Computer Vision I

Tim Luchterhand, Paul Nykiel (Group 17)

28. Mai 2018

1 Histogram Calculation

1. Matlab-Funktion:

```
function H = myHistogram(im)
    H = zeros(1,256);
    [width, height] = size(im);
    for x = 1:width
        for y = 1:height
            intensity = im(x,y);
            assert(intensity >= 0 & intensity <= 255, "Not
                all matrix elements are between 0 and 255");
            H(intensity+1) = H(intensity+1) + 1;
        end
    end
    H .*= 1/(width*height);
end
```

2. Generate the plots:

```
% Read the images
fruitsA = imread('images/fruitsA.png');
fruitsB = imread('images/fruitsB.png');

% Calculate the histograms
histA = myHistogram(fruitsA);
histB = myHistogram(fruitsB);

% Plot the histograms
figure();

subplot(2,2,1);
```

```
imshow(fruitsA);
title('fruitsA.png');

subplot(2,2,2);
plot(0:255, histA);
axis([0 255 0 0.04]);
title('Histogramm for fruitsA.png');
xlabel('Intensity');
ylabel('Probability');

subplot(2,2,3);
imshow(fruitsB);
title('fruitsB.png');

subplot(2,2,4);
plot(0:255, histB);
axis([0 255 0 0.05]);
title('Histogramm for fruitsB.png');
xlabel('Intensity');
ylabel('Probability');

print('Histograms','-depsc')
```

3.

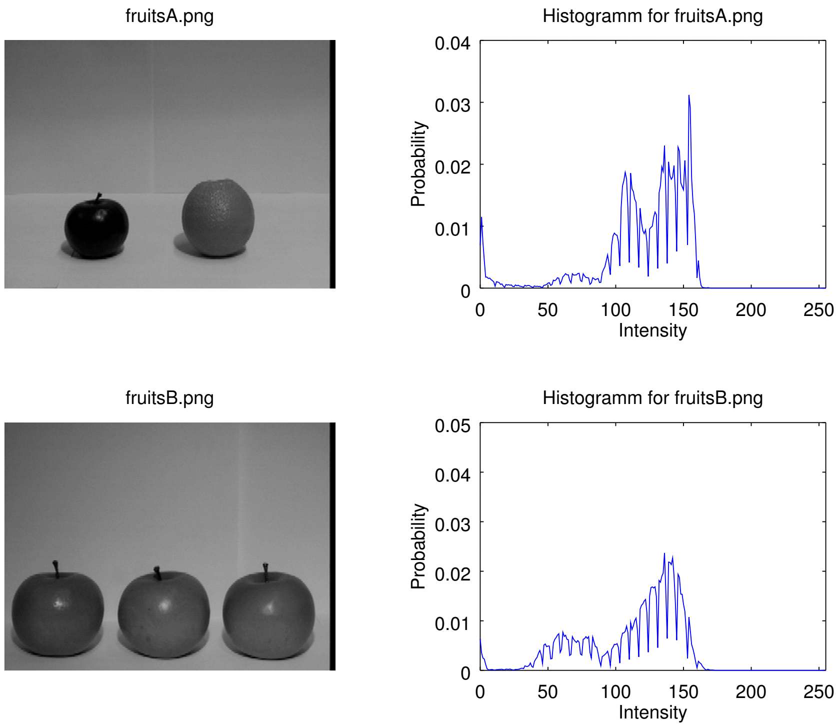

Abbildung 1: Plot of the histograms

4.

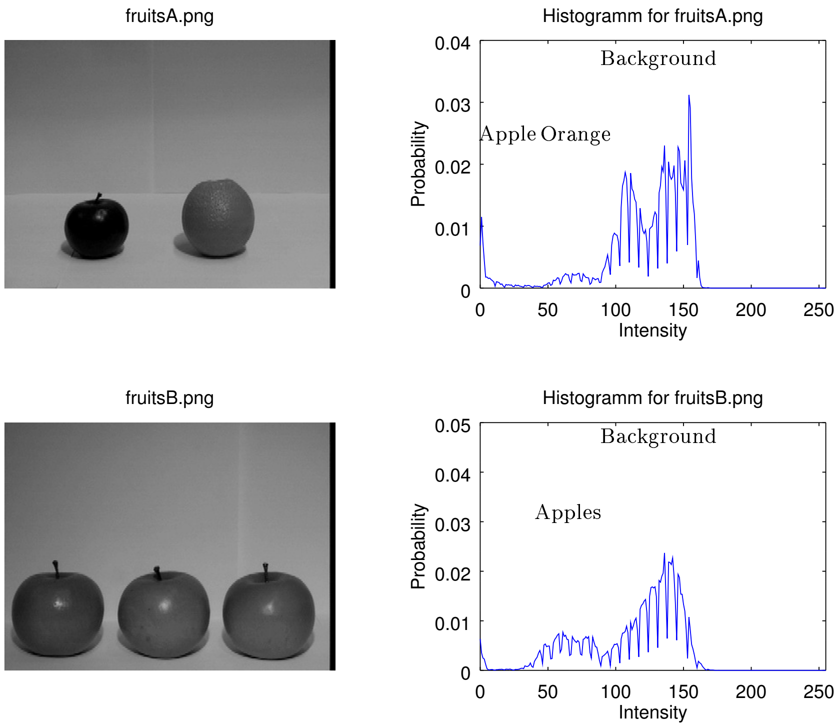

Abbildung 2: Plot of the histograms

5.

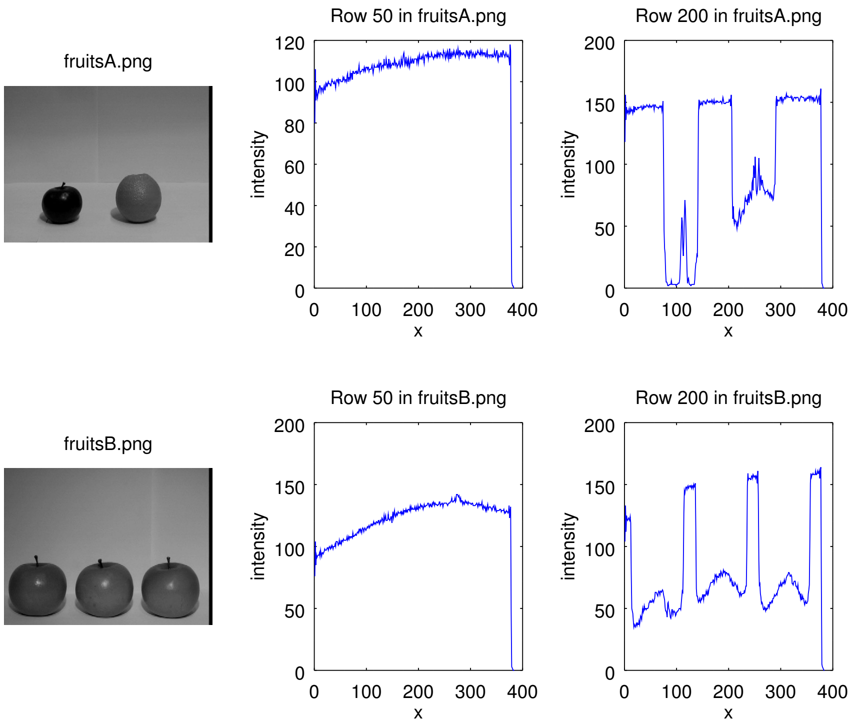

Abbildung 3: Plot of the rows

2 Local weighting

1.
$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ 1\cdot 1+1\cdot 1 & 1\cdot 1 & -1\cdot 1+-1\cdot 1+1\cdot 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ 2 & 1 & -1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

2. Negative derivative in red, positive in green, zero in blue.

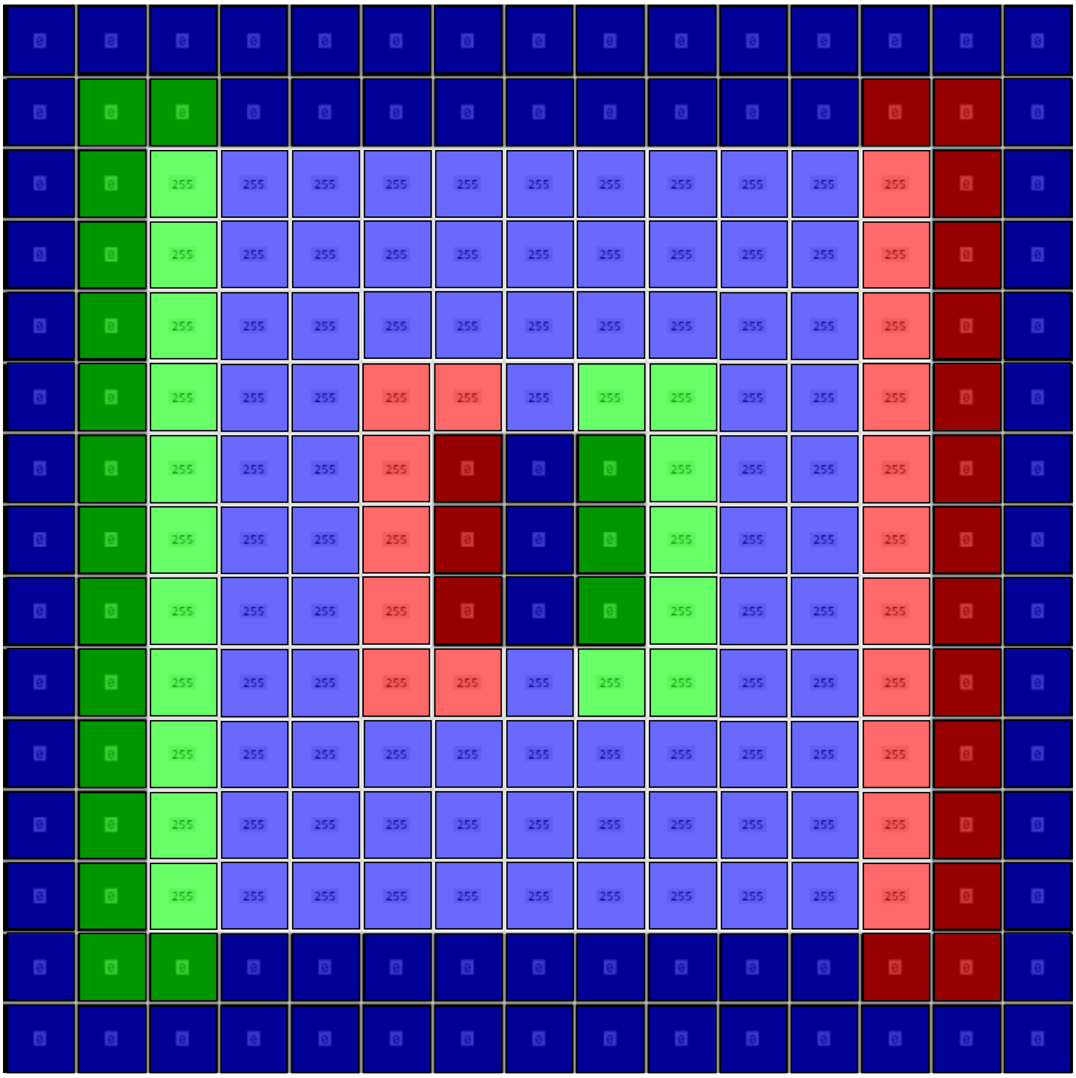

Abbildung 4: Local derivative using G_x

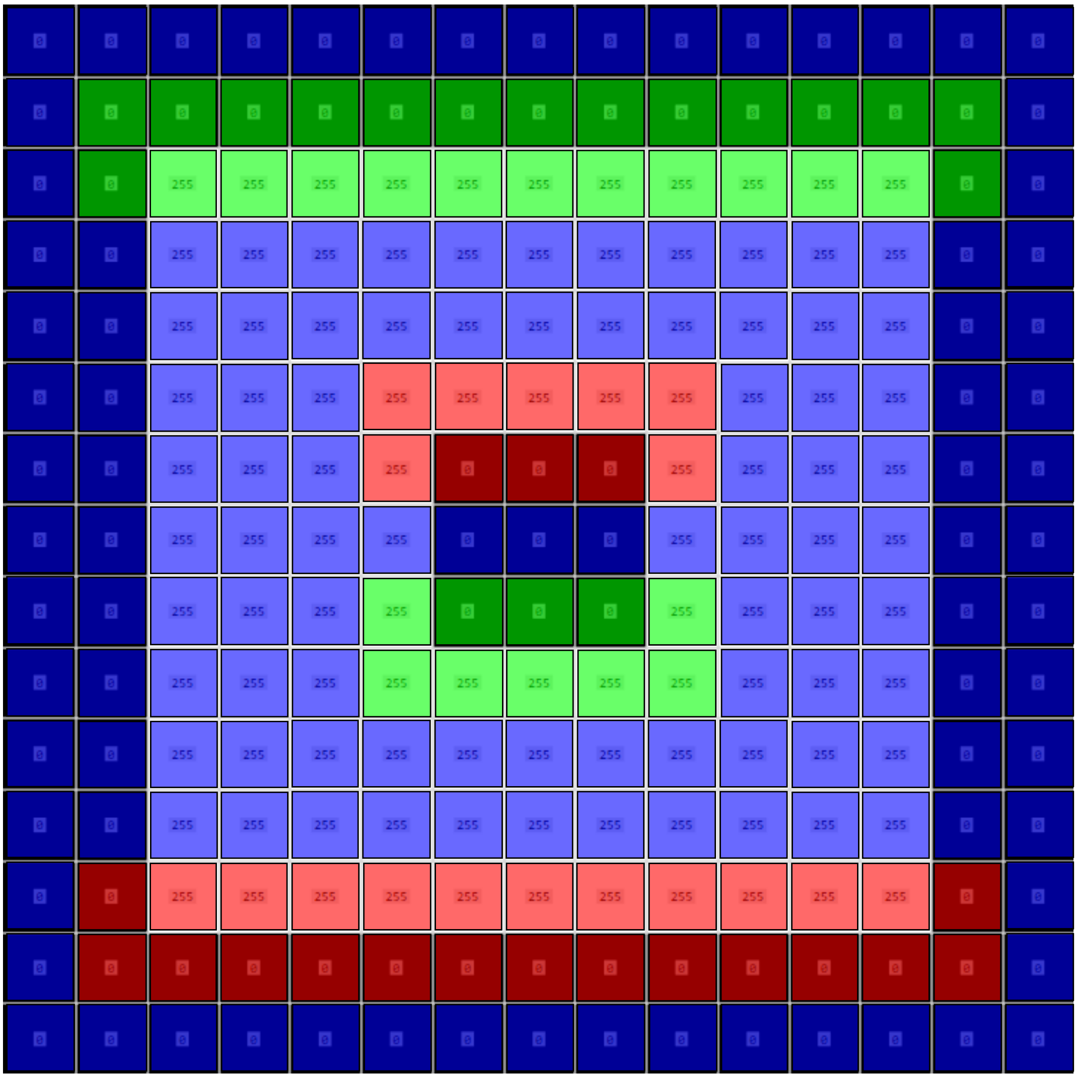

Abbildung 5: Local derivative using G_y

3.

```
lena = imread('images/lena.tif');
lenaNoise = imread('images/lenaNoise.tif');

B = ones(3,3) / 9;

subplot(2,2,1);
imshow(lena);
```

```
title('lena.tif');

subplot(2,2,2);
imshow(imfilter(lena,B));
title('Filtered lena.tif');

subplot(2,2,3);
imshow(lenaNoise);
title('lenaNoise.tif');

subplot(2,2,4);
imshow(imfilter(lenaNoise,B));
title('Filtered lenaNoise.tif');

print('BoxFilter', '-depsc');
```

This filter can be used to smooth an image and therefor reduce noise. It is sort of a blurring filter.

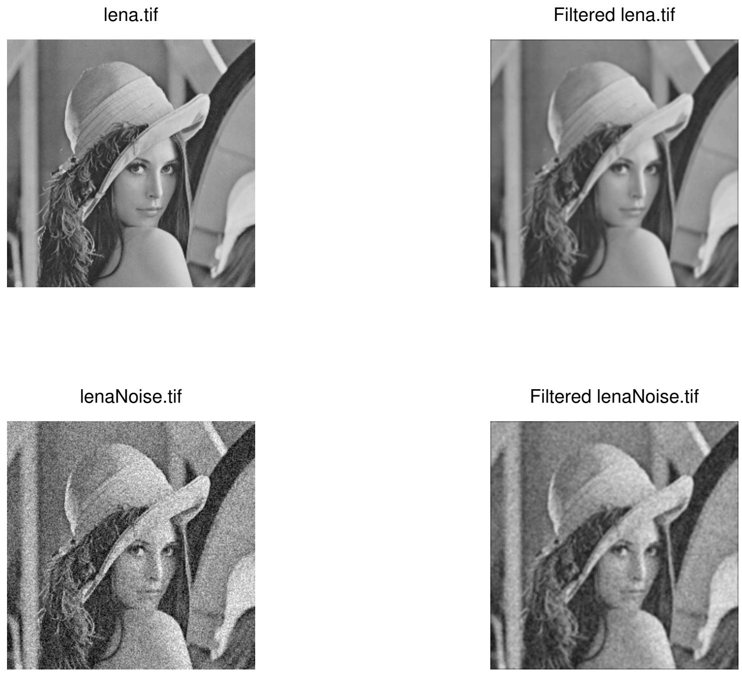

Abbildung 6: Original and filtered images