

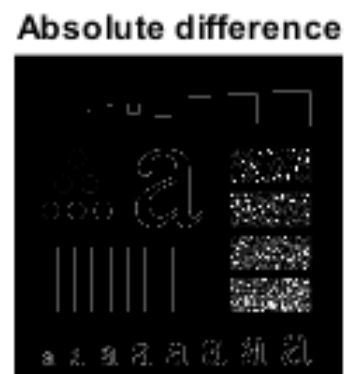
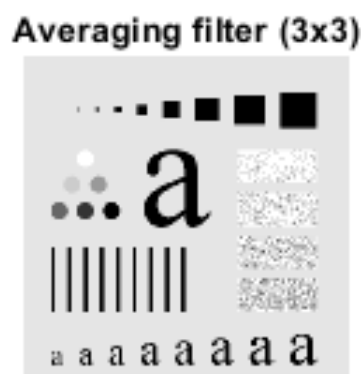
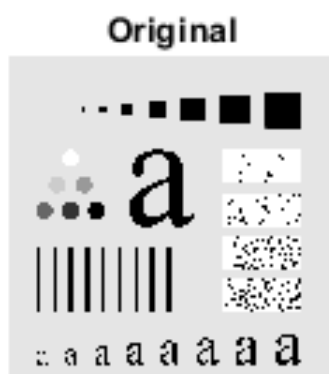
Digital image processing and vision systems – lab #4

Date performed: 13.04.2021	Group 2
Author name: Krzysztof Klimczyk	

1. Source codes and screenshots:

Task 5.3. Linear filters:

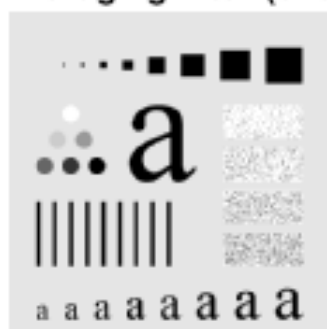
```
1. %averaging filter
2. figure('Name','Averaging filter','NumberTitle','off');
3. filter = fspecial('average',x); % x - kernel size
4. a = conv2(image,filter,'same');
5. a = uint8(a);
6. subplot(1,3,1);
7. imshow(image);
8. title("Original");
9.
10. subplot(1,3,2);
11. imshow(a);
12. title("Averaging filter");
13.
14. subplot(1,3,3);
15. absDiff = imabsdiff(image,a);
16. imshow(absDiff,[]);
17. title("Absolute difference");
```



Original



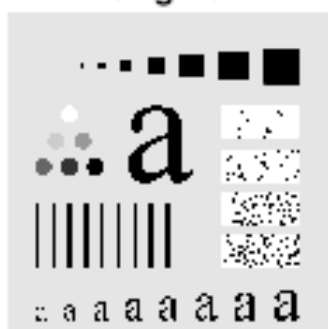
Averaging filter (5x5)



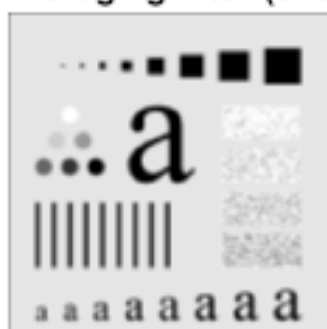
Absolute difference



Original



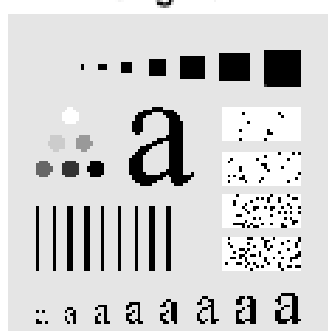
Averaging filter (9x9)



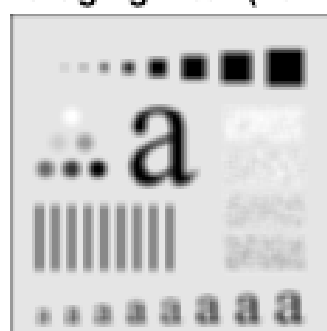
Absolute difference



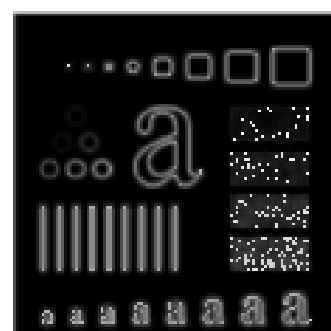
Original



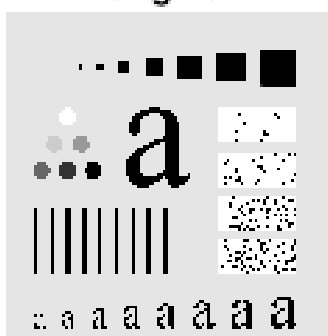
Averaging filter (15x15)



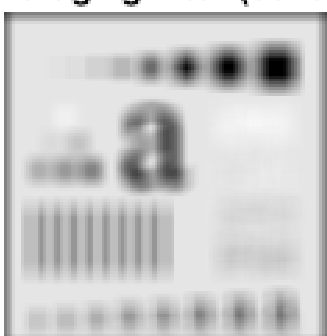
Absolute difference



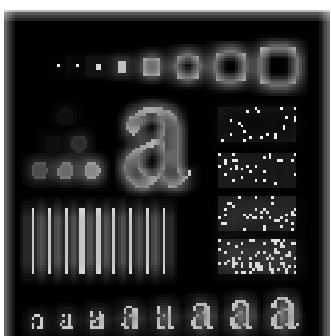
Original



Averaging filter (35x35)



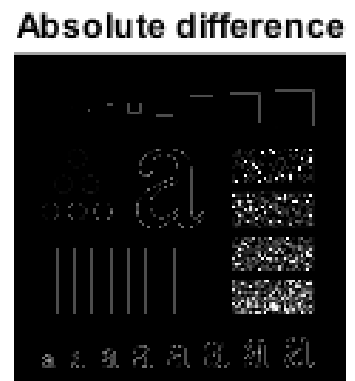
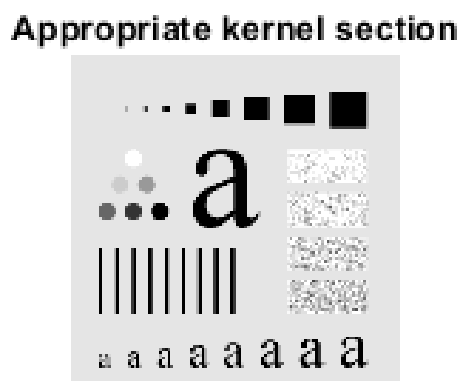
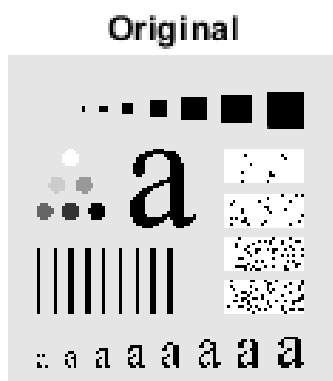
Absolute difference



```

1. %appropriate kernel section
2. M = [1 2 1; 2 4 2; 1 2 1];
3. M = M/sum(sum(M));
4. figure('Name','Appropriate kernel section','NumberTitle','off');
5.
6. filter = M;
7. a = conv2(image,filter,'same');
8. a = uint8(a);
9. subplot(1,3,1);
10. imshow(image);
11. title("Original");
12.
13. subplot(1,3,2);
14. imshow(a);
15. title("Appropriate kernel section");
16.
17. subplot(1,3,3);
18. absDiff = imabsdiff(image,a);
19. imshow(absDiff,[]);
20. title("Absolute difference");

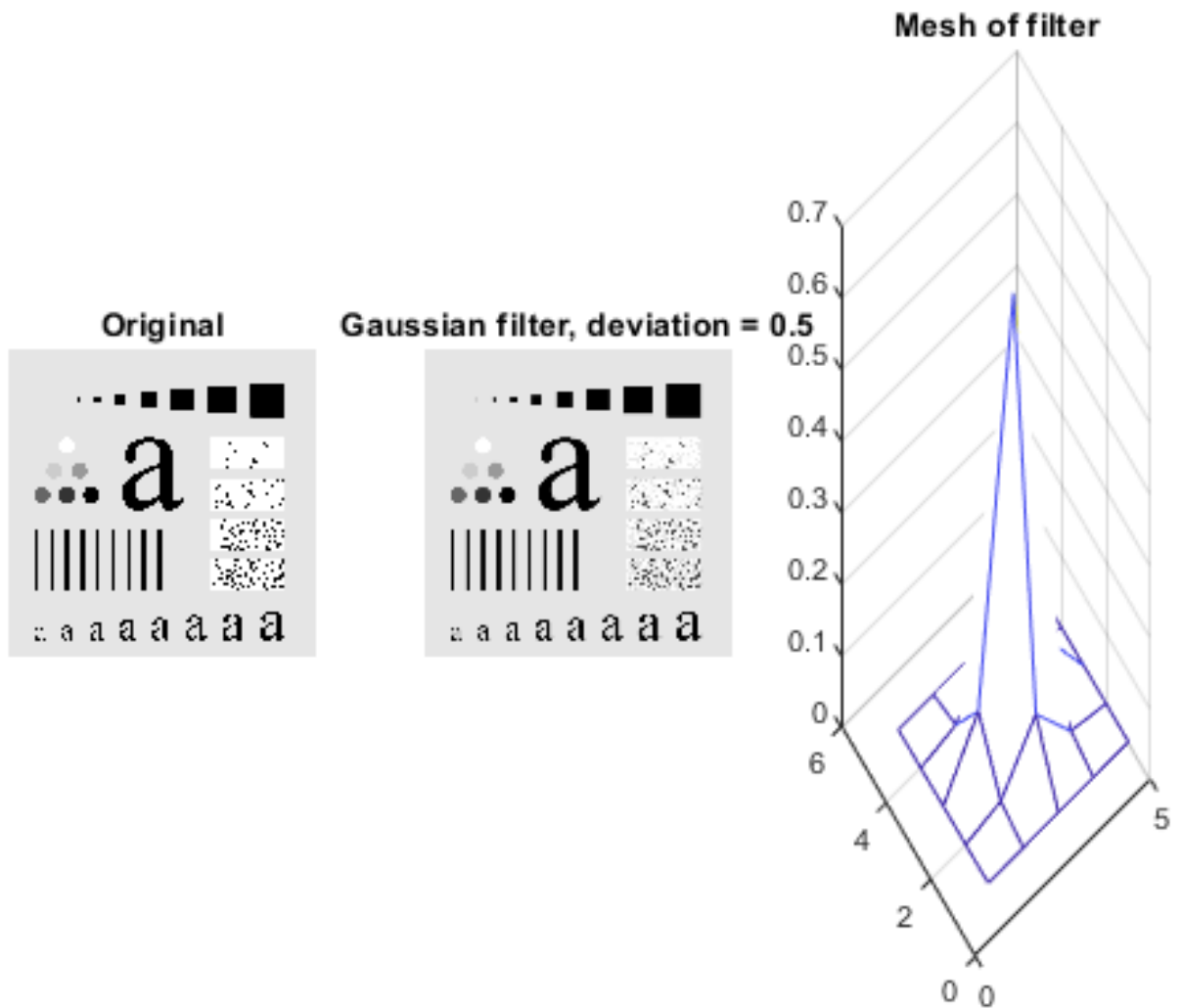
```

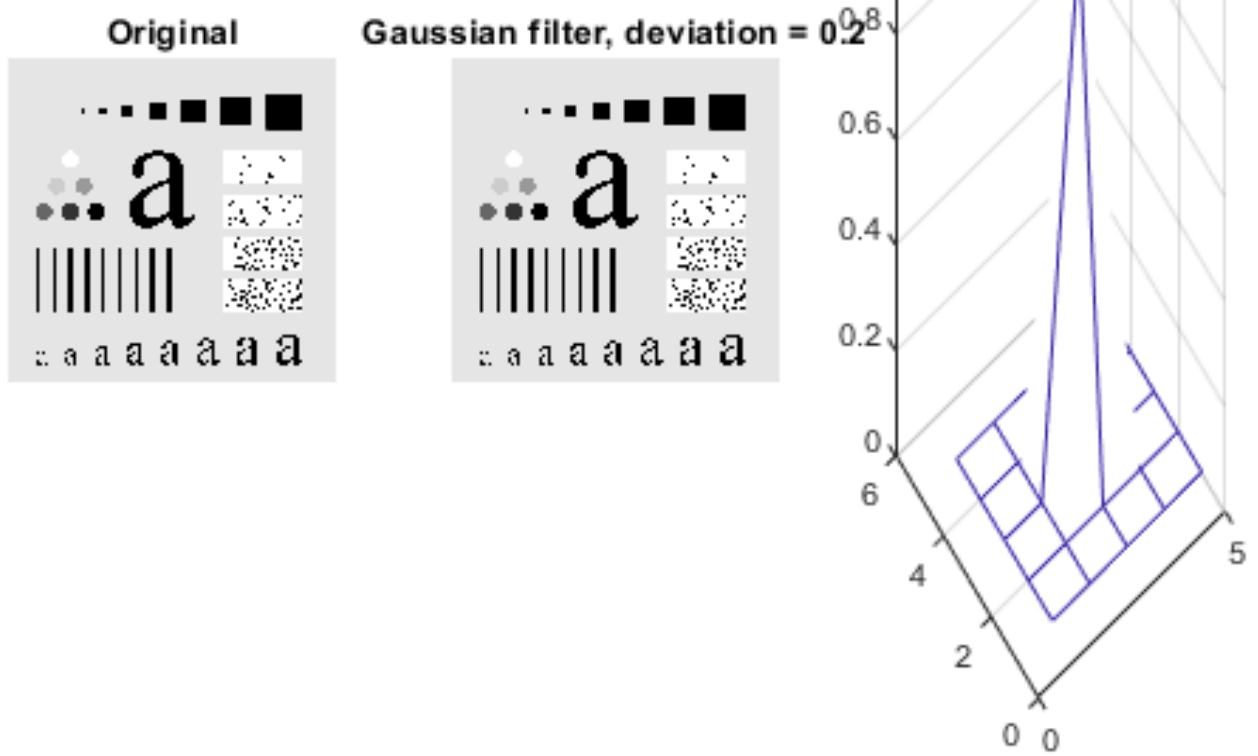
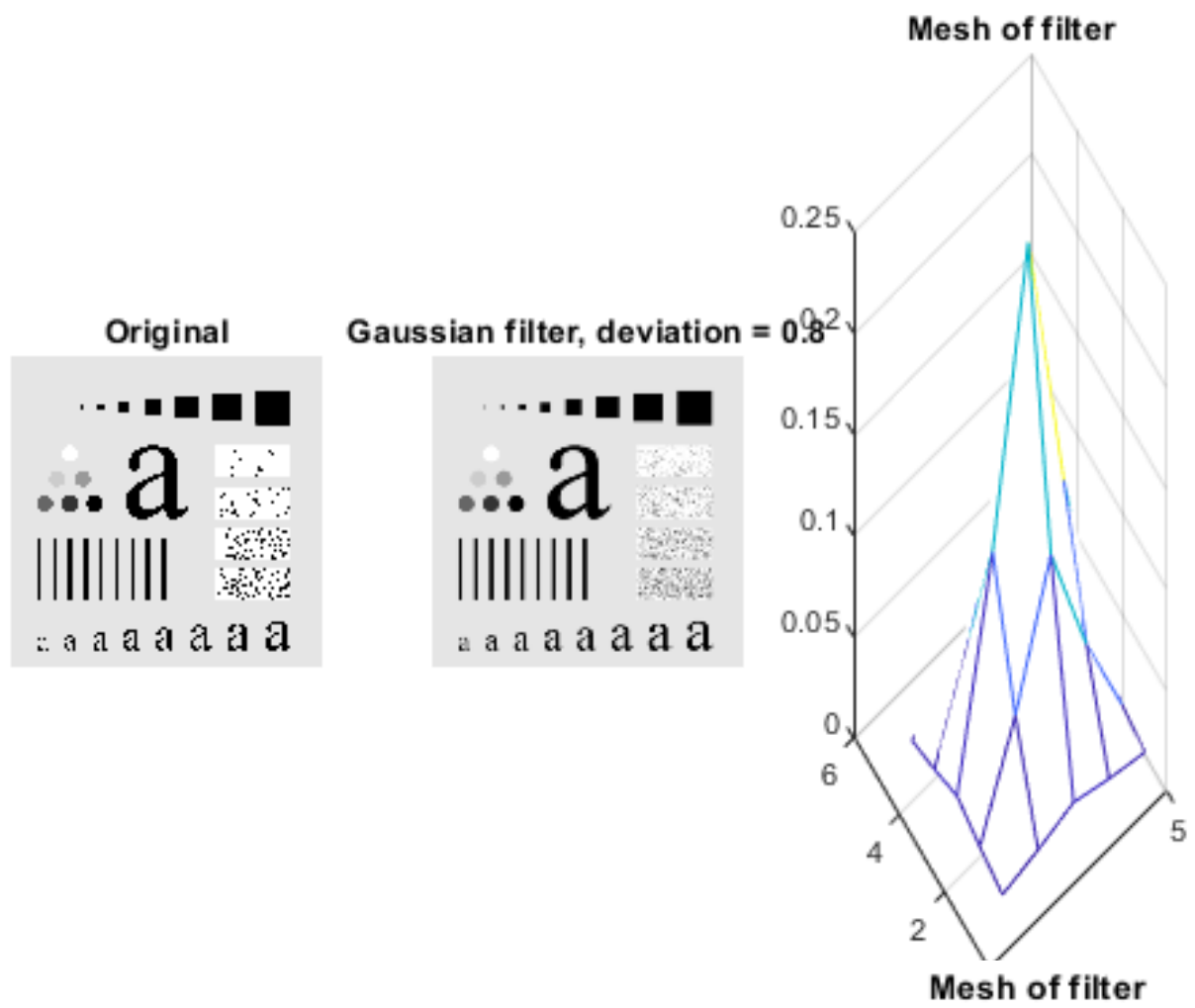


```

1. %gaussian filter, deviation = x
2. figure('Name','Gaussian filter, deviation =
   0.5','NumberTitle','off');
3. filter = fspecial('gaussian',5,x); % x - value of deviation
4.
5. a = conv2(image,filter,'same');
6. a = uint8(a);
7. subplot(1,3,1);
8. imshow(image);
9. title("Original");
10.
11. subplot(1,3,2);
12. imshow(a);
13. title("Gaussian filter, deviation = 0.5");
14.
15. subplot(1,3,3);
16. mesh(filter);
17. title("Mesh of filter");

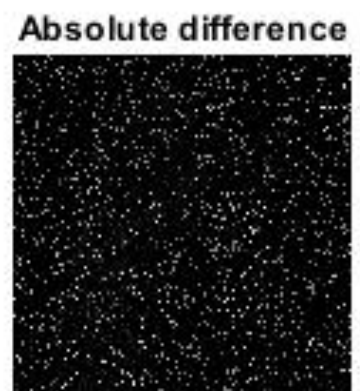
```





Task 5.4. Non-linear filters:

```
1. %median filtering lenaNoise
2. figure('Name','Median filter','NumberTitle','off');
3. a = medfilt2(image);
4. a = uint8(a);
5. subplot(1,3,1);
6. imshow(image);
7. title("Original");
8.
9. subplot(1,3,2);
10. imshow(a);
11. title("Median filter (3x3)");
12.
13. subplot(1,3,3);
14. absDiff = imabsdiff(image,a);
15. imshow(absDiff,[]);
16. title("Absolute difference");
```



```
1. %median and averaging filtering lena
2. image = imread("lena.bmp");
3.
4. figure('Name','Median & averaging filter','NumberTitle','off');
5. a = medfilt2(image);
6. a = uint8(a);
7. subplot(2,3,1);
8. imshow(image);
9. title("Original");
10.
11. subplot(2,3,2);
12. imshow(a);
13. title("Median filter (3x3)");
14.
15. subplot(2,3,3);
16. absDiff = imabsdiff(image,a);
17. imshow(absDiff,[]);
18. title("Absolute difference");
19. %%
20. filter = fspecial('average',3);
```

```

21. a = conv2(image,filter,'same');
22. a = uint8(a);
23. subplot(2,3,4);
24. imshow(image);
25. title("Original");
26.
27. subplot(2,3,5);
28. imshow(a);
29. title("Averaging filter (3x3)");
30.
31. subplot(2,3,6);
32. absDiff = imabsdiff(image,a);
33. imshow(absDiff,[]);
34. title("Absolute difference");

```

Original



Median filter (3x3)



Absolute difference



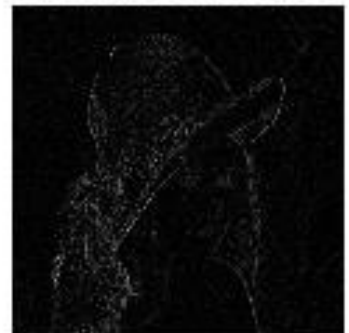
Original



Averaging filter (3x3)



Absolute difference



```

1. %loop x10 median filtering lena
2. image = imread("lena.bmp");
3.
4. figure('Name','Posterization x10','NumberTitle','off');
5.
6. for i=1:10
7. a = medfilt2(image,[5 5]);
8. a = uint8(a);
9. end
10.
11. subplot(1,3,1);
12. imshow(image);
13. title("Original");
14.
15. subplot(1,3,2);
16. imshow(a);
17. title("Posterization x10");
18.
19. subplot(1,3,3);
20. absDiff = imabsdiff(image,a);
21. imshow(absDiff,[]);
22. title("Absolute difference");

```

Original



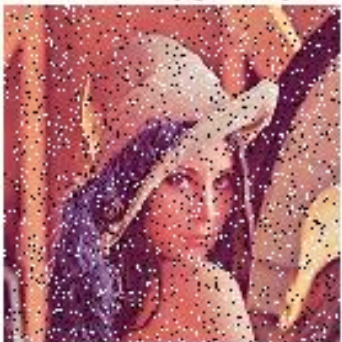
Posterization x10



Absolute difference



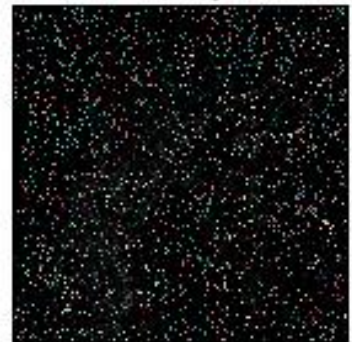
Obraz oryginalny



Obraz po filtracji medianowej



Modul z różnicy obrazów



Task 5.5.1. Laplacian:

```
1. %laplacian mask
2. figure('Name','Laplacian','NumberTitle','off');
3. M = [ 0 1 0; 1 -4 1; 0 1 0];
4. M = M./9;
5. filter = M;
6. a = conv2(image,filter,'same');
7. a = uint8(a);
8. subplot(1,3,1);
9. imshow(image);
10. title("Original");
11.
12. subplot(1,3,2);
13. b = a + 128;
14. imshow(b,[]);
15. title("Adding 128 to each pixel");
16.
17. subplot(1,3,3);
18. c = abs(a);
19. imshow(c,[]);
20. title("Absolute value");
```

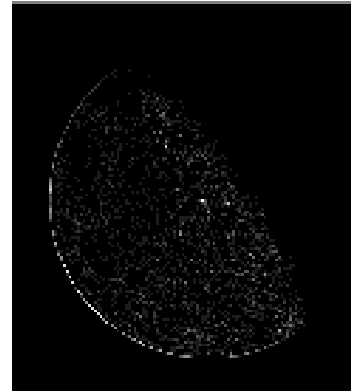
Original



Adding 128 to each pixel



Absolute value



```
1. %laplacian kernel - sharpening
2. figure('Name','Laplacian filter','NumberTitle','off');
3. filter = fspecial('laplacian');
4. a = conv2(image,filter,'same');
5. a = uint8(a);
6. subplot(1,3,1);
7. imshow(image);
8. title("Original");
9.
10. subplot(1,3,2);
11. b = image + a;
12. imshow(b,[]);
13. title("Adding laplacian filter");
14.
15. subplot(1,3,3);
16. c = image - a;
17. imshow(c,[]);
18. title("Subtracting laplacian filter");
```

Original



Adding laplacian filter



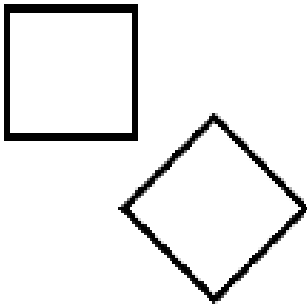
Subtracting laplacian filter



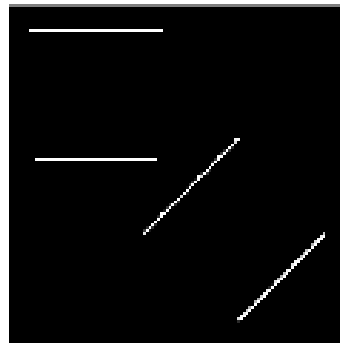
Task 5.5.2. Gradients:

```
1. %R1,R2,P1,P2,S1,S2
2. figure('Name','Roberts gradient','NumberTitle','off');
3. filter = x; %x - specific filter (R1,R2,P1,P2,S1,S2)
4. a = conv2(image,filter,'same');
5. a = uint8(a);
6. subplot(1,3,1);
7. imshow(image);
8. title("Original");
9.
10. subplot(1,3,2);
11. b = a + 128;
12. imshow(b,[]);
13. title("Adding 128 to each pixel");
14.
15. subplot(1,3,3);
16. c = abs(a);
17. imshow(c,[]);
18.
19. title("Absolute value");
```

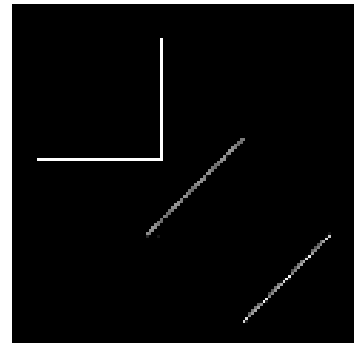
Original



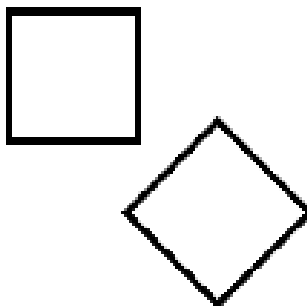
Adding 128 to each pixel (R1)



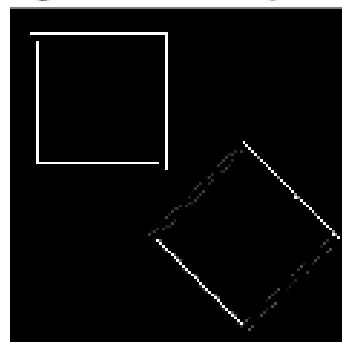
Absolute value (R1)



Original



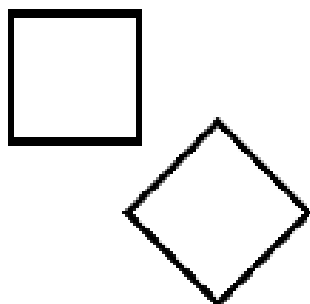
Adding 128 to each pixel (R2)



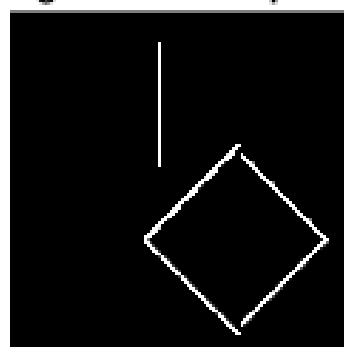
Absolute value (R2)



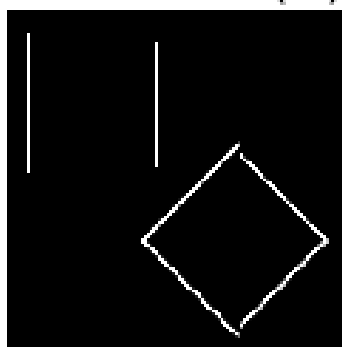
Original



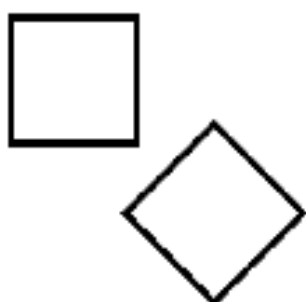
Adding 128 to each pixel (P1)



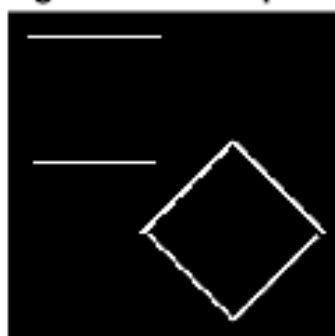
Absolute value (P1)



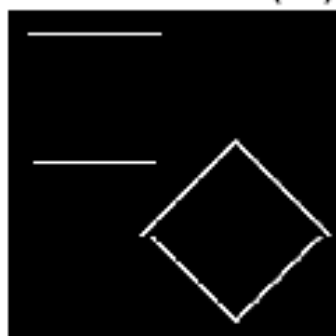
Original



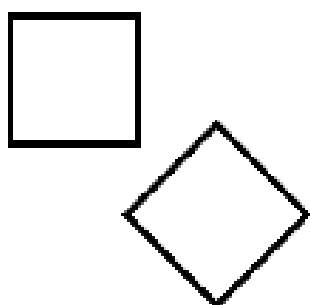
Adding 128 to each pixel (P2)



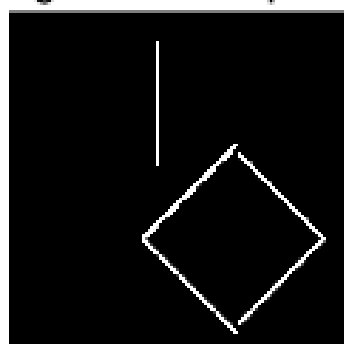
Absolute value (P2)



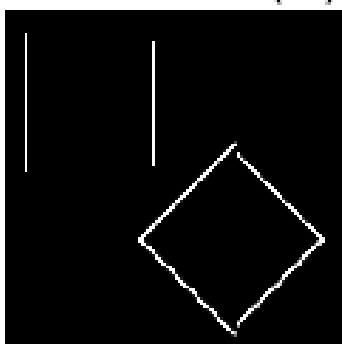
Original



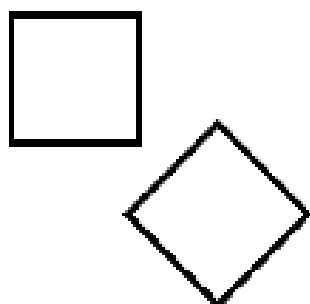
Adding 128 to each pixel (S1)



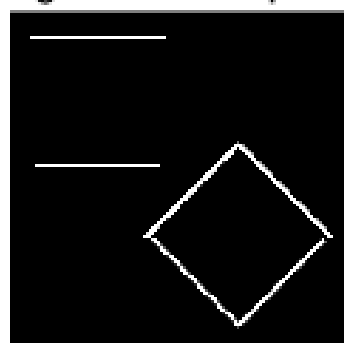
Absolute value (S1)



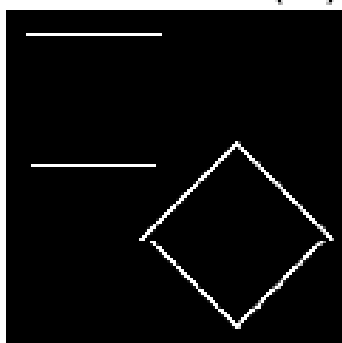
Original



Adding 128 to each pixel (S2)



Absolute value (S2)



```

1. %Sobels combination
2. figure('Name','Sobels combination','NumberTitle','off');
3. subplot(1,3,1);
4. imshow(image);
5.
6. title("Original");
7.
8. subplot(1,3,2);
9. OI = sqrt((conv2(image,S1)).^2 + (conv2(image,S2)).^2);
10. imshow(OI, []);
11. title("Square root");
12.
13. subplot(1,3,3);
14. OI = (conv2(image,S1)).^2 + (conv2(image,S2)).^2;
15. imshow(OI, []);
16. title("Square sum");

```



2. Conclusions:

The lowpass filtering is used to remove sharp edges of the image at the expense of sharpness. The output of absolute difference presents the pixels whose values are changed by the algorithm. The size of kernel mask has a high impact on overall sharpness of the image. The gaussian mask also affects sharpness and edges. The greater value of standard deviation means more loss of sharpness.

The median filtering removes the noise very well but there may stay a few pixels with incorrect value. Comparing median and averaging filtering it can be noticed that the output image is similar, the difference is visible in the picture of absolute difference. Multiple-use of the median filter called posterization makes the image look unnatural.

Two presented methods of normalization give similar results. In this case, adding the result of laplacian filtration to the original image leads to sharpening effect.

All of the presented masks give a different result of edge detection. Some detect verticals and other horizontal edges. Two versions of a combination of Sobles kernels recognize jet contour without any problems. Edges in the square root method are thicker and more visible.