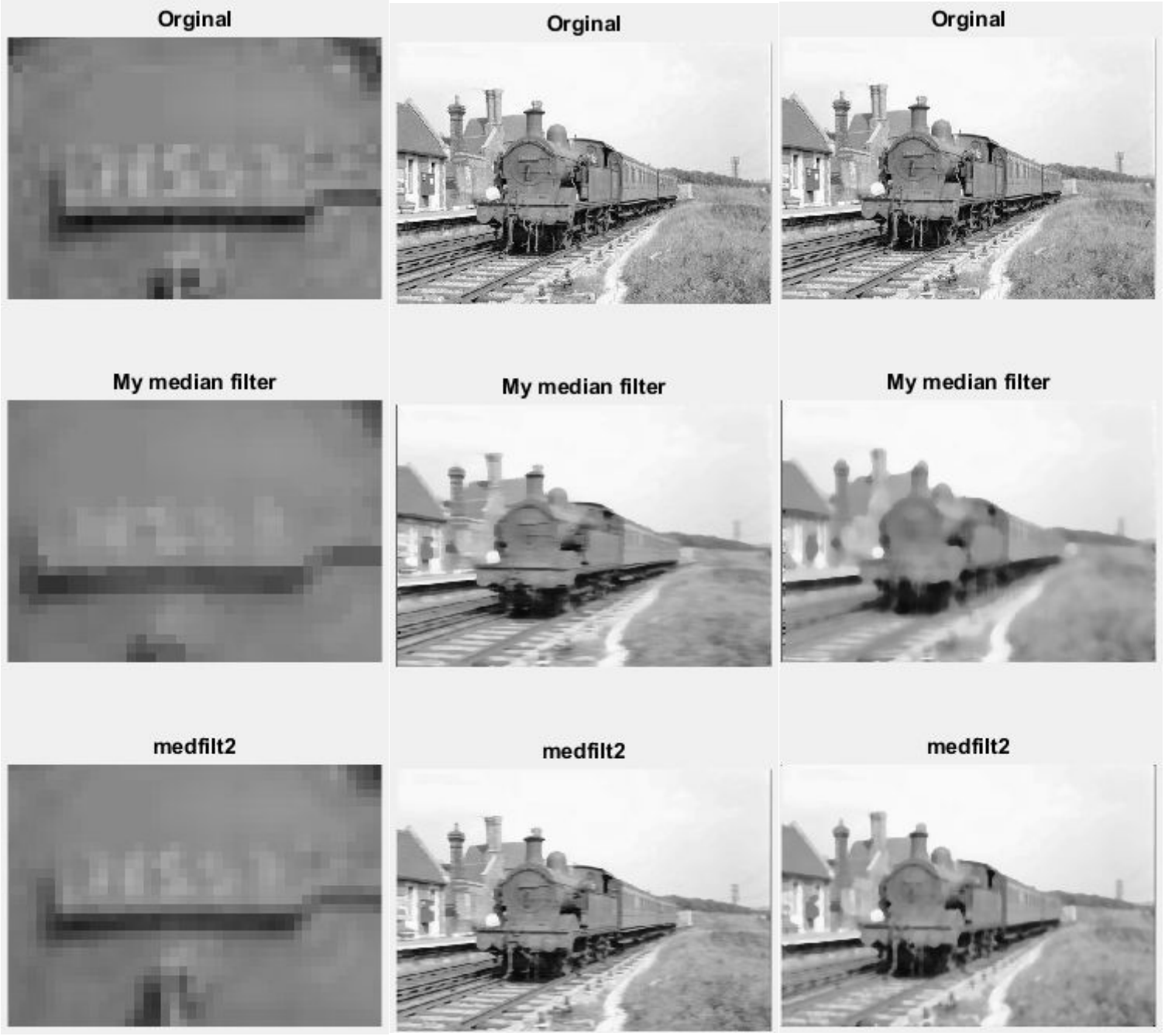
24.11.2020

Robot Vision

RV ex4 Non-linear image filtering

Mateusz Warmuz

2. Implement median filter algorithm. The only input data are:
• image I of size m×n,
• vector [a,b] of mask size, a and b are odd numbers.
Test your program. You can compare the obtained results with the results obtained using MATLAB imfilt2 instruction. In your implementation, you can not use median or medfilt2 instruction, but you can use the sort instruction.



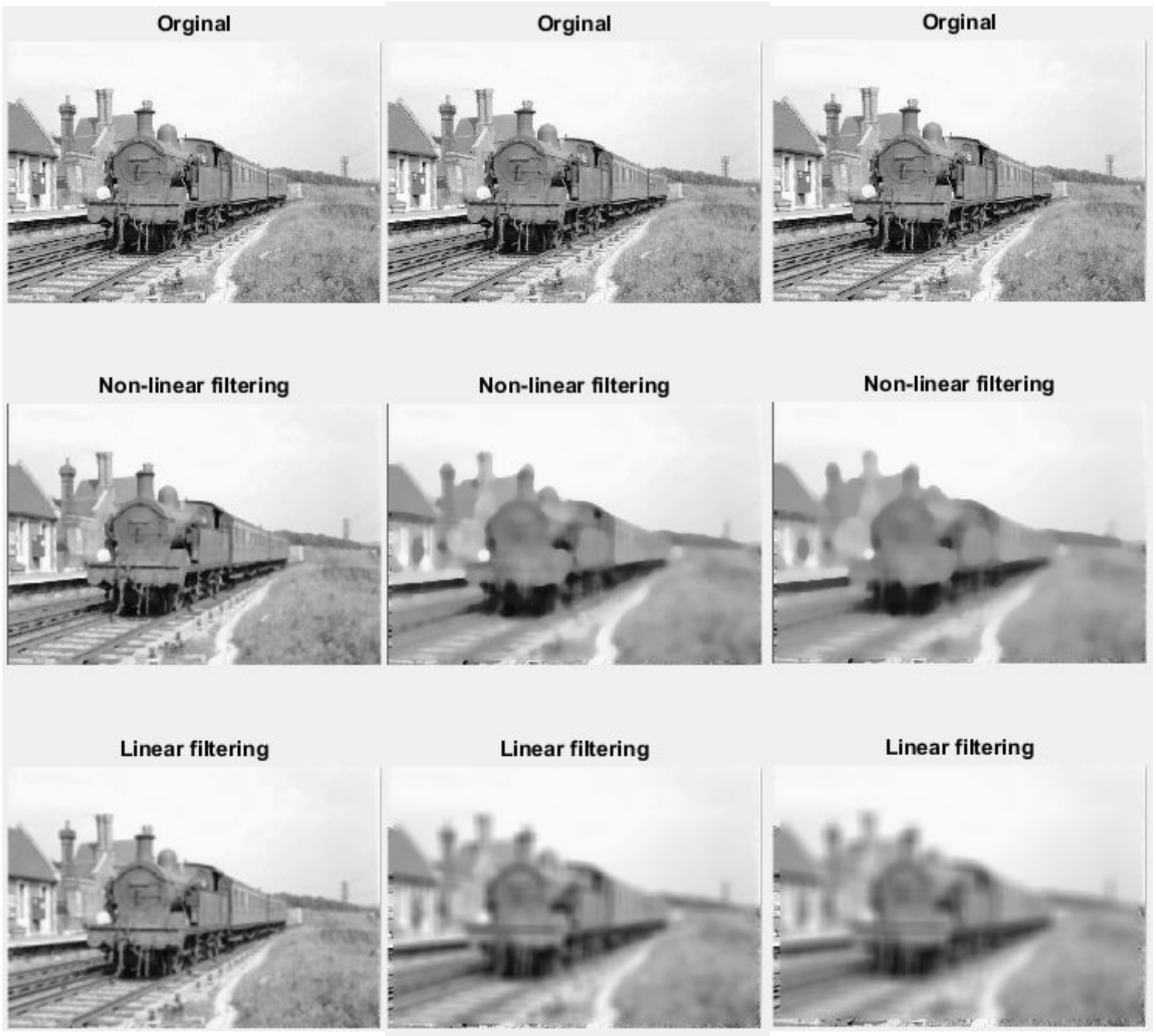
The masks are (3,3), (3,15) and (15,15) respectively. We can see that for small mask size the effect is very weak. Adding a larger mask with rectangular shape where horizontal size is significantly larger. We can notice blurring in horizontal orientation. In the third image we can see the biggest blur due to choosing the biggest size of mask.

Code:

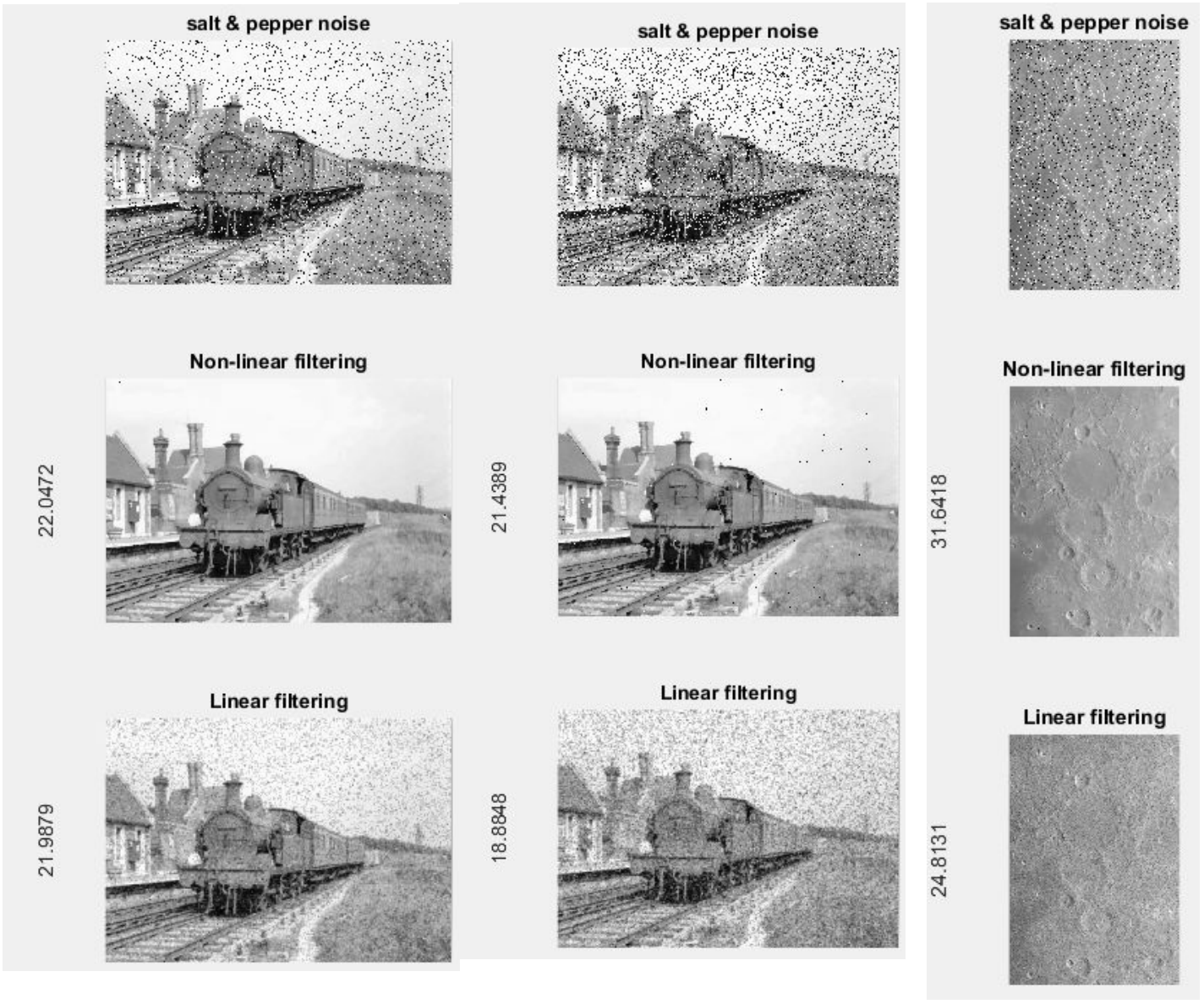
```
photo=imread('test3.jpg');
A = rgb2gray(photo);


h = ones(15,15)/200;

s1 = size(h,1)/2+0.5; %row
s2 = size(h,2)/2+0.5; %column



for row = (s1):size(A,1)-(s1)
   for col = (s2):size(A,2)-(s2)
      B = A((row-s1+1):(row+s1-1),(col-s2+1):(col+s2-1));
      C = sort(reshape(B,1,[]));
      median = C(size(C,2)/2+0.5);
      D(row,col) = median;
   end
end

subplot(3,1,1);
imshow(A);
title('Orginal');
subplot(3,1,2);
imshow(D);
title('My median filter');
J = medfilt2(A, [s1, s2]);
subplot(3,1,3);
imshow(J);
title('medfilt2');
```

24.11.2020

3. For impulsive noise compare noise reduction results obtained from median filter and linear filter. Use PSNR for results evaluation. Use imnoise instruction for contamination of images.



Comparing linear filtering to non-linear we can notice some difference. In my opinion median is more blurry than linear It probably caused by that pixel in a neighborhood will not significantly affect the median value.



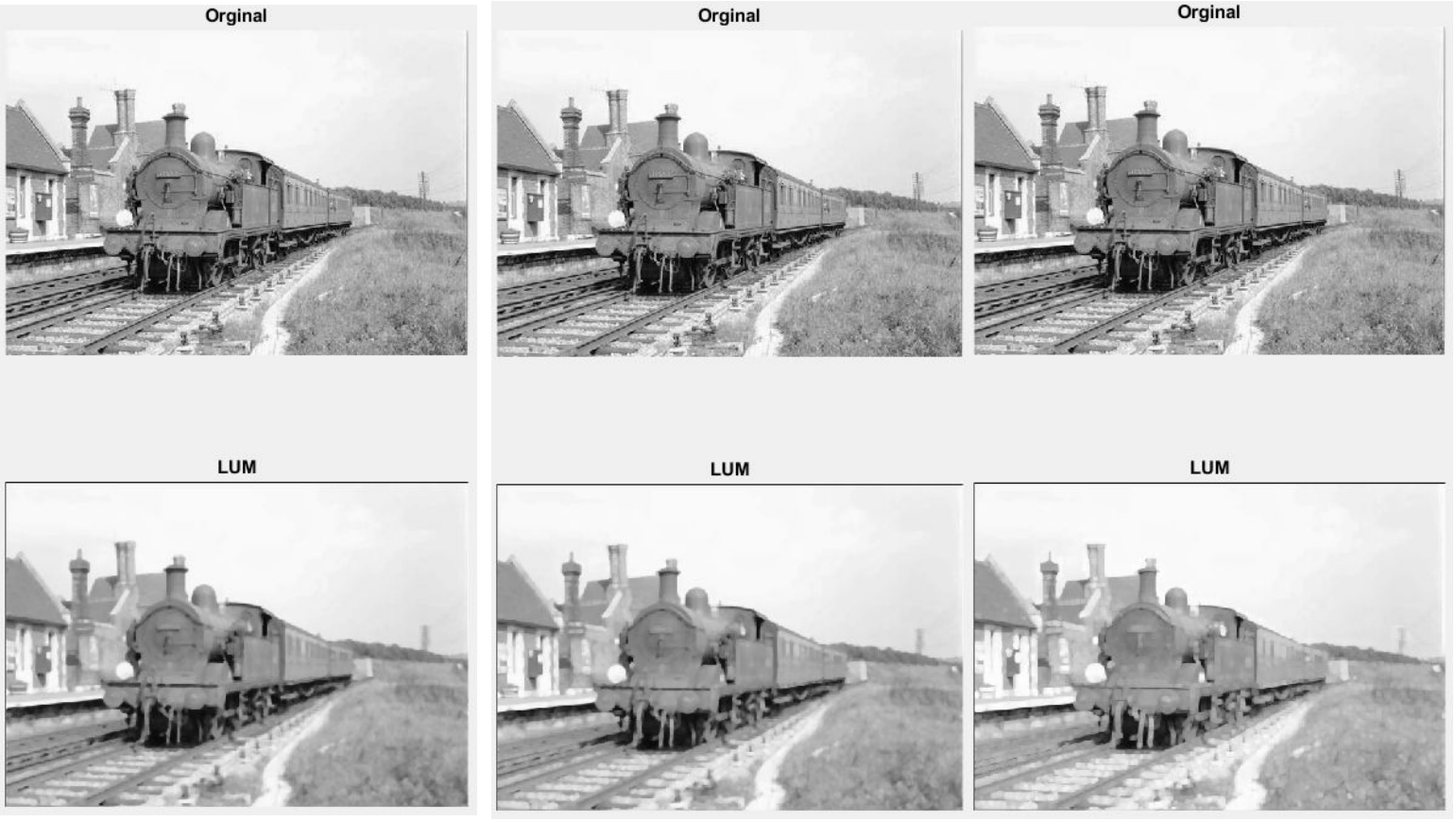
We can observe that nonlinear filters work better than linear. It reduces salt & pepper noise which makes that image is a more clear image than in linear filtering.

Code:

```
photo=imread('test3.jpg');
A = rgb2gray(photo);
grey=rgb2gray(photo);
grey2=rgb2gray(photo);
grey3=rgb2gray(photo);
grey4=rgb2gray(photo);

h = ones(3,3)/25;

s1 = size(h,1)/2+0.5; %row
s2 = size(h,2)/2+0.5; %column

F = imnoise(grey3,'salt & pepper',0.2);

point = size(h,1)/2-0.5;

for row = (s1):size(F,1)-(s1)
   for col = (s2):size(F,2)-(s2)
      B = F((row-s1+1):(row+s1-1),(col-s2+1):(col+s2-1));
      C = sort(reshape(B,1,[]));
      median = C(size(C,2)/2+0.5);
      D(row-1,col-1) = median;
   end
end
E = padarray(D,[1 1],214);

for row = (point+1):size(F,1)-(point+1)
   for col = (point+1):size(F,2)-(point+1)
      grey2(row,col)=conv2(F((row-point):(row+point),(col-point):(col+point)),h,
'valid')/sum(h,'all');
   end
end

A(1,:) = [];
A(:,1) = [];

peaksnr1 = psnr(E,A);
peaksnr2 = psnr(grey2,grey4);

subplot(3,1,1);
imshow(F);
title('salt & pepper noise');

subplot(3,1,2);
imshow(E)
title('Non-linear filtering');
ylabel(peaksnr1);


subplot(3,1,3);
imshow(grey2);
title('Linear filtering');
ylabel(peaksnr2);
```

24.11.2020

4. Implement LUM filter algorithm. The only input data are:
• image I of size m×n,
• vector [a,b] of mask size, a and b are odd numbers,
• parameter k.
Test your program.



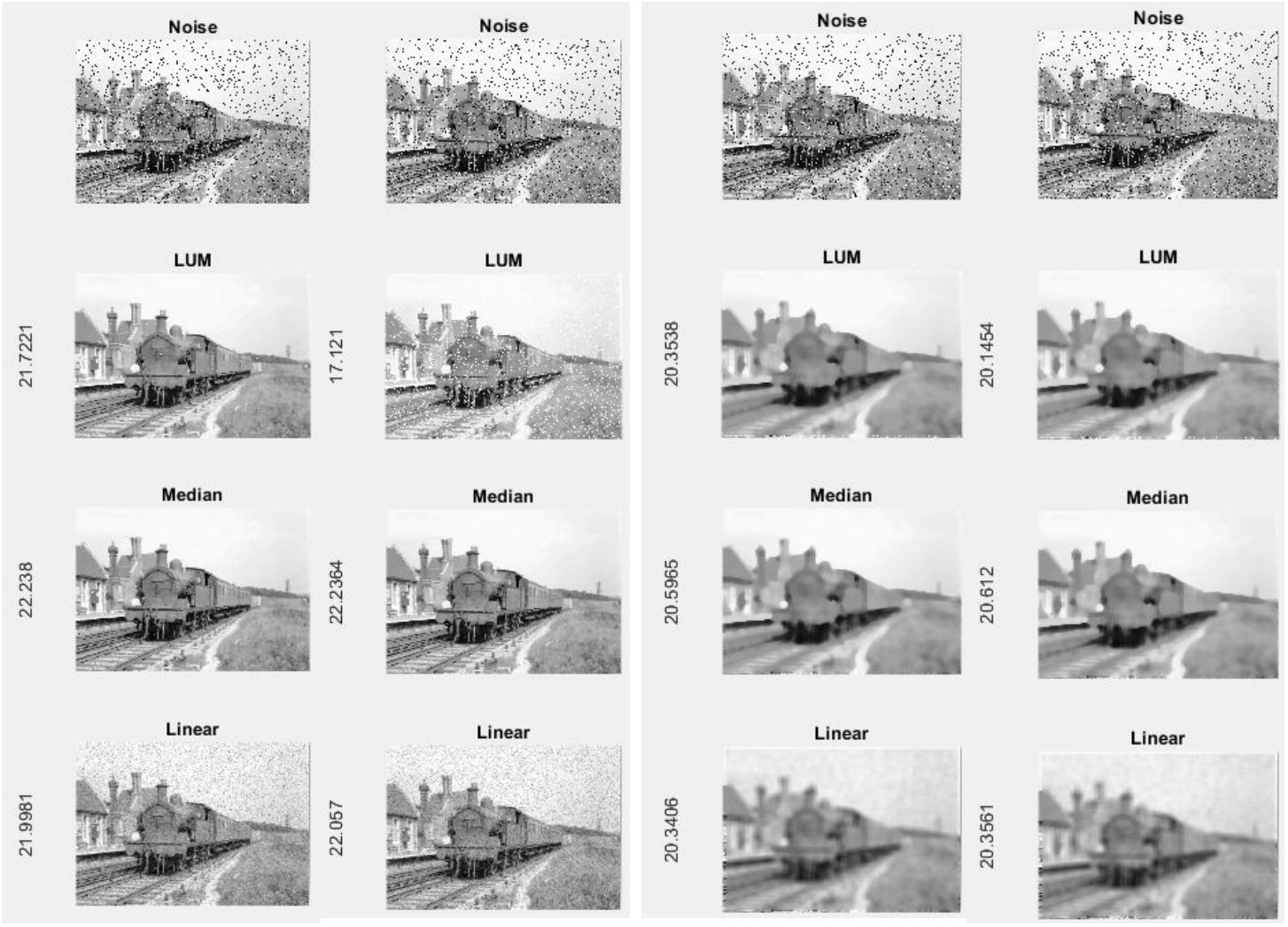
Shape of mask is the same (5x5) the only difference is in parameter k which is 2, 5, 10 respectively. It becomes more blurred along with increasing size and parameter k.

Code:

```
photo=imread('test3.jpg');
A = rgb2gray(photo);

h = ones(15,15)/25;

s1 = size(h,1)/2+0.5; %row
s2 = size(h,2)/2+0.5; %column
k = 25;

%LUM

for row = (s1):size(A,1)-(s1)
   for col = (s2):size(A,2)-(s2)
      B = A((row-s1+1):(row+s1-1),(col-s2+1):(col+s2-1));
      x0 = B(s1,s2);
      C = sort(reshape(B,1,[]));
      median = C(size(C,2)/2+0.5);
      l = C((size(C,2)/2+0.5)-k);
      u = C((size(C,2)/2+0.5)+k);

      if x0>l
         x0=l;
      end
      if x0<u
         x0=u;
      end
      D(row-(s1-1),col-(s2-1)) = x0;
   end
end

subplot(2,1,1)
imshow(A);
title('Orginal');
subplot(2,1,2)
imshow(D)
title('LUM');
```

5. Compare the noise reduction results obtained from LUM filter with results obtained from median filter and linear filter. Perform this comparison for Gaussian noise and impulse noise. Use PSNR for results evaluation.

3x3, k=2 3x3, k=3 15x15, k=3 15x15, k10



In case with a small mask we can observe that the median filter quite well filters out a noise. But with increasing the size of mask the method of filter give quite the same results.

Code:

```
photo=imread('test3.jpg');
A = rgb2gray(photo);
grey2 = rgb2gray(photo);

maskrow = 15;
maskcol = 15;
h = ones(maskrow,maskcol)/25;

point = size(h,1)/2-0.5;

s1 = size(h,1)/2+0.5; %row
s2 = size(h,2)/2+0.5; %column
k = 10;

% F = imnoise(I,'gaussian',0,0.025);
F = imnoise(A,'salt & pepper',0.1);
K = A ;
K(:,1:(2)) = [];
K((1:2),:) = [];
%LUM

for row = (s1):size(F,1)-(s1)
   for col = (s2):size(F,2)-(s2)
       B = F((row-s1+1):(row+s1-1),(col-s2+1):(col+s2-1));
       x0 = B(s1,s2);
       C = sort(reshape(B,1,[]));
       median = C(size(C,2)/2+0.5);
       l = C((size(C,2)/2+0.5)-k);
       u = C((size(C,2)/2+0.5)+k);

       if x0>l
          x0=l;
       end
       if x0<u
          x0=u;
       end
        D(row-1,col-1) = x0;
   end
end

subplot(4,1,1);
imshow(F);
title('Noise');
subplot(4,1,2);
imshow(D);
title('LUM');
ylabel(psnr(D,K));

%Median

for row = (s1):size(F,1)-(s1)
   for col = (s2):size(F,2)-(s2)
       G = F((row-s1+1):(row+s1-1),(col-s2+1):(col+s2-1));
       H = sort(reshape(G,1,[]));
```

```
        median = H(size(H,2)/2+0.5);
        I(row-1,col-1) = median;
    end
end

subplot(4,1,3);
imshow(I);
title('Median');
ylabel(psnr(I,K));

%Linear
for row = (point+1):size(F,1)-(point+1)
    for col = (point+1):size(F,2)-(point+1)
        grey2(row,col)=conv2(F((row-point):(row+point),(col-point):(col+point)),h,
'valid')/sum(h,'all');
    end
end

linear = psnr(grey2,A);


subplot(4,1,4);
imshow(grey2);
title('Linear');
ylabel(linear);
```