Computer Image Processing

Classes 8 - image filtering continued

Example 1

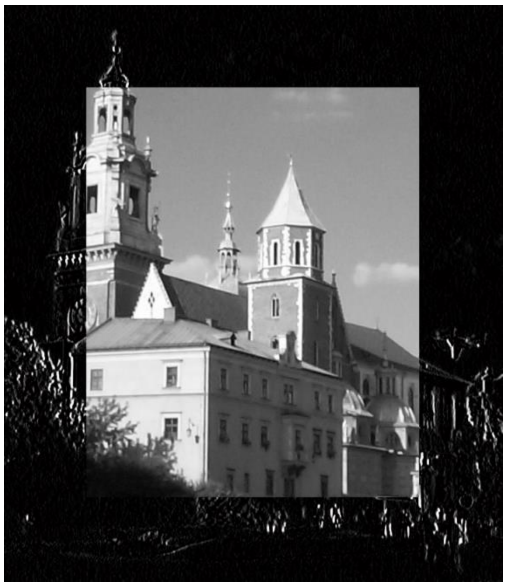
Gradient filters - Roberts, Prewitt and Sobel.

```
L=imread('Cracow_1.jpg','jpg');
L =rgb2gray(L);
figure(1);
imshow(L);

%ROBERTS
Kernel1=[0,0,0; -1, 0, 0; 0, 1, 0];
L1=filter2(Kernel1, L);
L1=mat2gray(L1);
figure(2); imshow(L1);
%PREWITT
Kernel2=[-1,-1,-1; 0, 0, 0; 1, 1, 1];
L2=filter2(Kernel2, L);
L2=mat2gray(L2);
figure(3); imshow(histeq(L2));
%SOBEL
Kernel3=[-1,0,1; -2, 0, 2; -1, 0, 1];
L3=filter2(Kernel3, L);
L3=mat2gray(L3);
figure(4); imshow(histeq(L3));
```

Exercise 1

Combine the image filtered with the use of Sobel filter with the according monochrome image in the way that filtered image creates a frame around the monochrome image.



L=imread('Krakow_3.jpg','jpg');
L =rgb2gray(L);

```
figure(1);
imshow(L);

[w k] = size(L);
n=125;
L2 = L(n:(w-n),n:(k-n));
L2 = double(L2)/255;
Kernel=[-1,0,1; -2, 0, 2; -1, 0, 1];
L3=filter2(Kernel, L);
L3=L3/255;
L3(n:(w-n),n:(k-n)) = L2;
figure(2); imshow(L3);
```

Example 2

Dynamic creation of filter kernels.

```
L = imread('portrait.jpg');
figure;
imshow(L);
hold on;
setmask = [strcat('m = [str2num(get(t1,''String'')), str2num(get(t2,''String'')), 
str2num(get(t3,''String''));',...
    'str2num(get(t4,''String'')), str2num(get(t5,''String'')), 
str2num(get(t6,''String''));',...
    'str2num(get(t7,''String'')), str2num(get(t8,''String'')), 
str2num(get(t9,''String''))];',...
    'L1 = filter2(m,L); L1 = mat2gray(L1); imshow(L1)')];
t1 = uicontrol('Style', 'edit', 'String', num2str(1), 'Position', [10 120 30 30], 
'Callback', setmask);
t2 = uicontrol('Style', 'edit', 'String', num2str(1), 'Position', [50 120 30 30], 
'Callback', setmask);
t3 = uicontrol('Style', 'edit', 'String', num2str(1), 'Position', [90 120 30 30], 
'Callback', setmask);
t4 = uicontrol('Style', 'edit', 'String', num2str(1), 'Position', [10 80 30 30], 
'Callback', setmask);
t5 = uicontrol('Style', 'edit', 'String', num2str(1), 'Position', [50 80 30 30], 
'Callback', setmask);
t6 = uicontrol('Style', 'edit', 'String', num2str(1), 'Position', [90 80 30 30], 
'Callback', setmask);
t7 = uicontrol('Style', 'edit', 'String', num2str(1), 'Position', [10 40 30 30], 
'Callback', setmask);
t8 = uicontrol('Style', 'edit', 'String', num2str(1), 'Position', [50 40 30 30], 
'Callback', setmask);
t9 = uicontrol('Style', 'edit', 'String', num2str(1), 'Position', [90 40 30 30], 
'Callback', setmask);
```

Exercise 2

Create Sobel filters detecting edges with different inclinations - horizontal, vertical and diagonal.

```
L=imread('Cracow_3.jpg');
figure(1);
L =rgb2gray(L);
imshow(L);

Kernel1=[-1,0,1; -2, 0, 2; -1, 0, 1];
L1=imfilter(L,Kernel1);
figure(2); imshow(L1);
title('vertical');
```

```
Kernel1=[-1,-2,-1; 0,0,0; 1,2,1];
L1=imfilter(L,Kernel1);
figure(3); imshow(L1);
title('horizontal');
```

Example 3

Laplace filter detecting all edges, regardless of the inclination.

```
L=imread('Cracow_3.jpg');
figure(1);
imshow(L);
Kernel1=[0,-1,0; -1,4,-1; 0,-1,0];
L1=imfilter(L,Kernel1);
figure(2); imshow(L1);
```

Exercise 3

Use the LAPL2 and LAPL3 filters to detect edges. To make the effect clearer, try to binarize the image.

```
L=imread('Cracow_3.jpg');
figure(1);
imshow(L);
Kernel1=[-1,-1,-1; -1,8,-1; -1,-1,-1];
L1=imfilter(L,Kernel1);
figure(2); imshow(L1);
title('LAPL2');

Kernel1=[1,-2,1; -2,4,-2; 1,-2,1];
L1=imfilter(L,Kernel1);
figure(3); imshow(L1);
title('LAPL3');
```

Example 4

Solution to the problem of the frame appearing around the image after filtering.

```
L=imread('portrait.jpg');
imshow (L), title('source image);
h=ones(5)/25; %dividing by the sum of coefficients
L1=imfilter(L,h);
figure;
imshow(L1), title ('image without duplicating edge pixels');
L2=imfilter(L, h, 'replicate');
figure;
imshow (L2), title ('image with duplicated edge pixels');
```

Example 5

Predefined filters.

```
L = imread('portrait.jpg');
h = fspecial('unsharp');
L1 = imfilter(L,h,'replicate');
imshow(L);
figure, imshow(L1);
```

Try to use different filters: 'average', 'gaussian', 'laplacian', 'log', 'motion', 'prewitt', 'sobel'.

Example 6

Median filter.

```
L=imread('portrait.jpg');
figure(1);
imshow(L);
L1=medfilt2(L, [5,5]);
L1=mat2gray(L1);
figure(2); imshow(L1);
```

Exercise 4

Perform filtering the image *square.bmp* with the use of median filter with kernel size 5, 15 and 45. What happens to the corners of the square?

```
L=imread('square.bmp');
figure(1);
imshow(L);
L1=medfilt2(L, [5,5]);
L1=mat2gray(L1);
figure(2); imshow(L1);
title('kernel size 5 x 5');

L1=medfilt2(L, [15,15]);
L1=mat2gray(L1);
figure(3); imshow(L1);
title('kernel size 15 x 15');

L1=medfilt2(L, [45,45]);
L1=mat2gray(L1);
figure(4); imshow(L1);
title('kernel size 45 x 45');

L1=medfilt2(L, [75,75]);
L1=mat2gray(L1);
figure(5); imshow(L1);
title('kernel size 75 x 75 :)');
```

Example 7

Minimum filter.

```
L=ones([16, 16])*256;
L(6:11, 6:11)=(0:43:256)'*ones([1 6]);
L(14,14)=0;
L=uint8(L);
figure(1);
imshow(L, 'InitialMagnification', 'fit');

L1=imnoise(L, 'salt & pepper', 0.3);
figure(2);
imshow(L1,'InitialMagnification', 'fit');

Kernel=[3,3]
L2=nlfilter(L1, Kernel, 'min(x(:))');
L2=mat2gray(L2);
figure(3); imshow(L2,'InitialMagnification', 'fit');
```

```
Kernel=[10,10];
L3=nlfilter(L1, Kernel, 'min(x(:))');
L3=mat2gray(L3);
figure(4); imshow(L3, 'InitialMagnification', 'fit');
```

Exercise 5

The code below presents an example of adding salt and pepper noise to the image. Try to filter this distortion with the most suitable filter from the ones we've known so far.

```
L=rgb2gray(L);
L1=imnoise(L, 'salt & pepper', 0.3);
figure;
imshow(L1);
title('noise = 0.3');

L1=medfilt2(L1, [5,5]);
L1=mat2gray(L1);
figure(2); imshow(L1);
title('median filter 5 x 5');

L1=imnoise(L, 'salt & pepper', 0.8);
figure(3);
imshow(L1);
title('noise = 0.8');

L1=medfilt2(L1, [17,17]);
L1=mat2gray(L1);
figure(4); imshow(L1);
title('median filter 17 x 17');
```