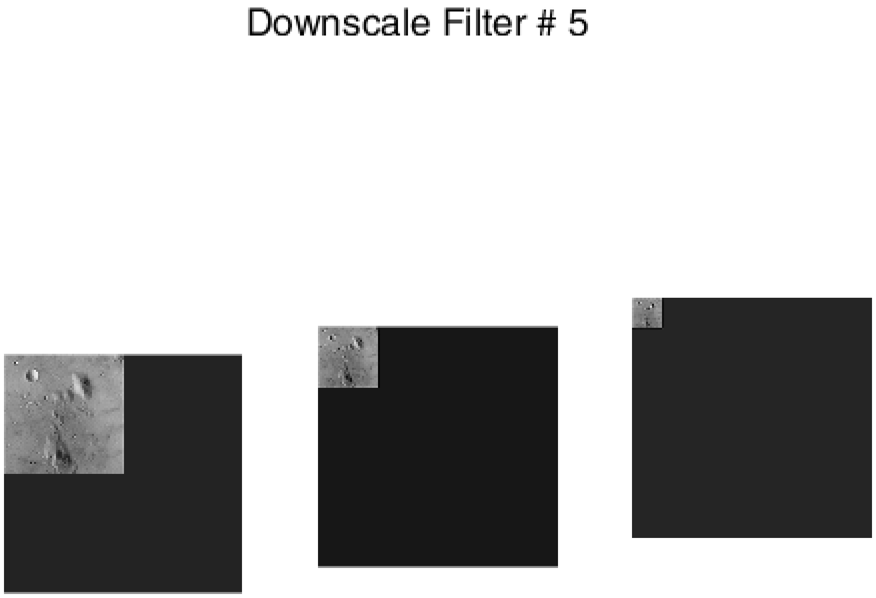
Downscale Filter # 4



Downscale Filter # 5



Downscale Filter # 5



Downscale Filter # 5

Problem 2

```
up1(:,:,:,1) = uscale2(dow1(:,:,:,1),h1,lev);
up1(:,:,:,2) = uscale2(dow1(:,:,:,2),h2,lev);
up1(:,:,:,3) = uscale2(dow1(:,:,:,3),h3,lev);
up1(:,:,:,4) = uscale2(dow1(:,:,:,4),h4,lev);
up1(:,:,:,5) = uscale2(dow1(:,:,:,5),h5,lev);

up2(:,:,:,1) = uscale2(dow2(:,:,:,1),h1,lev);
up2(:,:,:,2) = uscale2(dow2(:,:,:,2),h2,lev);
up2(:,:,:,3) = uscale2(dow2(:,:,:,3),h3,lev);
up2(:,:,:,4) = uscale2(dow2(:,:,:,4),h4,lev);
up2(:,:,:,5) = uscale2(dow2(:,:,:,5),h5,lev);

up3(:,:,:,1) = uscale2(dow3(:,:,:,1),h1,lev);
up3(:,:,:,2) = uscale2(dow3(:,:,:,2),h2,lev);
up3(:,:,:,3) = uscale2(dow3(:,:,:,3),h3,lev);
up3(:,:,:,4) = uscale2(dow3(:,:,:,4),h4,lev);
up3(:,:,:,5) = uscale2(dow3(:,:,:,5),h5,lev);

% Display estimated levels 0, 1, 2

for j = 0:nf-1
    for i = 1:lev
        str2 = sprintf('Upscale Filter # %d',j+1);
        figure(3*(j+count)+1);
        subplot(1,lev,i),imshow(up1(:,:,i,j+1),[]);
        suptitle(str2);
        figure(3*(j+count)+2);
        subplot(1,lev,i),imshow(up2(:,:,i,j+1),[]);
        suptitle(str2);
        figure(3*(j+count)+3);
        subplot(1,lev,i),imshow(up3(:,:,i,j+1),[]);
        suptitle(str2);
    end
end

% Create residual images

for j = 0:nf-1
    for i = 1:lev
        if i == 1
            res1(:,:,i,j+1) = double(im(:,:,1)) - double(up1(:,:,1,j
+1));
            res2(:,:,i,j+1) = double(im(:,:,2)) - double(up2(:,:,1,j
+1));
            res3(:,:,i,j+1) = double(im(:,:,3)) - double(up3(:,:,1,j
+1));
        else
            res1(:,:,i,j+1) = dow1(:,:,i-1,j+1) - up1(:,:,i,j+1);
            res2(:,:,i,j+1) = dow2(:,:,i-1,j+1) - up2(:,:,i,j+1);
            res3(:,:,i,j+1) = dow3(:,:,i-1,j+1) - up3(:,:,i,j+1);
        end
        ms1(i,j+1) = meansqr(res1(:,:,i,j+1));
        ms2(i,j+1) = meansqr(res2(:,:,i,j+1));
        ms3(i,j+1) = meansqr(res3(:,:,i,j+1));

        str1 = sprintf('MSE Residual Image 1, Filter # %d Level %d:
 %d',j+1,i-1,ms1(i,j+1));
        disp(str1);
        str1 = sprintf('MSE Residual Image 2, Filter # %d Level %d:
 %d',j+1,i-1,ms2(i,j+1));
        disp(str1);
        str1 = sprintf('MSE Residual Image 3, Filter # %d Level %d:
 %d',j+1,i-1,ms3(i,j+1));
        disp(str1);

        str2 = sprintf('Image 1 Filter # %d Residual',j+1);
        str3 = sprintf('Level %d',i-1);
        figure(6*(j+count)+1);
        subplot(1,lev,i),imshow(res1(:,:,i,j+1),[]);
        suptitle(str2);
        figure(6*(j+count)+2);
        subplot(1,lev,i), histogram(res1(:,:,i,j+1));
        title(str3);
        suptitle(str2);

        str2 = sprintf('Image 2 Filter # %d Residual',j+1);
        figure(6*(j+count)+3);
        subplot(1,lev,i),imshow(res2(:,:,i,j+1),[]);
        suptitle(str2);
        figure(6*(j+count)+4);
        subplot(1,lev,i), histogram(res2(:,:,i,j+1));
        title(str3);
        suptitle(str2);

        str2 = sprintf('Image 3 Filter # %d Residual',j+1);
        figure(6*(j+count)+5);
        subplot(1,lev,i),imshow(res3(:,:,i,j+1),[]);
        suptitle(str2);
        figure(6*(j+count)+6);
        subplot(1,lev,i), histogram(res3(:,:,i,j+1));
        title(str3);
        suptitle(str2);

    end
end
disp('No filter is useless for upscaling, making it bad choice for
 pyramid making.');
disp('Filter 2 works well and the interpolated image looks correct.
 You can easily see the edges on the residual at any level.');
disp('Filter 3 is similar to 2, but appears to be slightly worse
 overall. Again you can see images at any level.');
disp('Filters 4 and 5 both have noticeable banding across both.
 Neither are particularly suited to this application and the results
 are poor, if still visible');
disp('Order from best to worst is filter 2,3,4,5,1');
```

```
MSE Residual Image 1, Filter # 1 Level 0: 1.798193e+04
MSE Residual Image 2, Filter # 1 Level 0: 1.418152e+04
MSE Residual Image 3, Filter # 1 Level 0: 1.709191e+04
MSE Residual Image 1, Filter # 1 Level 1: 4.488263e+03
MSE Residual Image 2, Filter # 1 Level 1: 3.549687e+03
MSE Residual Image 3, Filter # 1 Level 1: 4.274314e+03
MSE Residual Image 1, Filter # 1 Level 2: 1.131877e+03
MSE Residual Image 2, Filter # 1 Level 2: 8.933270e+02
MSE Residual Image 3, Filter # 1 Level 2: 1.069524e+03
MSE Residual Image 1, Filter # 2 Level 0: 4.758880e+03
MSE Residual Image 2, Filter # 2 Level 0: 3.697415e+03
MSE Residual Image 3, Filter # 2 Level 0: 4.399677e+03
MSE Residual Image 1, Filter # 2 Level 1: 1.172520e+03
MSE Residual Image 2, Filter # 2 Level 1: 9.278472e+02
MSE Residual Image 3, Filter # 2 Level 1: 1.094569e+03
MSE Residual Image 1, Filter # 2 Level 2: 2.925246e+02
MSE Residual Image 2, Filter # 2 Level 2: 2.358596e+02
MSE Residual Image 3, Filter # 2 Level 2: 2.747317e+02
MSE Residual Image 1, Filter # 3 Level 0: 4.867461e+03
MSE Residual Image 2, Filter # 3 Level 0: 3.773645e+03
MSE Residual Image 3, Filter # 3 Level 0: 4.452162e+03
MSE Residual Image 1, Filter # 3 Level 1: 1.178530e+03
MSE Residual Image 2, Filter # 3 Level 1: 9.411225e+02
MSE Residual Image 3, Filter # 3 Level 1: 1.100716e+03
MSE Residual Image 1, Filter # 3 Level 2: 2.937034e+02
MSE Residual Image 2, Filter # 3 Level 2: 2.342697e+02
MSE Residual Image 3, Filter # 3 Level 2: 2.760655e+02
MSE Residual Image 1, Filter # 4 Level 0: 4.862683e+03
MSE Residual Image 2, Filter # 4 Level 0: 3.771003e+03
MSE Residual Image 3, Filter # 4 Level 0: 4.450017e+03
MSE Residual Image 1, Filter # 4 Level 1: 1.179326e+03
MSE Residual Image 2, Filter # 4 Level 1: 9.412443e+02
MSE Residual Image 3, Filter # 4 Level 1: 1.101406e+03
MSE Residual Image 1, Filter # 4 Level 2: 2.939331e+02
MSE Residual Image 2, Filter # 4 Level 2: 2.344315e+02
MSE Residual Image 3, Filter # 4 Level 2: 2.763500e+02
MSE Residual Image 1, Filter # 5 Level 0: 8.559735e+03
MSE Residual Image 2, Filter # 5 Level 0: 6.708571e+03
MSE Residual Image 3, Filter # 5 Level 0: 8.177129e+03
MSE Residual Image 1, Filter # 5 Level 1: 1.338921e+03
MSE Residual Image 2, Filter # 5 Level 1: 1.040605e+03
MSE Residual Image 3, Filter # 5 Level 1: 1.267818e+03
MSE Residual Image 1, Filter # 5 Level 2: 2.106971e+02
MSE Residual Image 2, Filter # 5 Level 2: 1.739011e+02
MSE Residual Image 3, Filter # 5 Level 2: 1.966051e+02
```

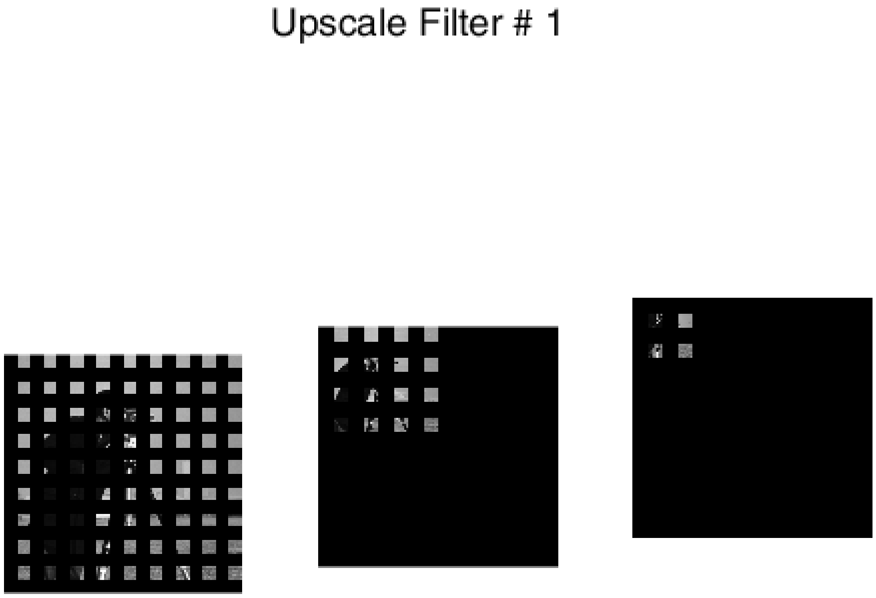
No filter is useless for upscaling, making it bad choice for pyramid making.

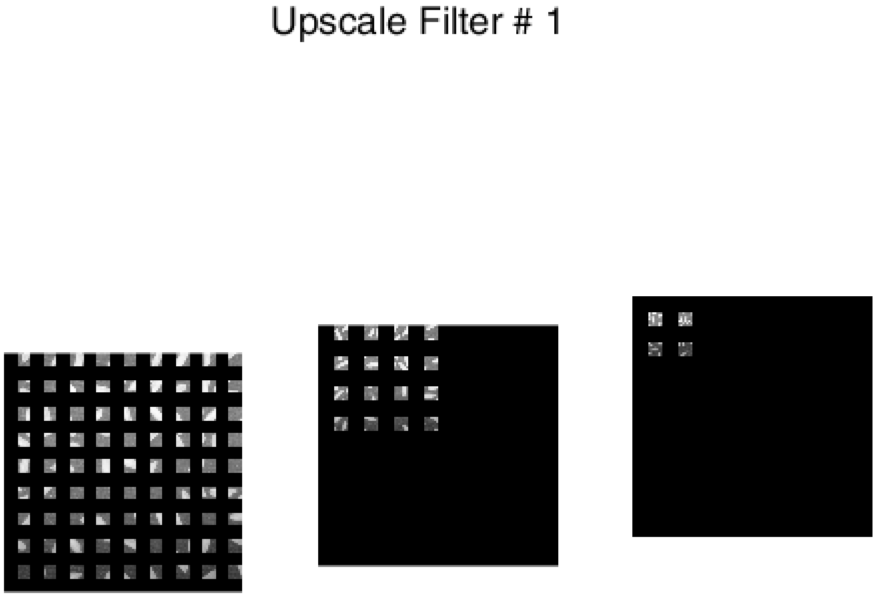
Filter 2 works well and the interpolated image looks correct. You can easily see the edges on the residual at any level.

Filter 3 is similar to 2, but appears to be slightly worse overall. Again you can see images at any level.

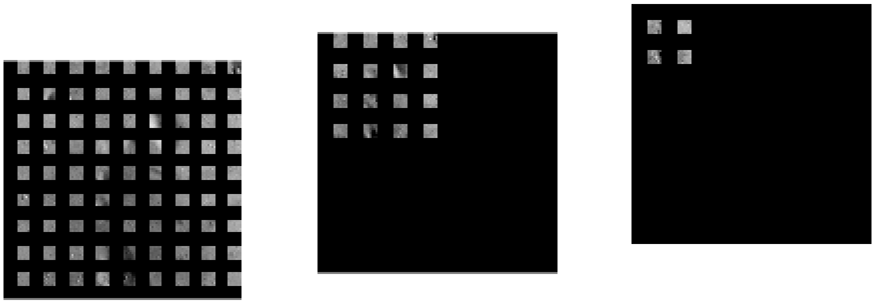
Filters 4 and 5 both have noticeable banding across both. Neither are particularly suited to this application and the results are poor, if still visible

Order from best to worst is filter 2,3,4,5,1

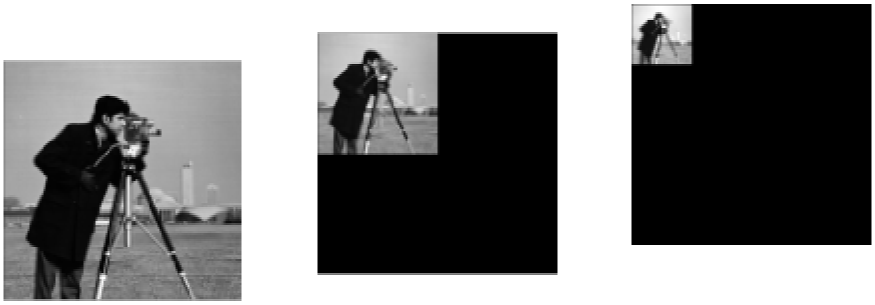

Upscale Filter # 1

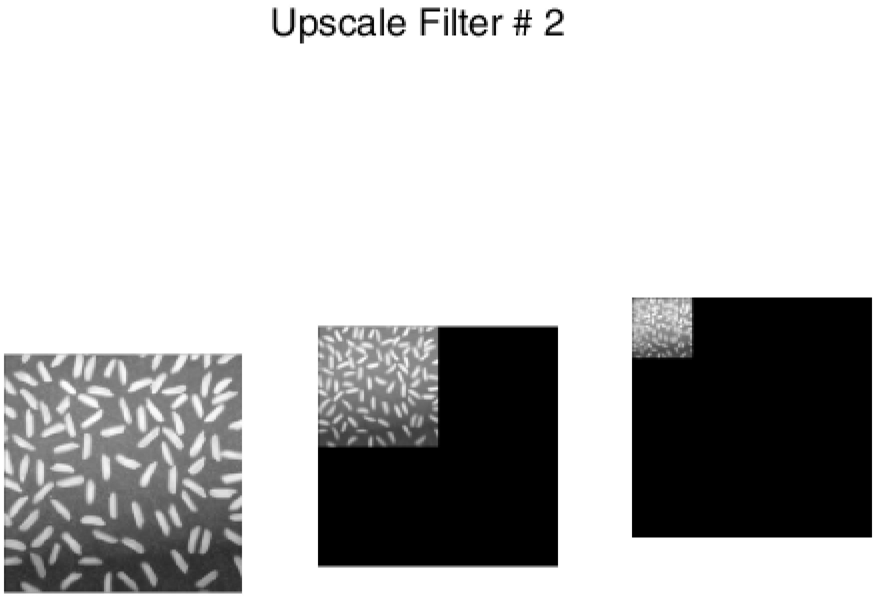

Upscale Filter # 1

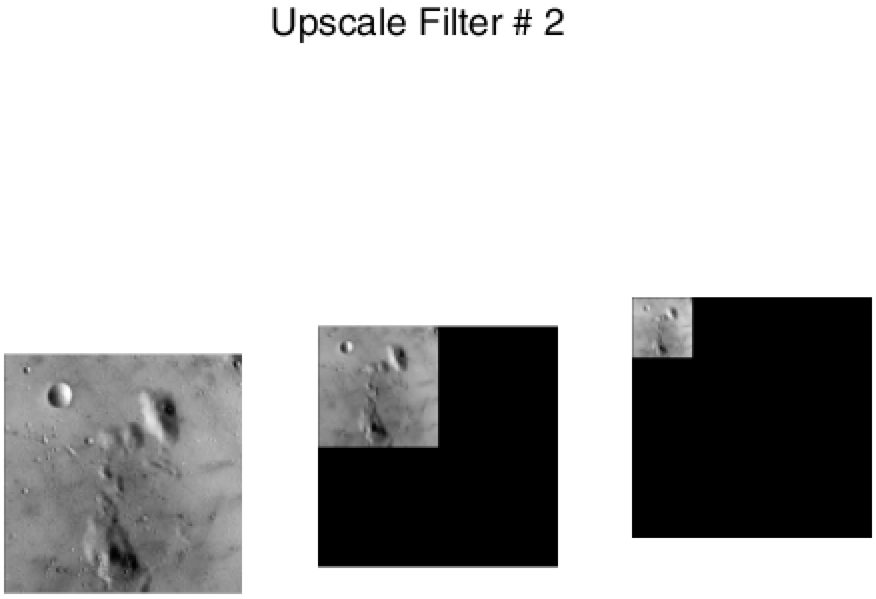
Upscale Filter # 1



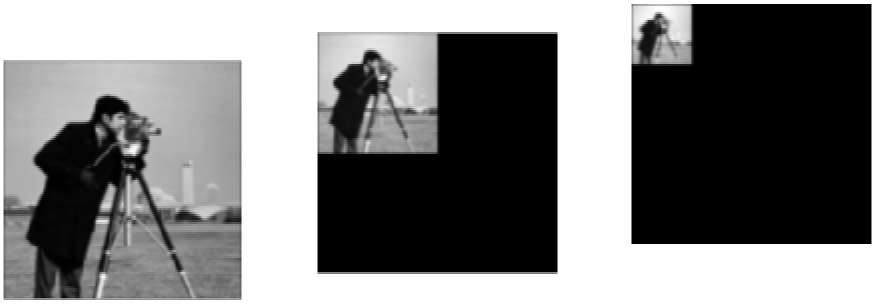
Upscale Filter # 2



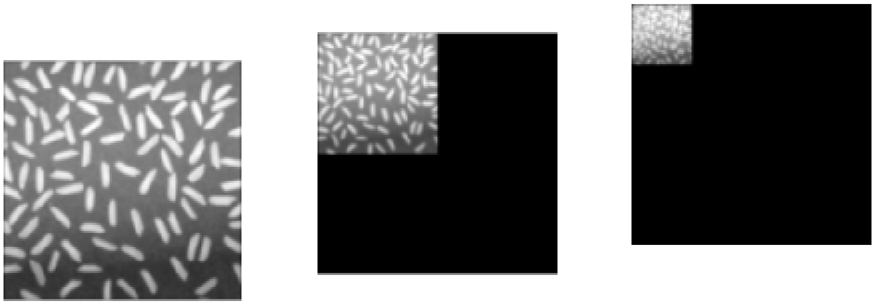

Upscale Filter # 2


Upscale Filter # 2

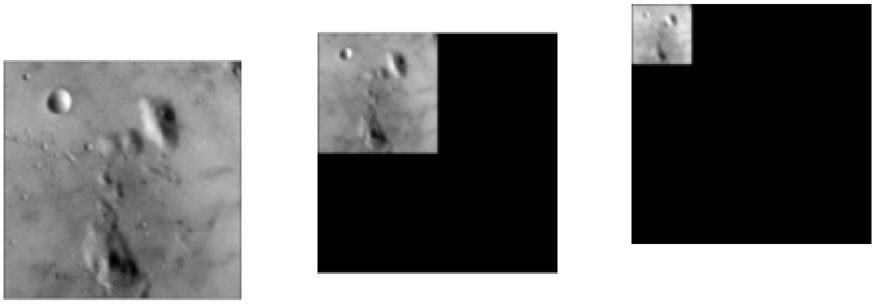
Upscale Filter # 3



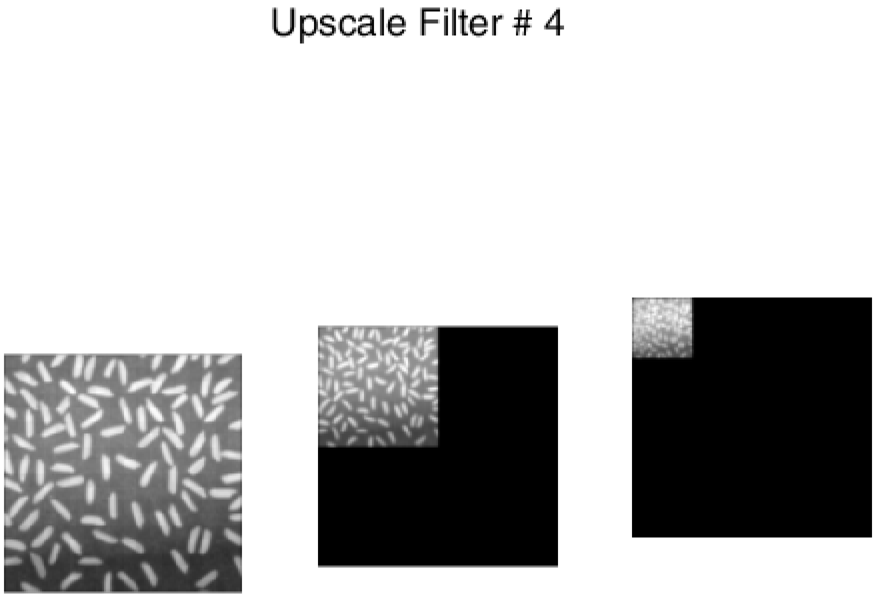
Upscale Filter # 3

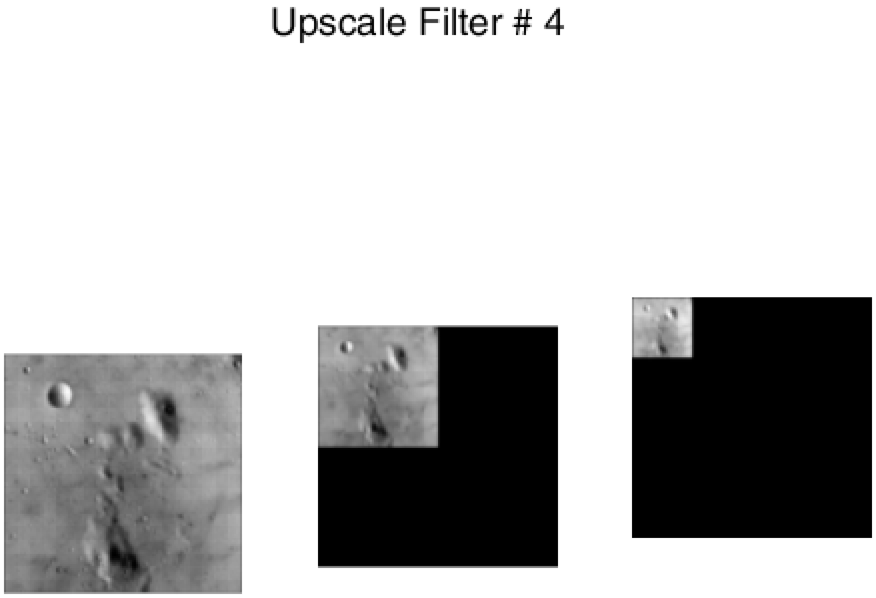


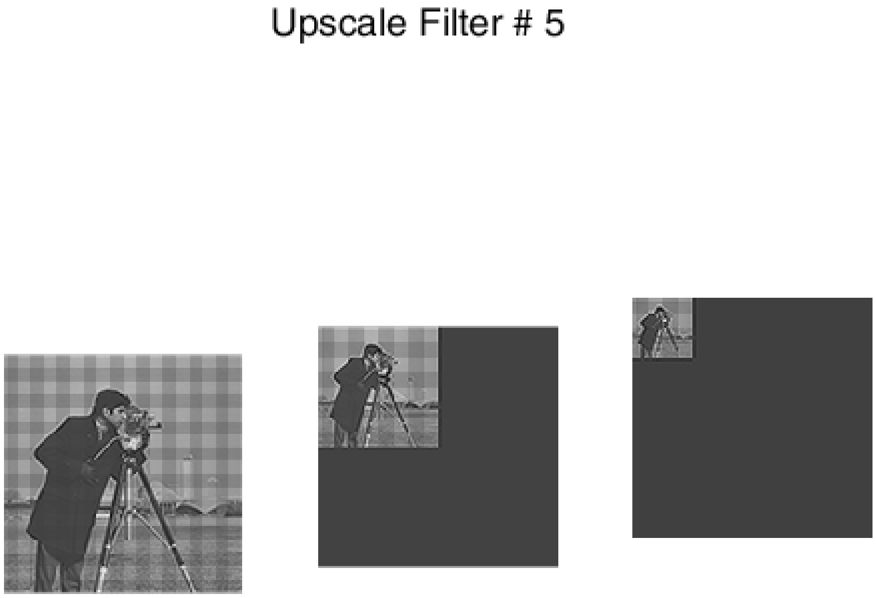
Upscale Filter # 3



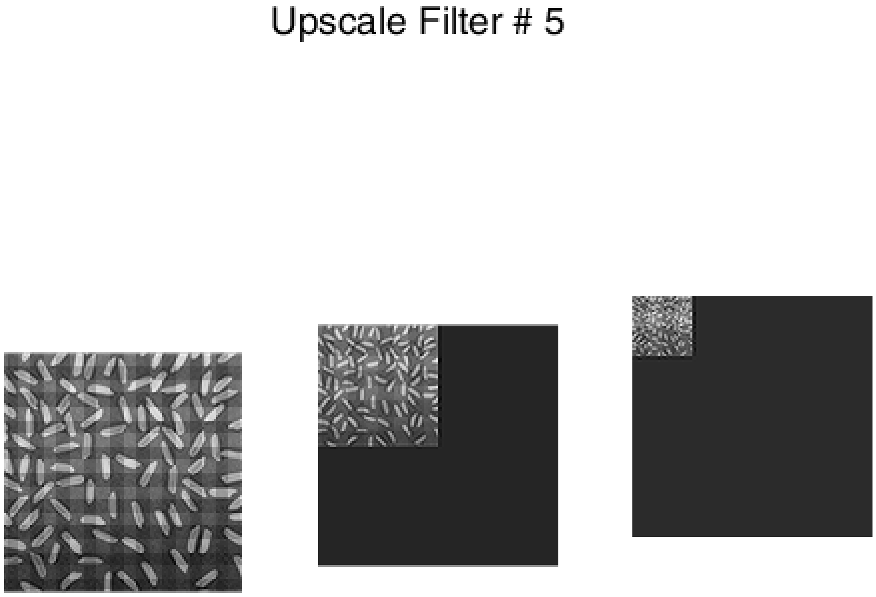
Upscale Filter # 4


Upscale Filter # 4

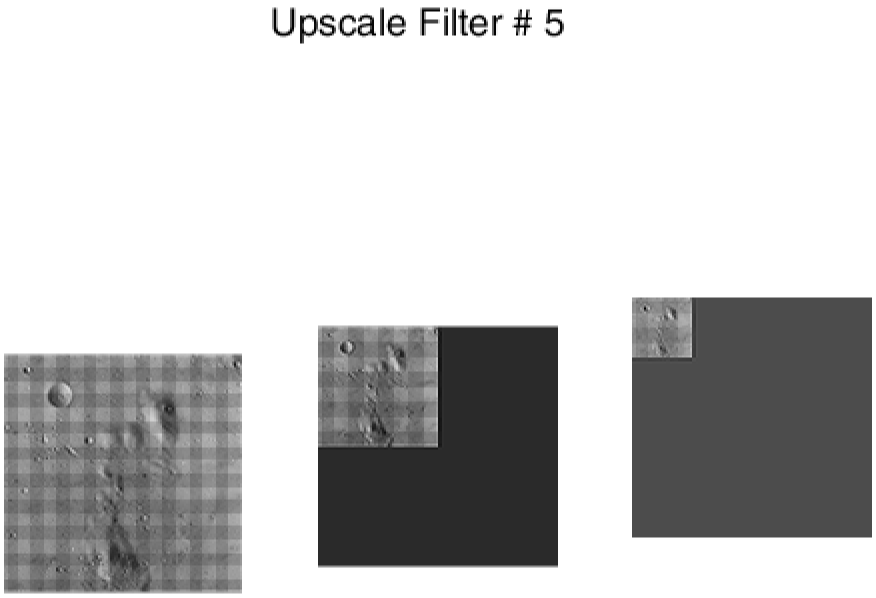

Upscale Filter # 4



Upscale Filter # 5



Upscale Filter # 5



Upscale Filter # 5

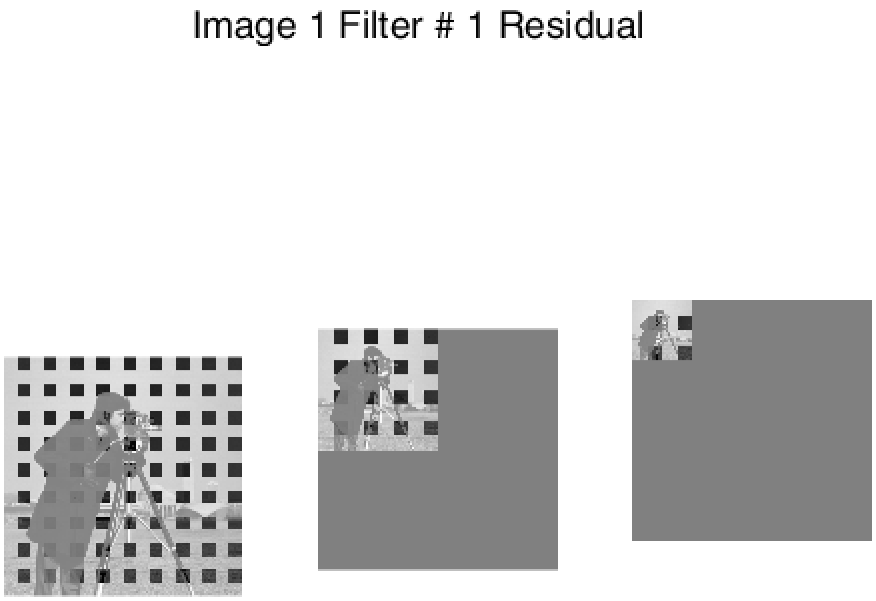


Image 1 Filter # 1 Residual

Image 1 Filter # 1 Residual

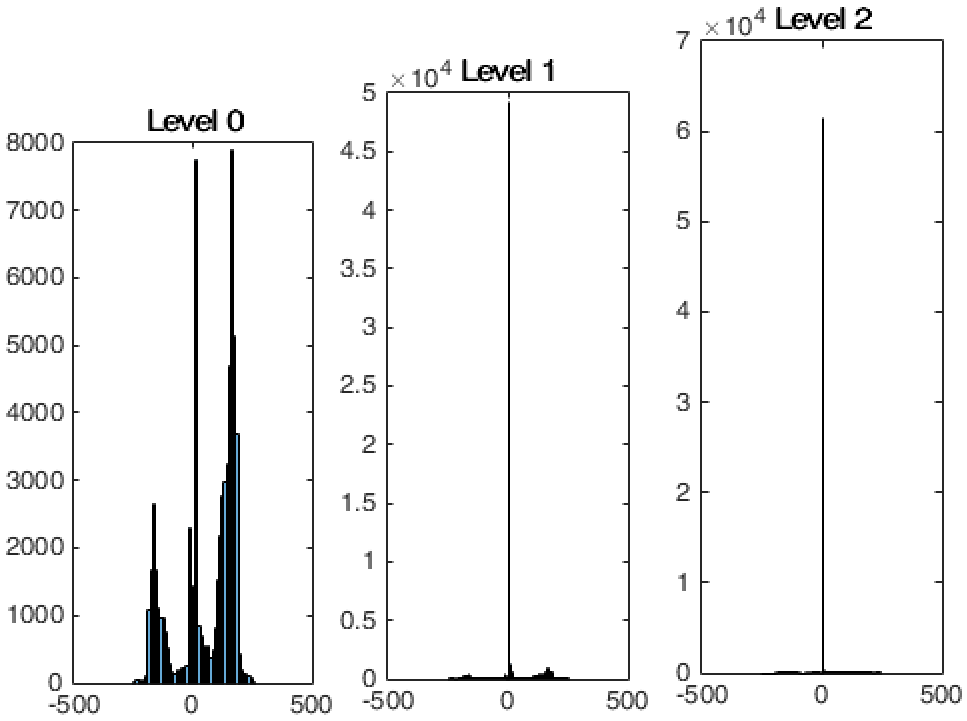
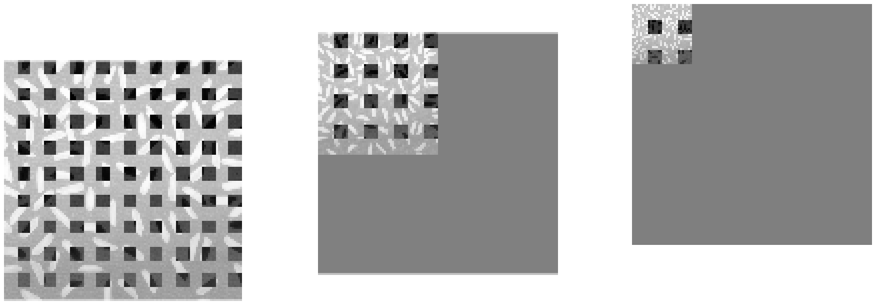


Image 2 Filter # 1 Residual



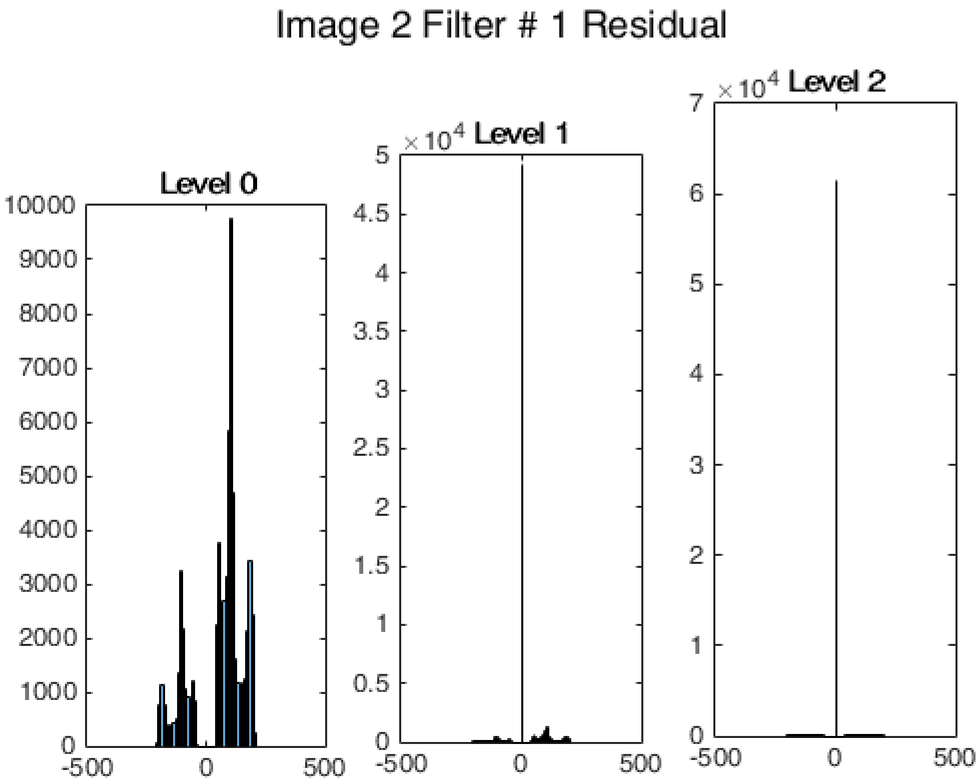


Image 2 Filter # 1 Residual
Level 0
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
-500
0
500
$\times 10^4$ Level 1
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-500
0
500
$\times 10^4$ Level 2
7
6
5
4
3
2
1
0
-500
0
500

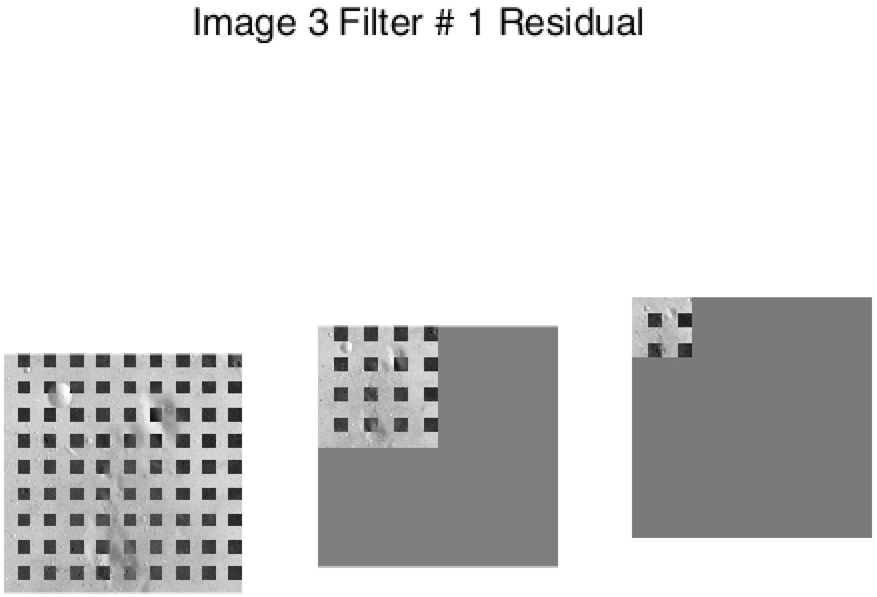


Image 3 Filter # 1 Residual

Image 3 Filter # 1 Residual

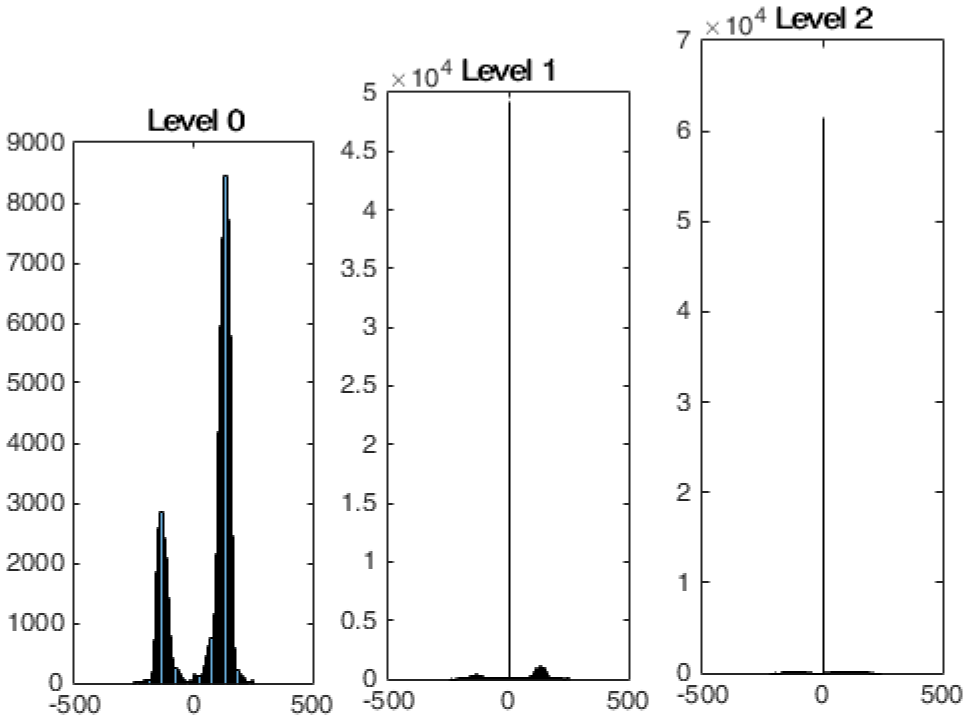
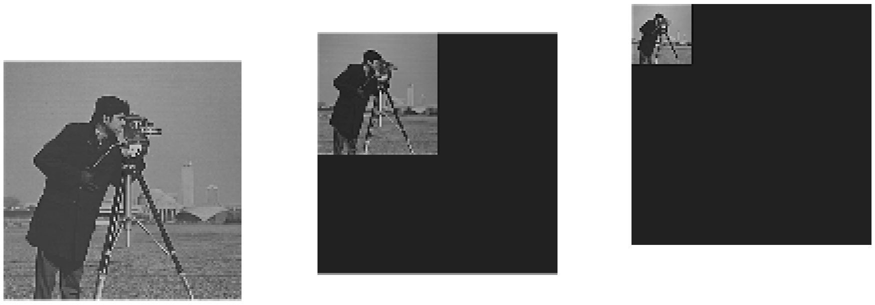


Image 1 Filter # 2 Residual



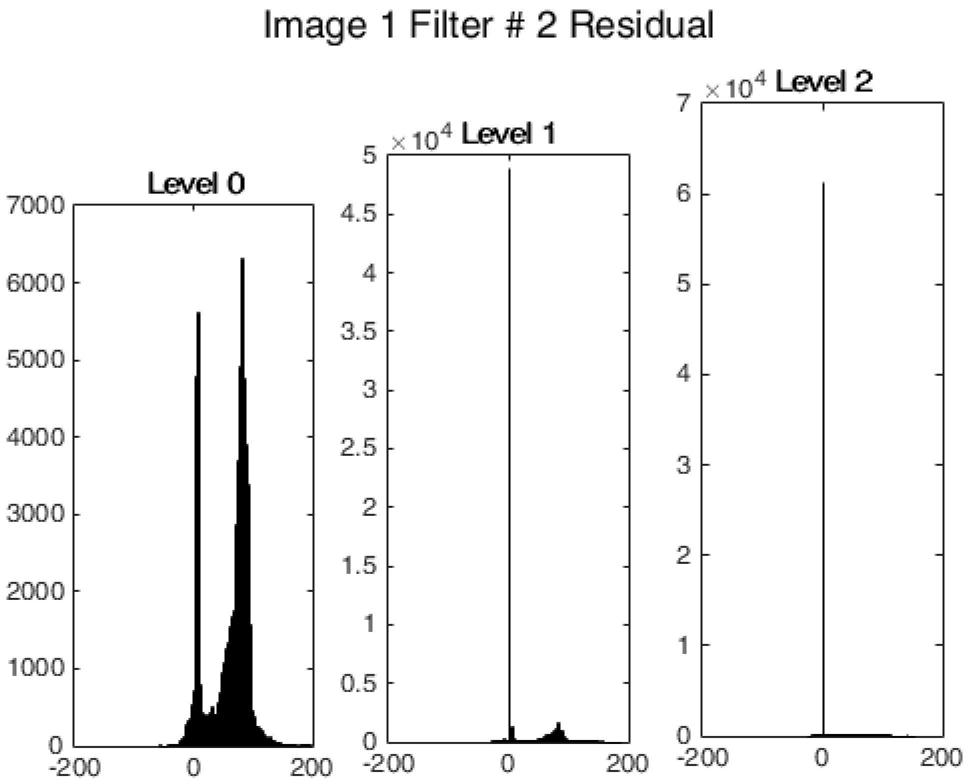


Image 1 Filter # 2 Residual
Level 0
7000
6000
5000
4000
3000
2000
1000
0
-200
0
200
$\times 10^4$ Level 1
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-200
0
200
$\times 10^4$ Level 2
7
6
5
4
3
2
1
0
-200
0
200

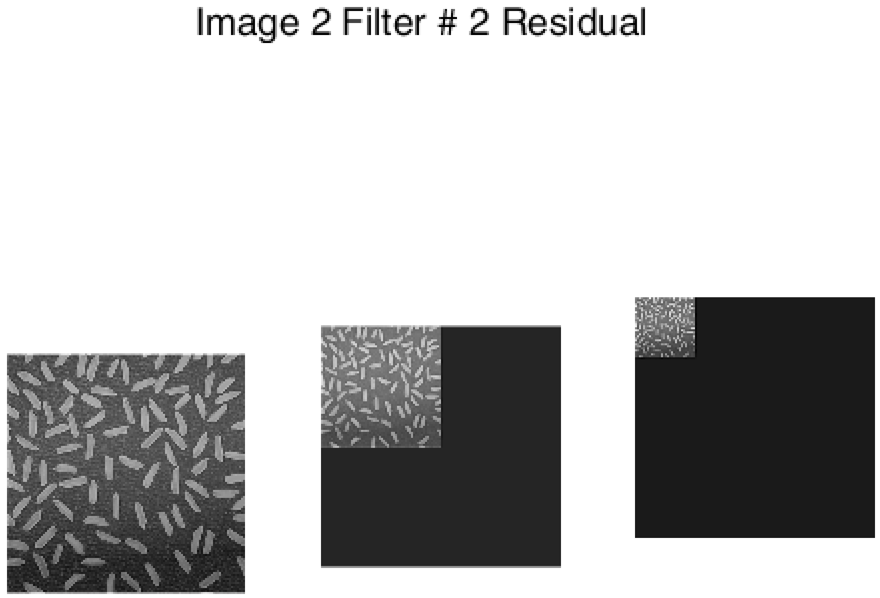


Image 2 Filter # 2 Residual

Image 2 Filter # 2 Residual

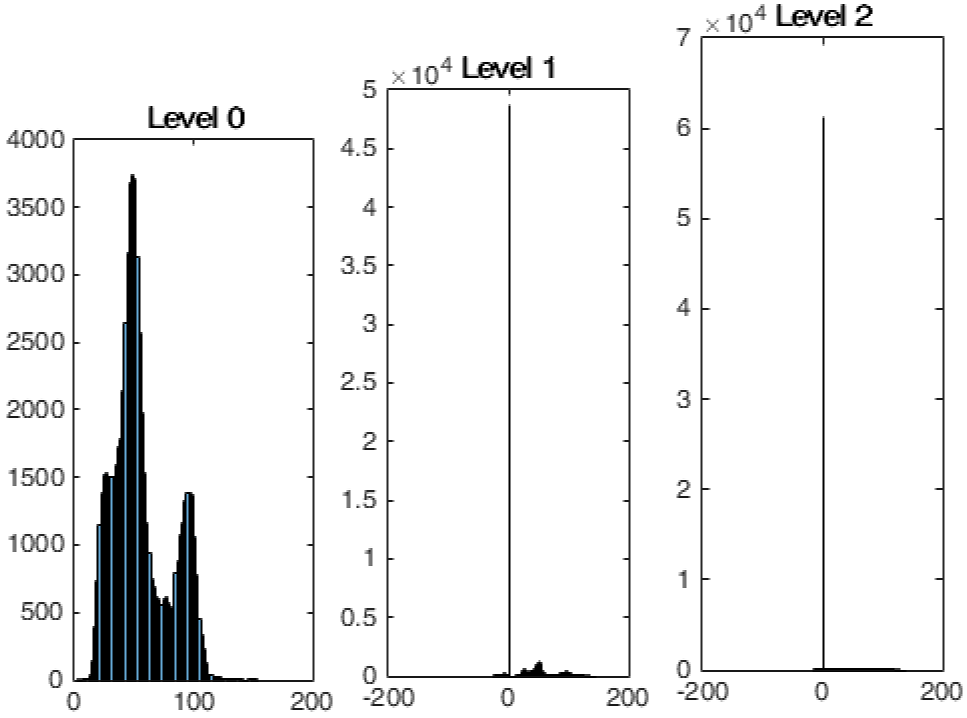


Image 3 Filter # 2 Residual

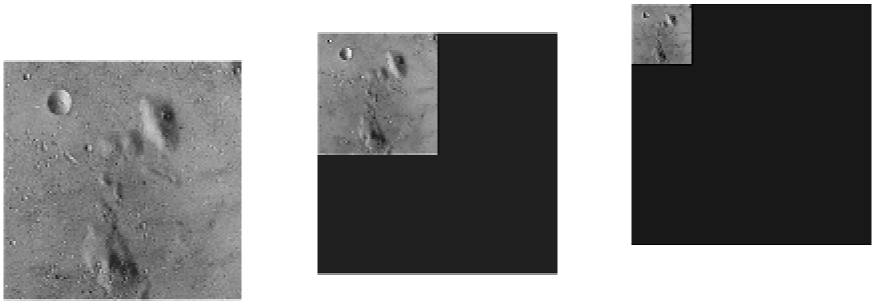


Image 3 Filter # 2 Residual

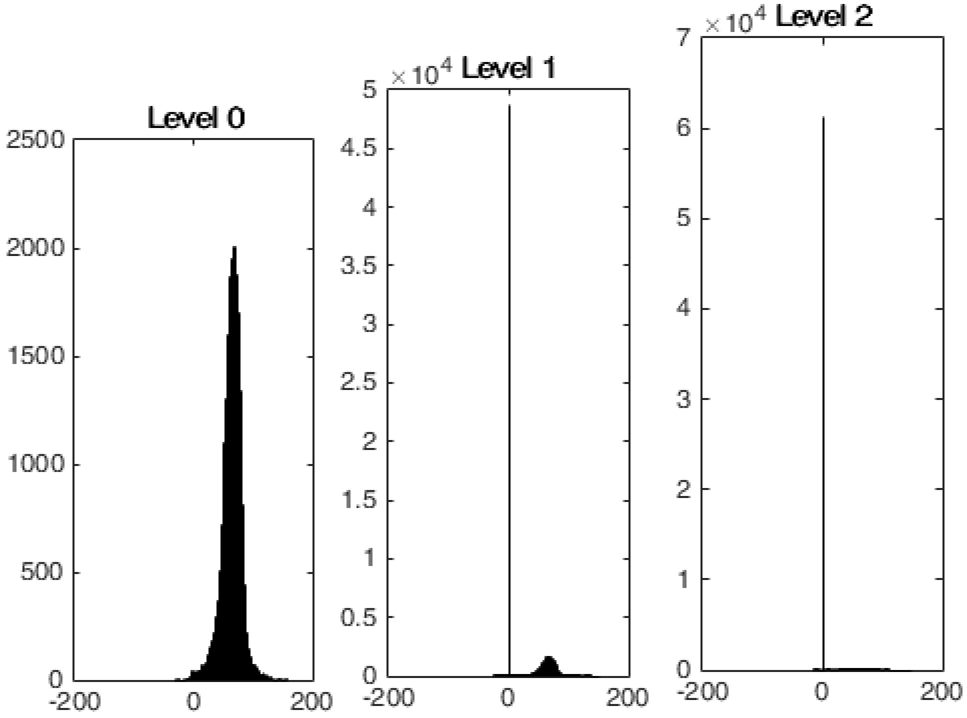


Image 1 Filter # 3 Residual

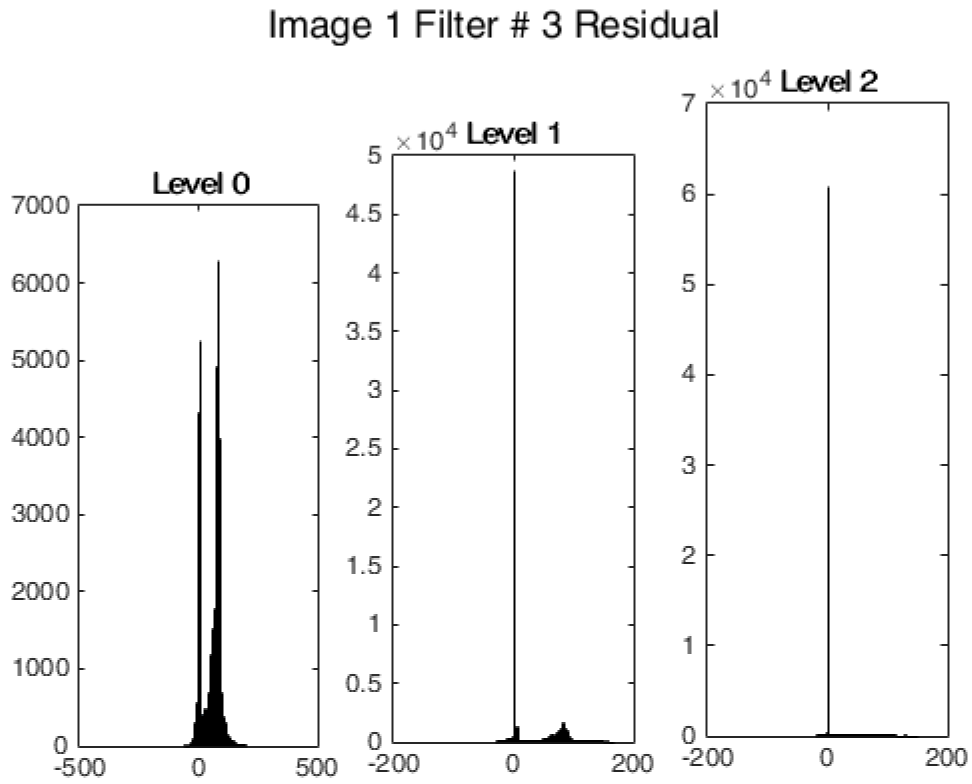


Image 1 Filter # 3 Residual
Level 0
7000
6000
5000
4000
3000
2000
1000
0
-500
0
500
×10^4 Level 1
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-200
0
200
×10^4 Level 2
7
6
5
4
3
2
1
0
-200
0
200

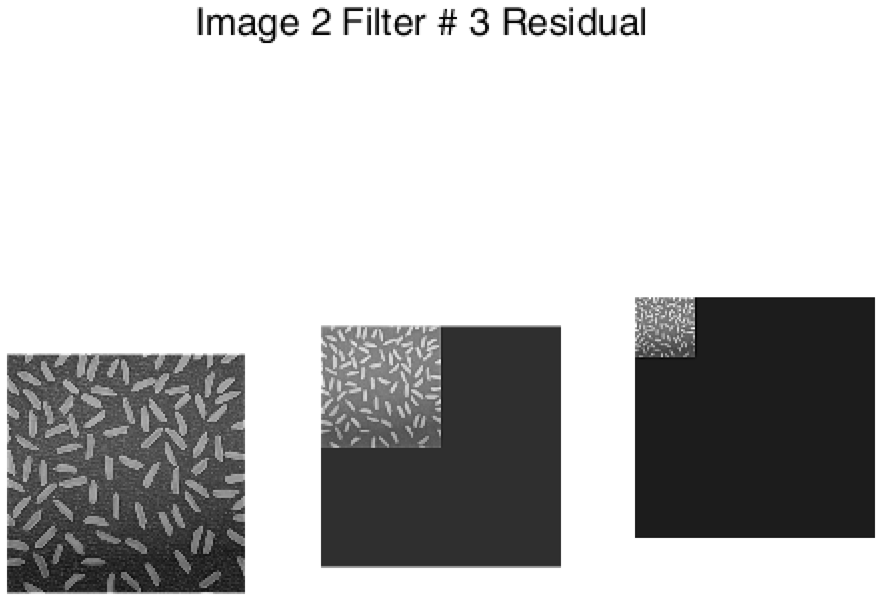


Image 2 Filter # 3 Residual

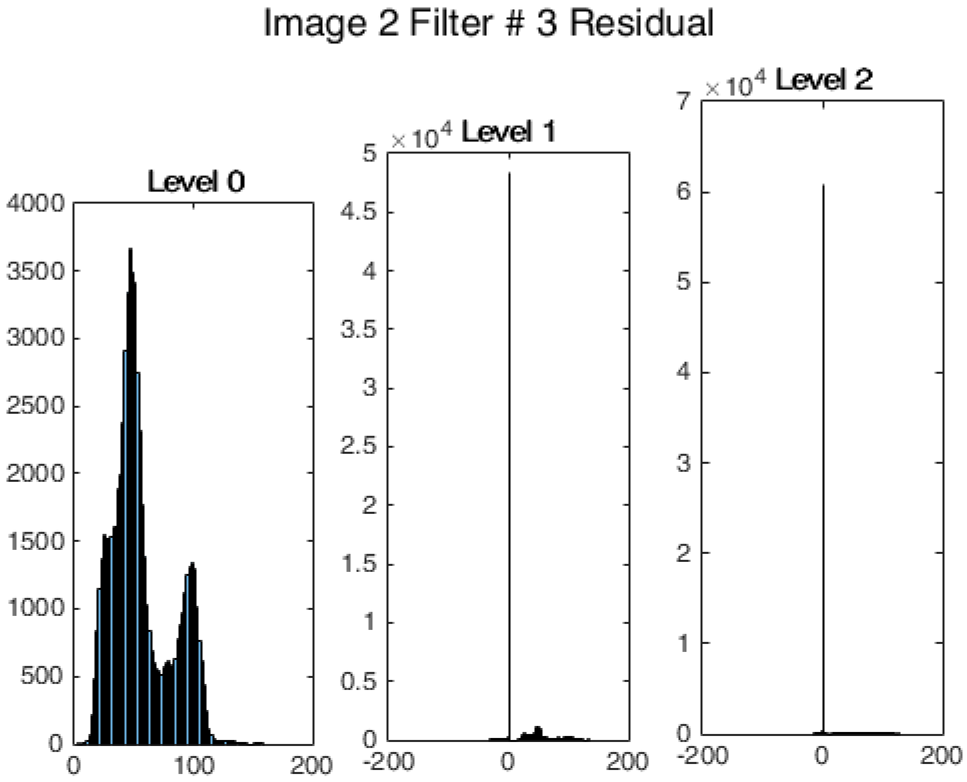


Image 2 Filter # 3 Residual
Level 0
4000
3500
3000
2500
2000
1500
1000
500
0
0
100
200
Level 1
$\times 10^4$
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-200
0
200
Level 2
$\times 10^4$
7
6
5
4
3
2
1
0
-200
0
200

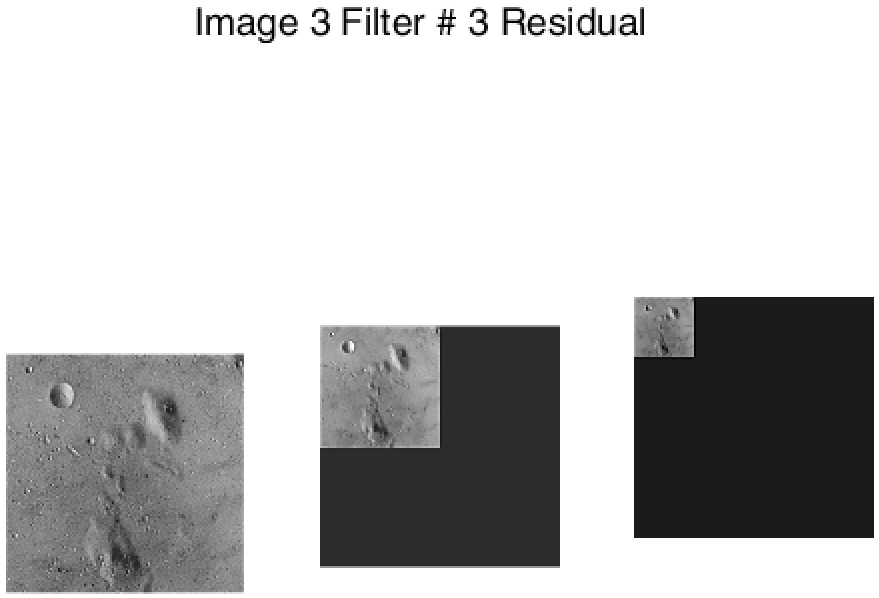


Image 3 Filter # 3 Residual

Image 3 Filter # 3 Residual

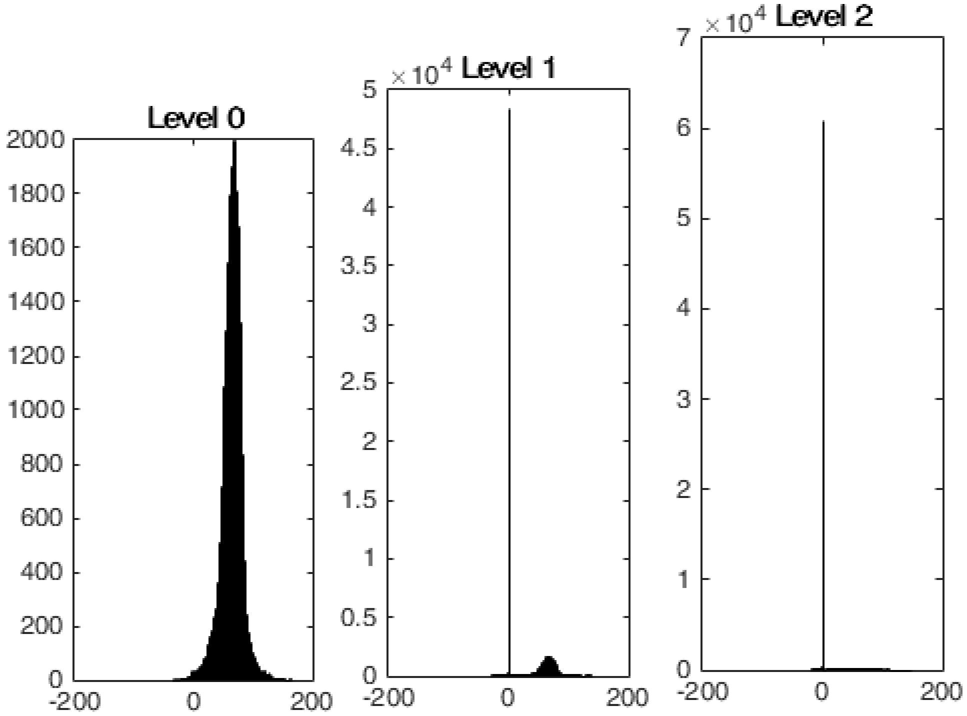


Image 1 Filter # 4 Residual

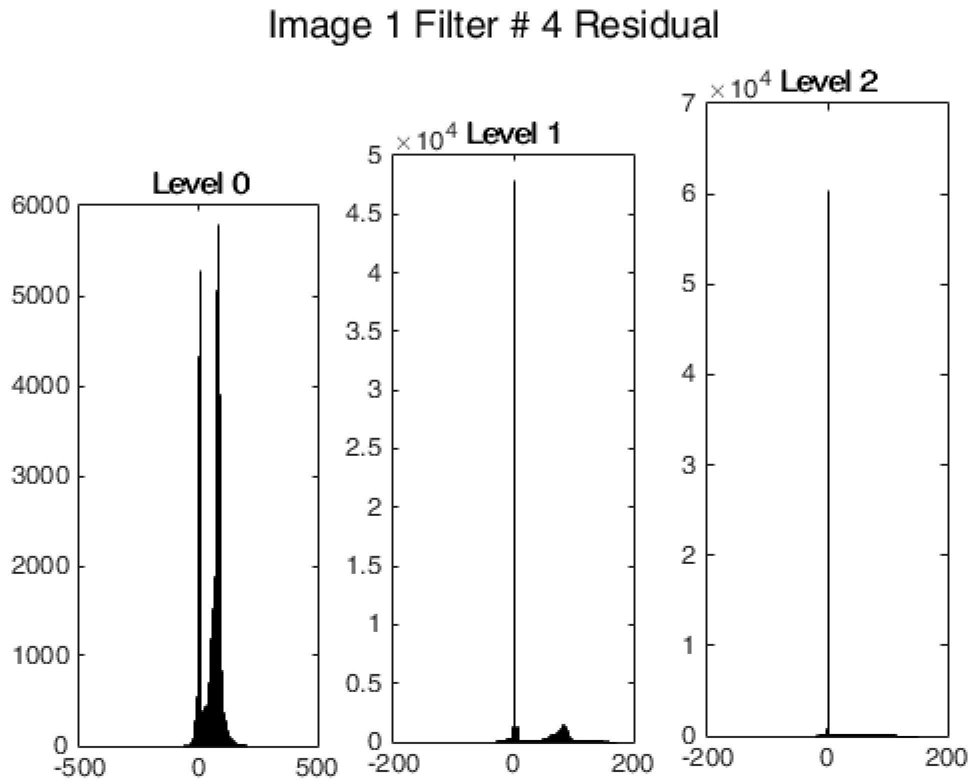


Image 1 Filter # 4 Residual
Level 0
6000
5000
4000
3000
2000
1000
0
-500
0
500
×10^4 Level 1
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-200
0
200
×10^4 Level 2
7
6
5
4
3
2
1
0
-200
0
200

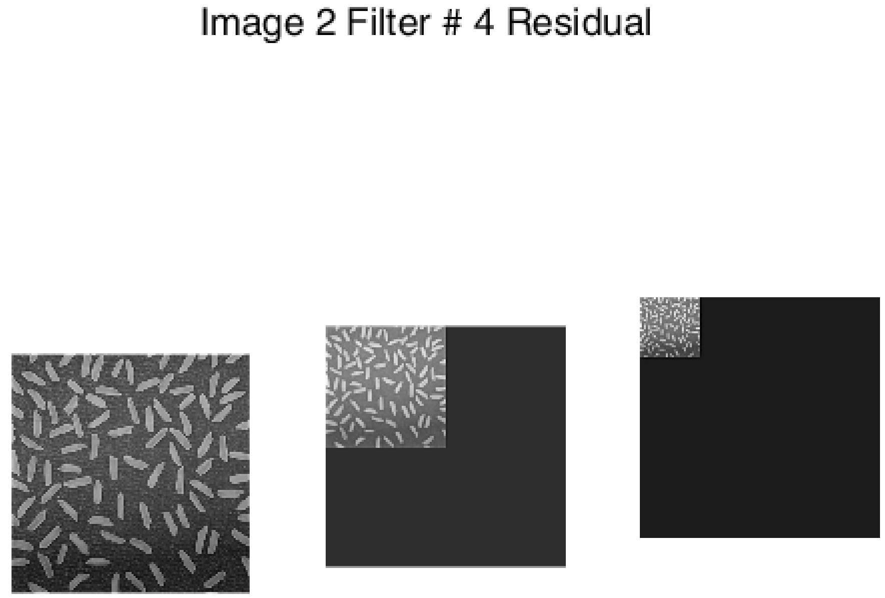


Image 2 Filter # 4 Residual

Image 2 Filter # 4 Residual

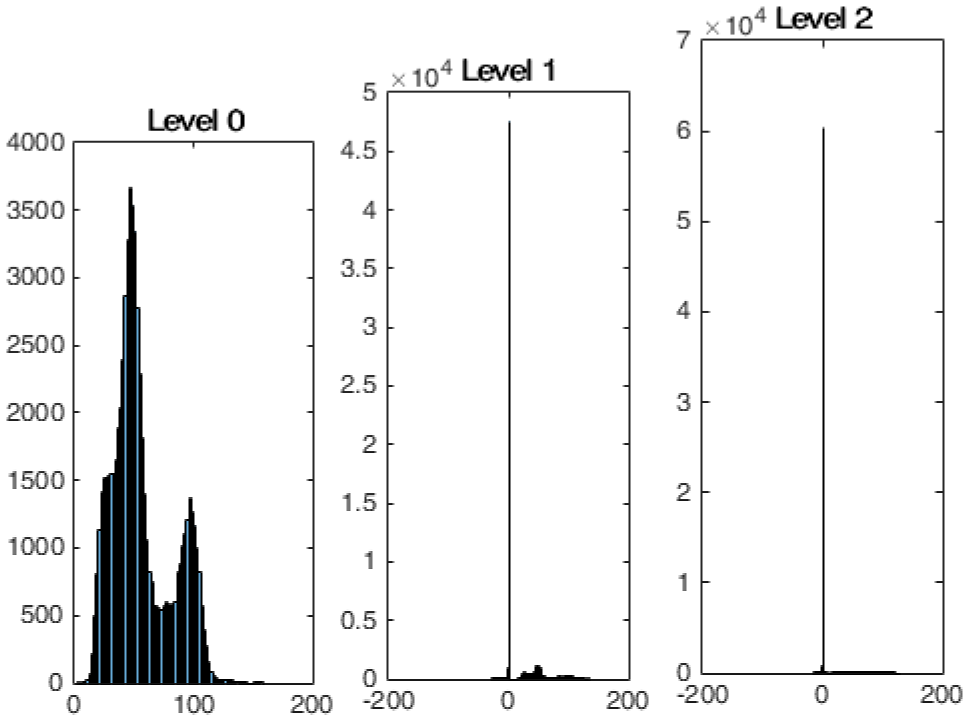
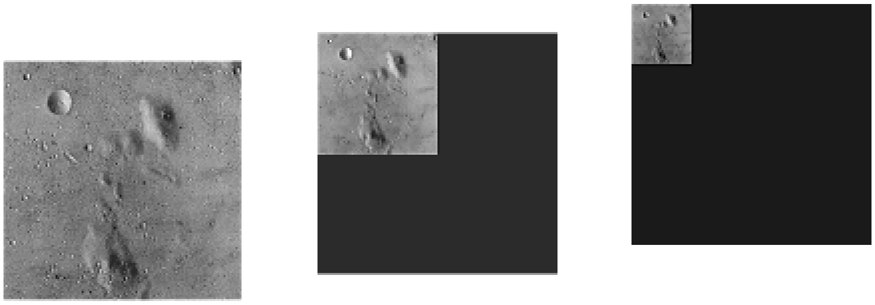


Image 3 Filter # 4 Residual



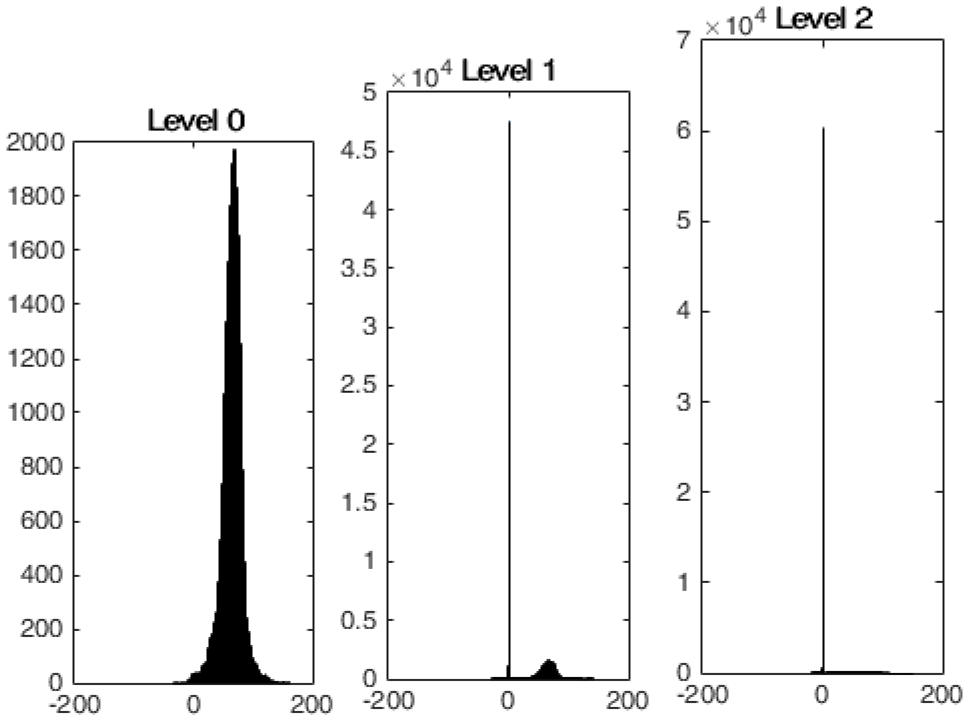


Image 3 Filter # 4 Residual

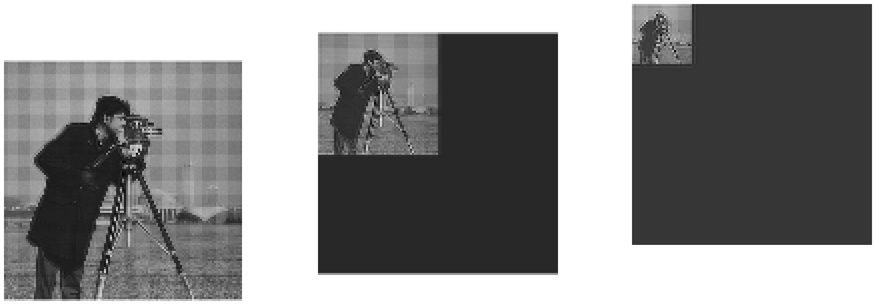


Image 1 Filter # 5 Residual

Image 1 Filter # 5 Residual

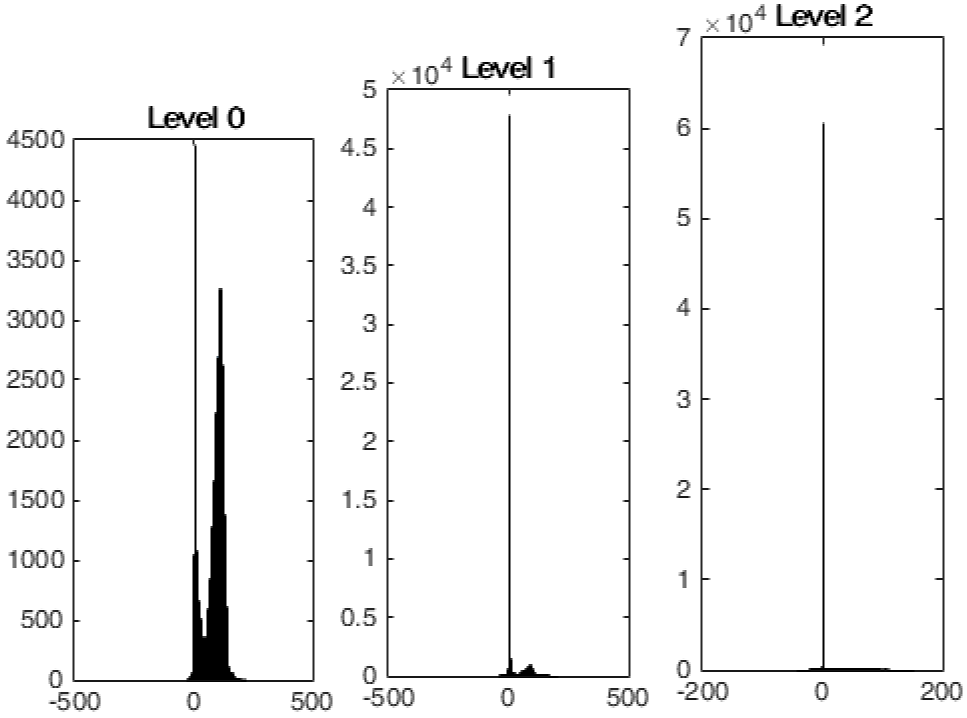


Image 2 Filter # 5 Residual

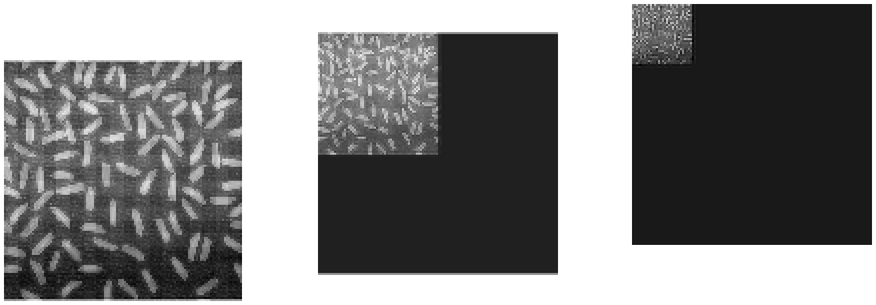


Image 2 Filter # 5 Residual

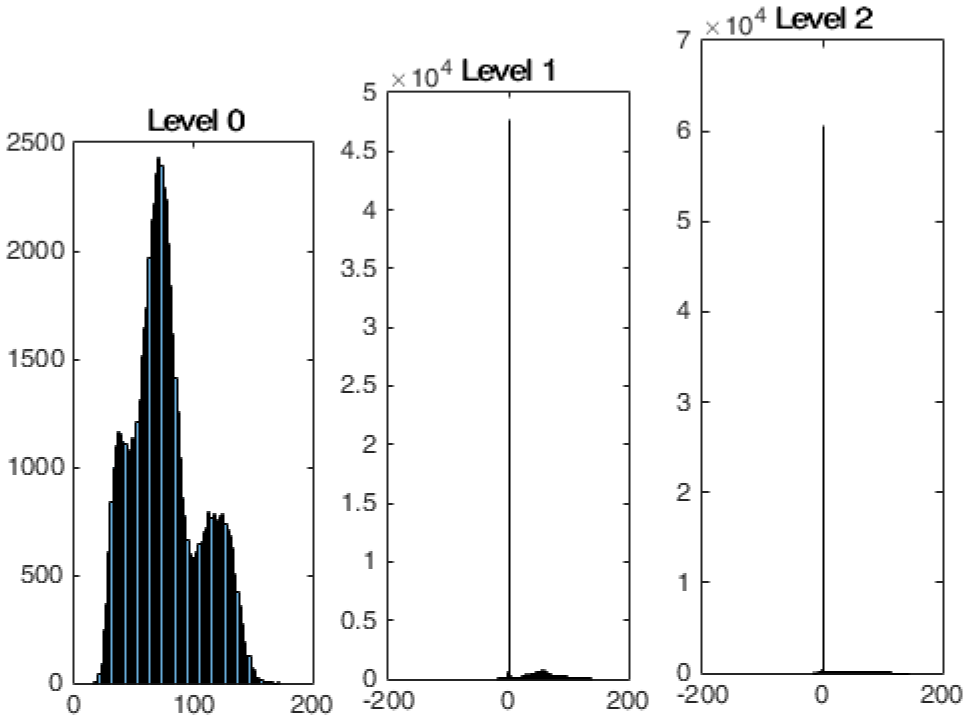
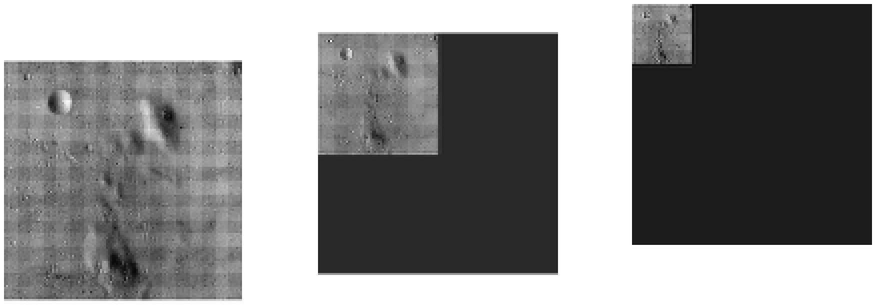
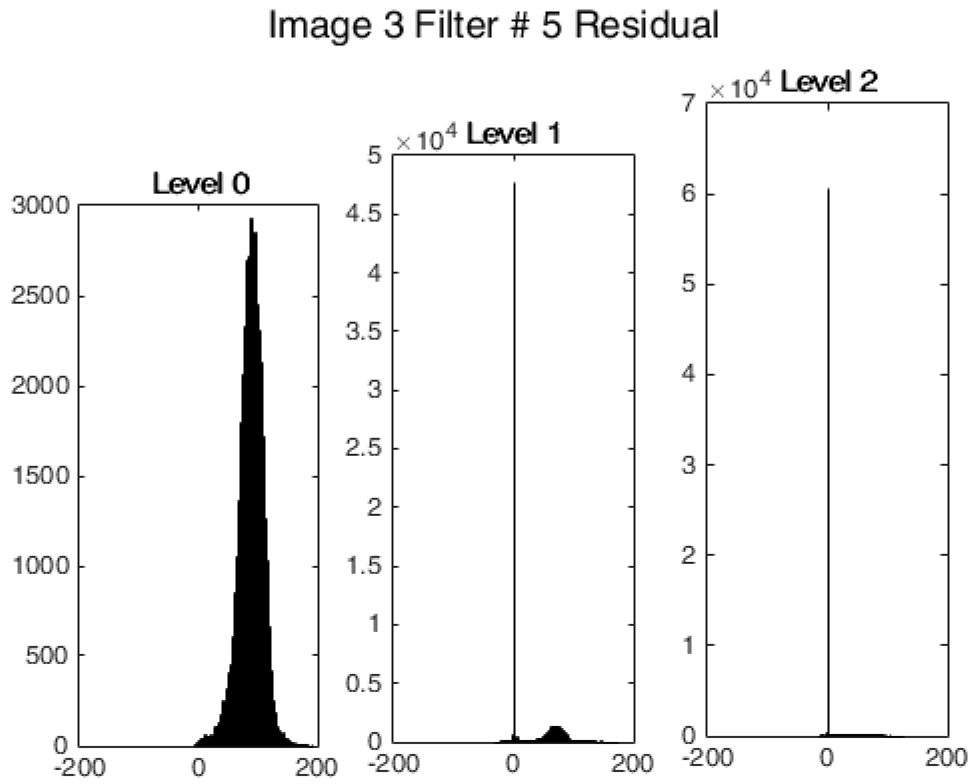


Image 3 Filter # 5 Residual





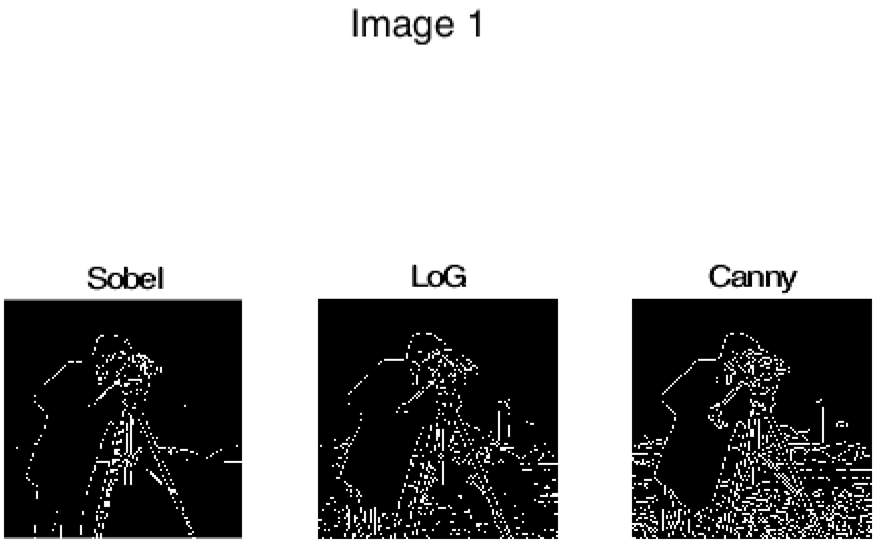
Problem 3

```
[edge1(:,:,1),thresh1] = edge(im(:,:,1),'Sobel');
[edge1(:,:,2),thresh2] = edge(im(:,:,1),'LoG');
[edge1(:,:,3),thresh3] = edge(im(:,:,1),'Canny');
[edge2(:,:,1),thresh4] = edge(im(:,:,2),'Sobel');
[edge2(:,:,2),thresh5] = edge(im(:,:,2),'LoG');
[edge2(:,:,3),thresh6] = edge(im(:,:,2),'Canny');
[edge3(:,:,1),thresh7] = edge(im(:,:,3),'Sobel');
[edge3(:,:,2),thresh8] = edge(im(:,:,3),'LoG');
[edge3(:,:,3),thresh9] = edge(im(:,:,3),'Canny');

fprintf('Image 1 Sobel Thresh: %d\n',thresh1);
fprintf('Image 1 LoG Thresh: %d\n',thresh2);
fprintf('Image 1 Canny Thresh: %d\n',thresh3);
fprintf('Image 2 Sobel Thresh: %d\n',thresh4);
fprintf('Image 2 LoG Thresh: %d\n',thresh5);
fprintf('Image 2 Canny Thresh: %d\n',thresh6);
fprintf('Image 3 Sobel Thresh: %d\n',thresh7);
fprintf('Image 3 LoG Thresh: %d\n',thresh8);
fprintf('Image 3 Canny Thresh: %d\n',thresh9);


edge4(:,:,1) = edge(im(:,:,1),'Sobel',2*thresh1);
edge4(:,:,2) = edge(im(:,:,1),'LoG',2*thresh2);
```

```
edge4(:,:,3) = edge(im(:,:,1),'Canny',2*thresh3);
edge5(:,:,1) = edge(im(:,:,2),'Sobel',2*thresh4);
edge5(:,:,2) = edge(im(:,:,2),'LoG',2*thresh5);
edge5(:,:,3) = edge(im(:,:,2),'Canny',2*thresh6);
edge6(:,:,1) = edge(im(:,:,3),'Sobel',2*thresh7);
edge6(:,:,2) = edge(im(:,:,3),'LoG',2*thresh8);
edge6(:,:,3) = edge(im(:,:,3),'Canny',2*thresh9);

edge7(:,:,1) = edge(im(:,:,1),'Sobel',thresh1/2);
edge7(:,:,2) = edge(im(:,:,1),'LoG',thresh2/2);
edge7(:,:,3) = edge(im(:,:,1),'Canny',thresh3/2);
edge8(:,:,1) = edge(im(:,:,2),'Sobel',thresh4/2);
edge8(:,:,2) = edge(im(:,:,2),'LoG',thresh5/2);
edge8(:,:,3) = edge(im(:,:,2),'Canny',thresh6/2);
edge9(:,:,1) = edge(im(:,:,3),'Sobel',thresh7/2);
edge9(:,:,2) = edge(im(:,:,3),'LoG',thresh8/2);
edge9(:,:,3) = edge(im(:,:,3),'Canny',thresh9/2);

figure;
subplot(1,3,1), imshow(edge1(:,:,1),[]);
title('Sobel');
subplot(1,3,2), imshow(edge1(:,:,2),[]);
title('LoG');
subplot(1,3,3), imshow(edge1(:,:,3),[]);
title('Canny');
suptitle('Image 1');

figure;
subplot(1,3,1), imshow(edge4(:,:,1),[]);
title('Sobel');
subplot(1,3,2), imshow(edge4(:,:,2),[]);
title('LoG');
subplot(1,3,3), imshow(edge4(:,:,3),[]);
title('Canny');
suptitle('Image 1, 2*thresh');

figure;
subplot(1,3,1), imshow(edge7(:,:,1),[]);
title('Sobel');
subplot(1,3,2), imshow(edge7(:,:,2),[]);
title('LoG');
subplot(1,3,3), imshow(edge7(:,:,3),[]);
title('Canny');
suptitle('Image 1, thresh/2');

figure;
subplot(1,3,1), imshow(edge2(:,:,1),[]);
title('Sobel');
subplot(1,3,2), imshow(edge2(:,:,2),[]);
title('LoG');
subplot(1,3,3), imshow(edge2(:,:,3),[]);
title('Canny');
suptitle('Image 2');
```

```
figure;
subplot(1,3,1), imshow(edge5(:,:,1),[]);
title('Sobel');
subplot(1,3,2), imshow(edge5(:,:,2),[]);
title('LoG');
subplot(1,3,3), imshow(edge5(:,:,3),[]);
title('Canny');
suptitle('Image 2, 2*thresh');

figure;
subplot(1,3,1), imshow(edge8(:,:,1),[]);
title('Sobel');
subplot(1,3,2), imshow(edge8(:,:,2),[]);
title('LoG');
subplot(1,3,3), imshow(edge8(:,:,3),[]);
title('Canny');
suptitle('Image 2, thresh/2');

figure;
subplot(1,3,1), imshow(edge3(:,:,1),[]);
title('Sobel');
subplot(1,3,2), imshow(edge3(:,:,2),[]);
title('LoG');
subplot(1,3,3), imshow(edge3(:,:,3),[]);
title('Canny');
suptitle('Image 3');

figure;
subplot(1,3,1), imshow(edge6(:,:,1),[]);
title('Sobel');
subplot(1,3,2), imshow(edge6(:,:,2),[]);
title('LoG');
subplot(1,3,3), imshow(edge6(:,:,3),[]);
title('Canny');
suptitle('Image 3, 2*thresh');

figure;
subplot(1,3,1), imshow(edge9(:,:,1),[]);
title('Sobel');
subplot(1,3,2), imshow(edge9(:,:,2),[]);
title('LoG');
subplot(1,3,3), imshow(edge9(:,:,3),[]);
title('Canny');
suptitle('Image 3, thresh/2');

disp('Increasing the threshold cuts out more edges, decreasing the
 threshold increases the number of edges allowed through. In general
 performance is the same for each filter type across thresholds.');
```

```
Image 1 Sobel Thresh: 1.433227e-01
Image 1 LoG Thresh: 5.093515e-03
Image 1 Canny Thresh: 3.125000e-02
Image 1 Canny Thresh: 7.812500e-02
Image 2 Sobel Thresh: 1.161891e-01
Image 2 LoG Thresh: 7.456524e-03
Image 2 Canny Thresh: 1.750000e-01
Image 2 Canny Thresh: 4.375000e-01
Image 3 Sobel Thresh: 6.744113e-02
Image 3 LoG Thresh: 2.594287e-03
Image 3 Canny Thresh: 3.750000e-02
Image 3 Canny Thresh: 9.375000e-02
Increasing the threshold cuts out more edges, decreasing the threshold
 increases the number of edges allowed through. In general performance
 is the same for each filter type across thresholds.
```



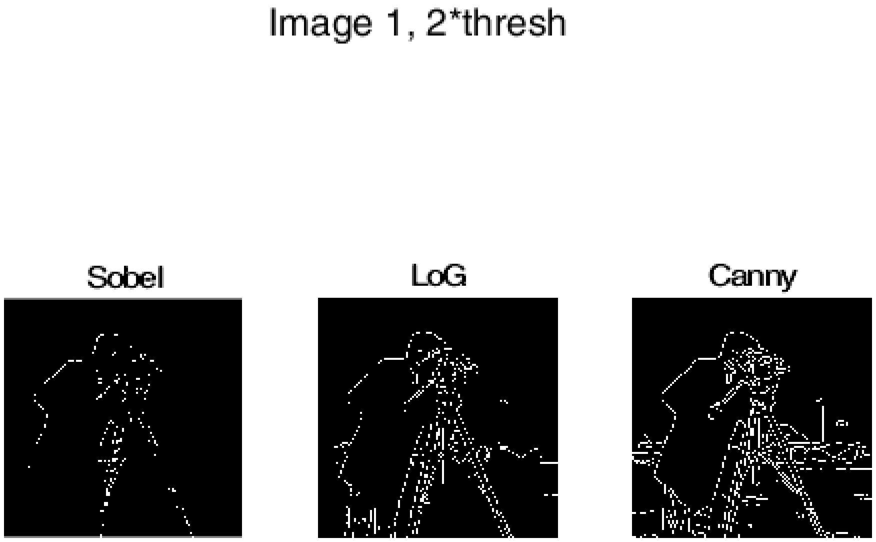


Image 1, 2*thresh
Sobel
LoG
Canny

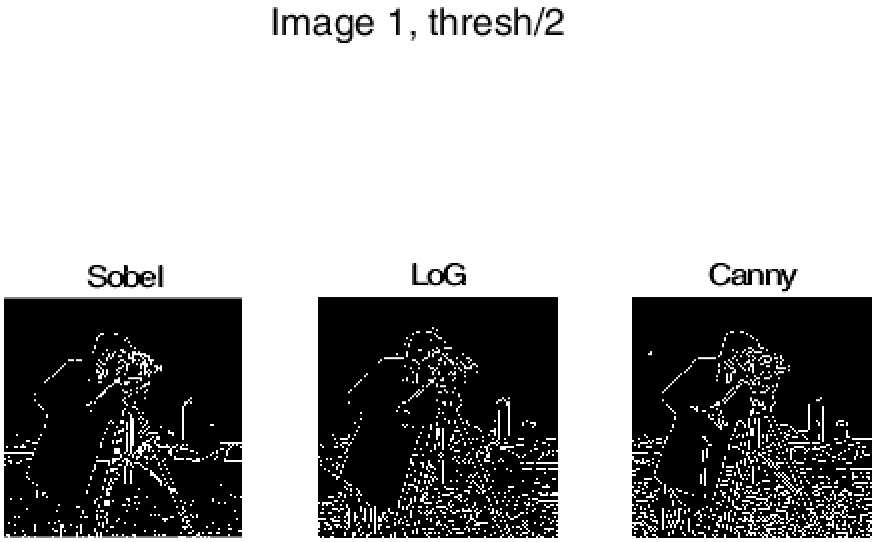


Image 1, thresh/2
Sobel
LoG
Canny

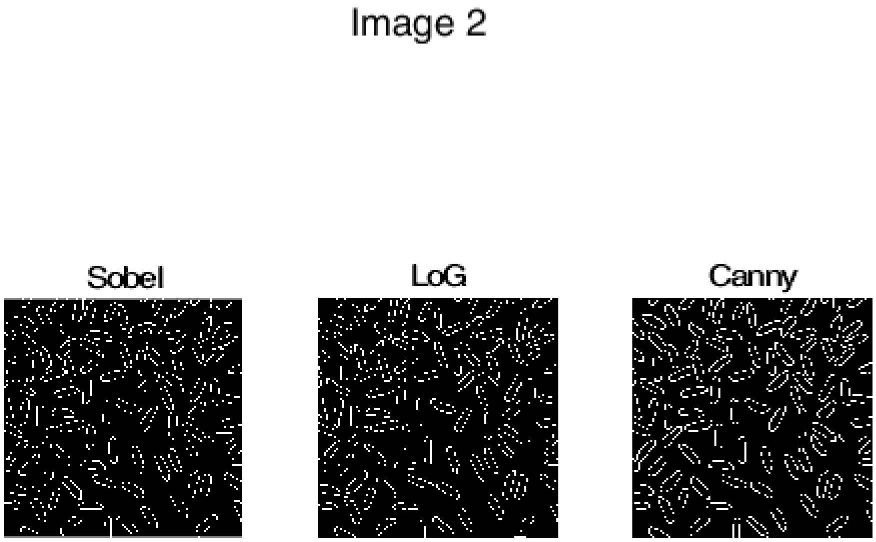


Image 2
Sobel
LoG
Canny

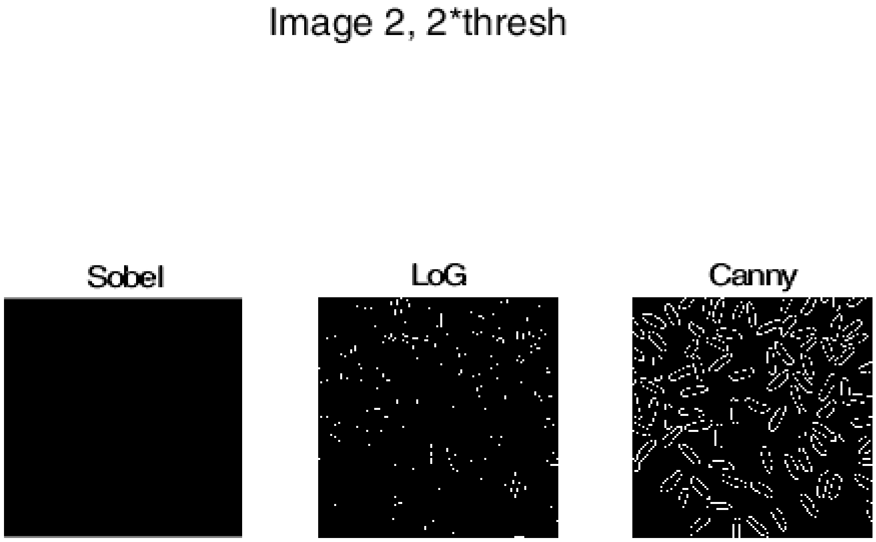


Image 2, 2*thresh
Sobel
LoG
Canny

Image 2, thresh/2

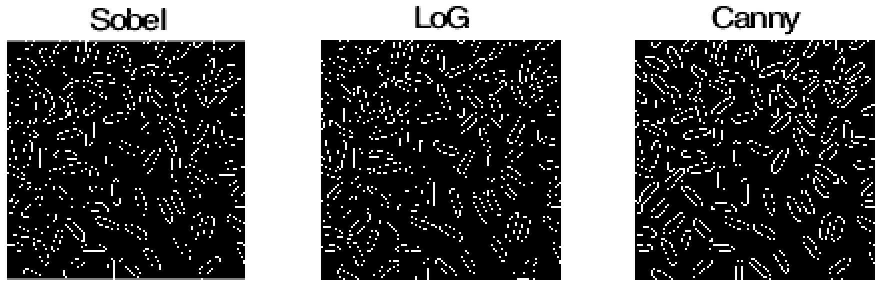
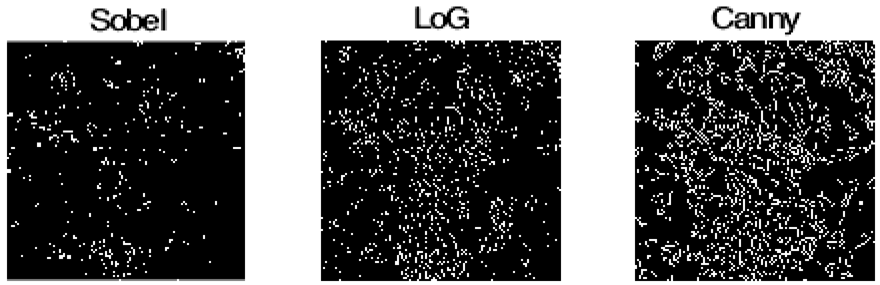


Image 3



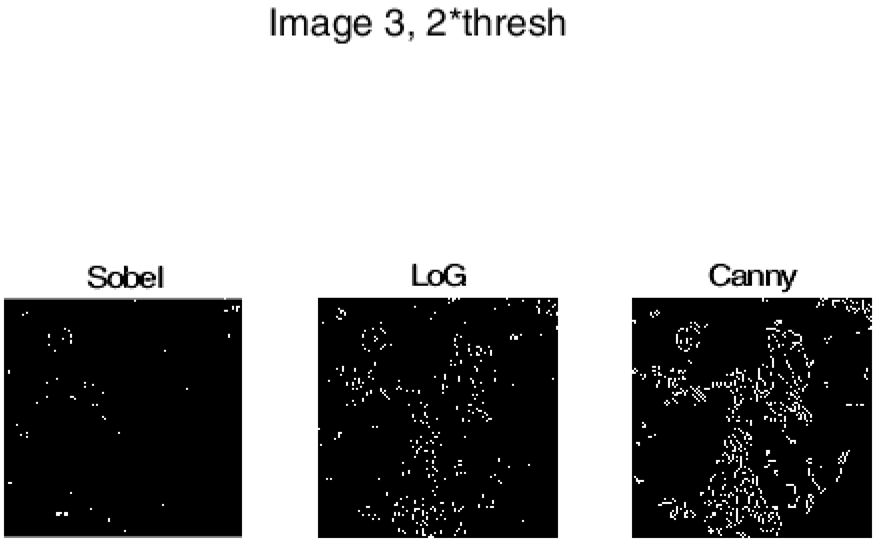


Image 3, 2*thresh
Sobel
LoG
Canny

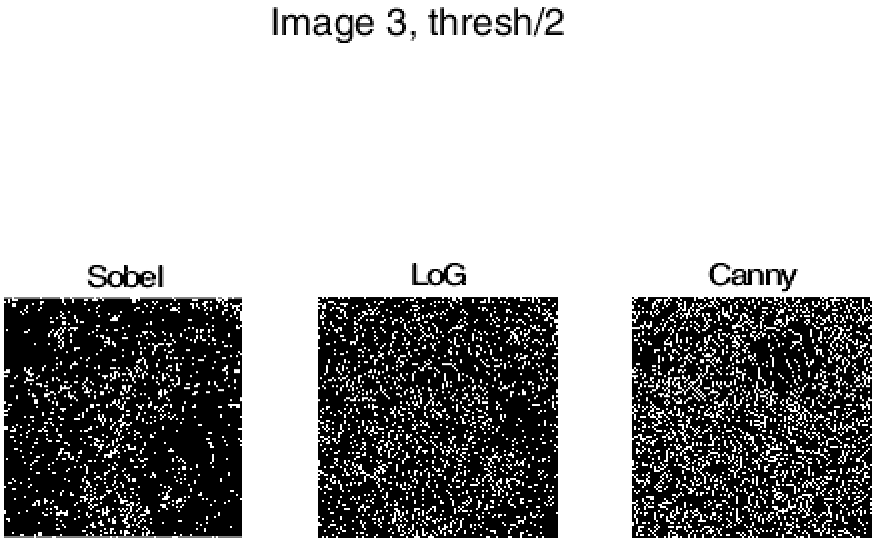


Image 3, thresh/2
Sobel
LoG
Canny

Published with MATLAB® R2015a

```
function res = uscale(im, h, lev)

[nh,mh] = size(h);
if nh == 1
    fil = 2*(h'*h);
else
    fil = 2*h;
end

[om, on] = size(im);
ex = zeros(om,on);

m = ceil(om/(lev+1));
n = ceil(on/(lev+1));

ex(1:2:2*m,1:2:2*n) = im(1:m,1:n);

res = filter2(fil,ex);
```

```
Error using uscale (line 3)
Not enough input arguments.
```

Published with MATLAB® R2015a

```
function res = uscale2(im, h, lev)

for i = 1:lev
    res(:,:,i) = uscale(im(:,:,i),h,i);
end
```

```
Error using uscale2 (line 3)
Not enough input arguments.
```

Published with MATLAB® R2015a

```
function out = dscale(im, h)

[n,m] = size(im);
out = zeros(n,m);

[nh,mh] = size(h);
if nh == 1
    fil = h'*h;
else
    fil = h;
end

ex = filter2(fil,im);
ex1 = ex(1:2:n,1:2:m);

fm = floor(m/2);
fn = floor(n/2);

out(1:fm,1:fn) = ex1;
```

```
Error using dscale (line 3)
Not enough input arguments.
```

Published with MATLAB® R2015a

```
function res = dscale2(im,h,lev)

for i = 1:lev
    if i == 1
        prev = im;
    else
        prev = res(:,:,i-1);
    end

    res(:,:,i) = dscale(prev,h);
end
```

```
Error using dscale2 (line 3)
Not enough input arguments.
```

Published with MATLAB® R2015a