Computer Vision I - Sheet 3
Group 2

Jonas Otto
jonas@jonasotto.com

Dominik Authaler
dominik.authaler@uni-ulm.de

June 20, 2019

1 Histogram equalization

The source code used for this task can be found beneath or in the file sh03ex01.m. Figure 1 shows the results of this exercise. The left part of the figure shows the original image together with it's histogram and the cumulative histogram. The right part shows the improved image, which was received after applying an equalization to the histogram. Therefore the cumulative histogram now shows some linear behaviour. As can be seen in the code we used a lookup-table for mapping between the original intensity value and the new intensity value.

```
%% Group 2: Dominik Authaler, Jonas Otto
close all;
clc;
clear;

%% Calculation
image = imread('../images/bookstore_dark.tif');

histogram = myHistogram(image);

cumulativeHistogram = cumsum(histogram);

equalizedImage = intlut(image, uint8(cumulativeHistogram
    *256));

%% Visualization
rows = 3;
cols = 2;

subplot(rows, cols, 1);
imshow(image);
title('Input Image');

subplot(rows, cols, 2);
imshow(equalizedImage);
title('Equalized Image');

subplot(rows, cols, 3);
bar(0:255, histogram);
title('Histogram');

subplot(rows, cols, 4);
bar(0:255, myHistogram(equalizedImage));
title('Histogram');

```

```
35 subplot(rows, cols, 5);
36 bar(0:255, cumulativeHistogram);
37 title("Cumulative Histogram");
38
39 subplot(rows, cols, 6);
40 bar(0:255, cumsum(myHistogram(equalizedImage)));
41 title("Cumulative Histogram");
42
43 saveas(gcf,'../images/ex01.eps', 'epsc')
```

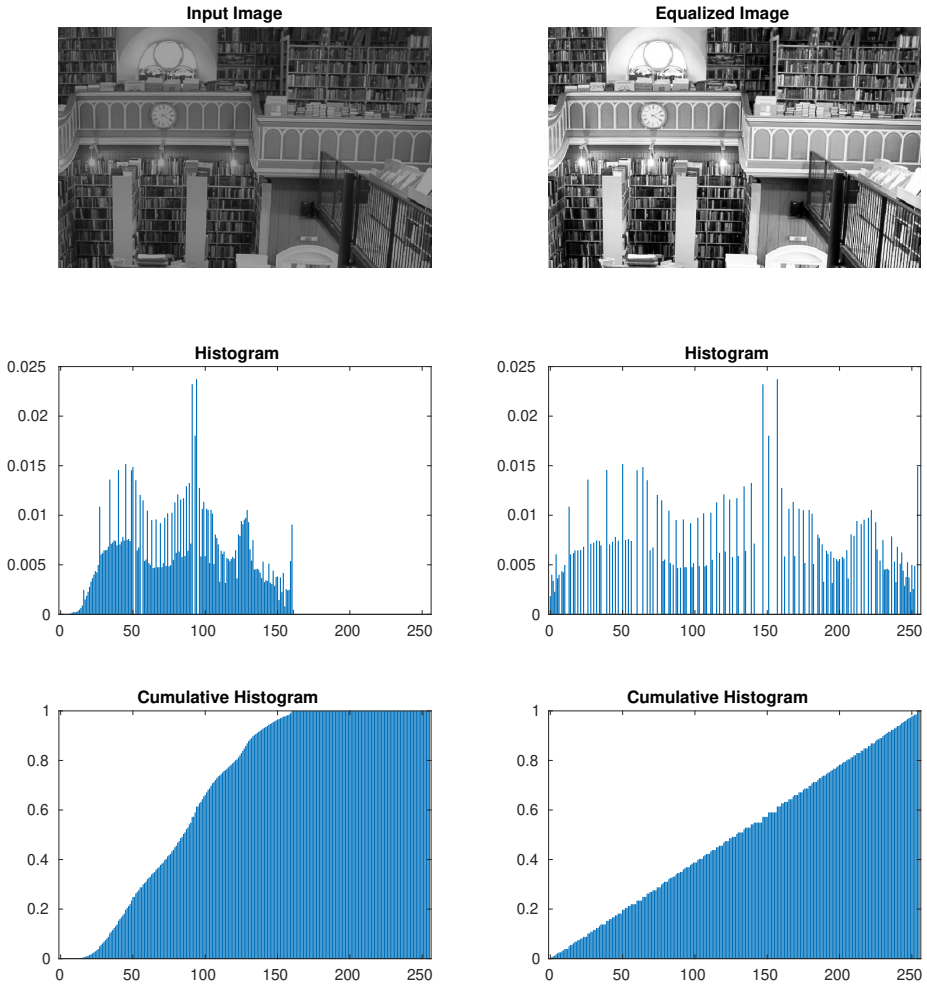

Figure 1: Results of exercise 1

2 Image gradients

The sign of the sobel operator decides whether the result approximates the gradient or the negative gradient. The resulting vector points towards the biggest increase if the sign is positive, therefore if one inverts the sign of both Sobel operators it would point to the biggest decrease. Nevertheless, the length of the gradient vector stays the same.

For the second task one needed to be aware that the y-axis points downwards, although the thoughts behind Sobel operators assumed a "normal" mathematical coordinate system. Therefore we either need to multiply the Sobel operator for the y-direction with -1 or the result I_y calculated with the given kernel. We've chosen the latter approach. The results for the different positions given in task 2 can be found in Table 2. For the positions v_2 and v_4 there is no nice rational fraction times π. Therefore this results are approximately given as a decimal number in radians.

Table 1: Results of exercise 2.2

Position v_i	(I_x, I_y)	$\|\nabla I(v_i)\|$	θ_i
v_1	$(-255, 255)$	360.624	$\frac{3}{4}\pi$
v_2	$(765, 255)$	806.381	0.322
v_3	$(-255, -255)$	360.624	$-\frac{3}{4}\pi$
v_4	$(-255, -765)$	806.381	-1.892
v_5	$(-1020, 0)$	1020	$-\pi$
v_6	$(765, -765)$	1081.873	$-\frac{1}{4}\pi$

The source code used to generate the images shown in figure 2 can be also found in the file sh03ex02.m. For the visualization of the original image we let Matlab rearrange the intensity values over the full range from 0 to 255, therefore the circles look brighter/darker than one would expect from their original intensity values.

```
1  %% Group 2: Dominik Authaler, Jonas Otto
2  close all;
3  clc;
4  clear;
5
6  %% Calculation
7  image = im2double(rgb2gray(imread("../images/circles.png
       ")));
8
9  % Rescale to fit exactly [0, 1]
10 image = image - min(image(:));
11 image = image ./ max(image(:));
12
```

```
13 %min(image(:))
14 %max(image(:))
15
16 sobelX = [1, 0, -1; 2, 0, -2; 1, 0, -1];
17 sobelY = [1, 2, 1; 0, 0, 0; -1, -2, -1];
18
19 I_x = imfilter(image, sobelX, 'replicate', 'conv');
20 I_y = imfilter(image, sobelY, 'replicate', 'conv');
21
22 magnitude = sqrt(I_x .^2 + I_y .^2);
23
24 angles = atan2(I_y, I_x);
25
26 rX = 110:120;
27 rY = 270:280;
28
29 %% Visualization
30 rows = 1;
31 cols = 3;
32
33 subplot(rows, cols, 1);
34 imshow(image);
35 title("original image");
36
37 subplot(rows, cols, 2);
38 imshow(magnitude(rY, rX), []);
39 title("magnitude and direction");
40 hold on;
41 quiver(I_x(rY, rX), I_y(rY, rX), 1);
42 hold off;
43
44 subplot(rows, cols, 3);
45 imshow(gradientColored(magnitude, angles, 0.25));
46 title("image orientation");
47 tightfig;
48 saveas(gcf,'../images/ex02.eps','epsc')
```

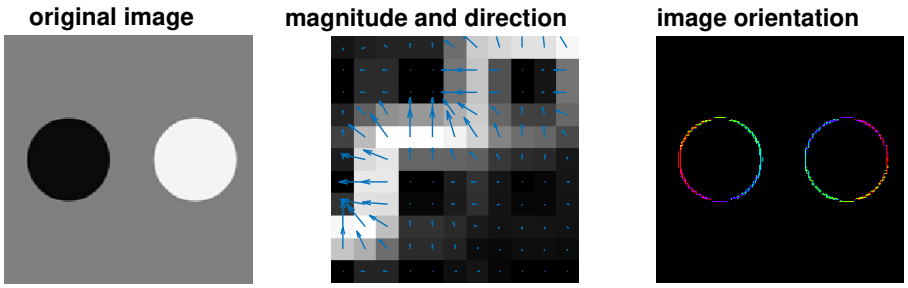

Figure 2: Results of exercise 2

3 Deconvolution

The source code seen below can also be found in file sh03ex03.m. Figure 3 shows the given filters both in the spatial and the frequency domain. Because most of the image in the spatial domain is filled with zero, this plot only shows the range from 1 to 30 in y-direction and 1 to 50 in x-direction. Figure 4 displays the original image together with the blurred image and the two images gained from the deconvolution operation applied on the blurred image for both filters. Comparing the original image with the resulting one of the deconvolution operation with filter 1, there aren't any obvious differences, whereas the difference to second deconvolved image is huge. Thus it seems like the filtered image was constructed using the first filter in a convolution operation.

Important for the abitity to reconstruct an image completely is, that the filter mustn't contain any zero values in the frequency domain, because these values can't be changed any more by the deconvolution operation.

```
%% Group 2: Dominik Authaler, Jonas Otto
close all;
clc;
clear;

%% Load .mat files and calculate
load("../images/Hfreq.mat");
load("../images/Hfreq2.mat");
load("../images/filtered.mat");

Hspat = ifft2(Hfreq);
Hspat2 = ifft2(Hfreq2);

orgImage = imread("../images/bookstore.tif");
filteredFreq = fft2(filtered);

deconvolved1 = ifft2(filteredFreq ./ Hfreq);
deconvolved2 = ifft2(filteredFreq ./ Hfreq2);

%% Visualization
rows = 2;
cols = 2;

figure()
subplot(rows, cols, 1);
imshow(log(abs(fftshift(Hfreq)) + 1), []);
title("filter 1 (frequency domain)");

subplot(rows, cols, 2);
```

```
30 imshow(fftshift(log(abs(Hfreq2) + 1)), []);
31 title("filter 2 (frequency domain)");
32
33 subplot(rows, cols, 3);
34 imshow(Hspat(1:30, 1:50), []);
35 title("filter 1 (spatial domain)");
36
37 subplot(rows, cols, 4);
38 imshow(Hspat2(1:30, 1:50), []);
39 title("filter 2 (spatial domain)");
40 saveas(gcf,'../images/ex03_filters.eps','epsc')
41
42 figure()
43 subplot(rows, cols, 1);
44 imshow(orgImage);
45 title("original image");
46
47 subplot(rows, cols, 2)
48 imshow(filtered);
49 title("filtered image");
50
51 subplot(rows, cols, 3);
52 imshow(deconvolved1);
53 title("deconvolved image (filter 1)");
54
55 subplot(rows, cols, 4)
56 imshow(deconvolved2);
57 title("deconvolved image (filter 2)");
58 saveas(gcf,'../images/ex03_images.eps','epsc');
```

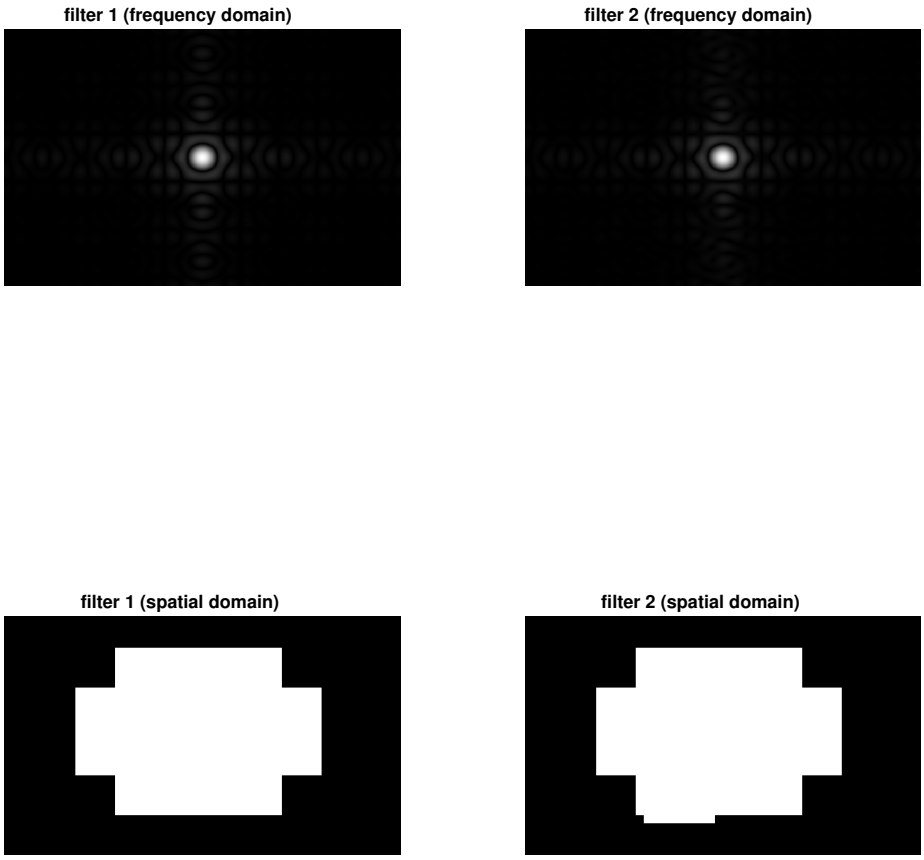

Figure 3: Filters used in exercise 3

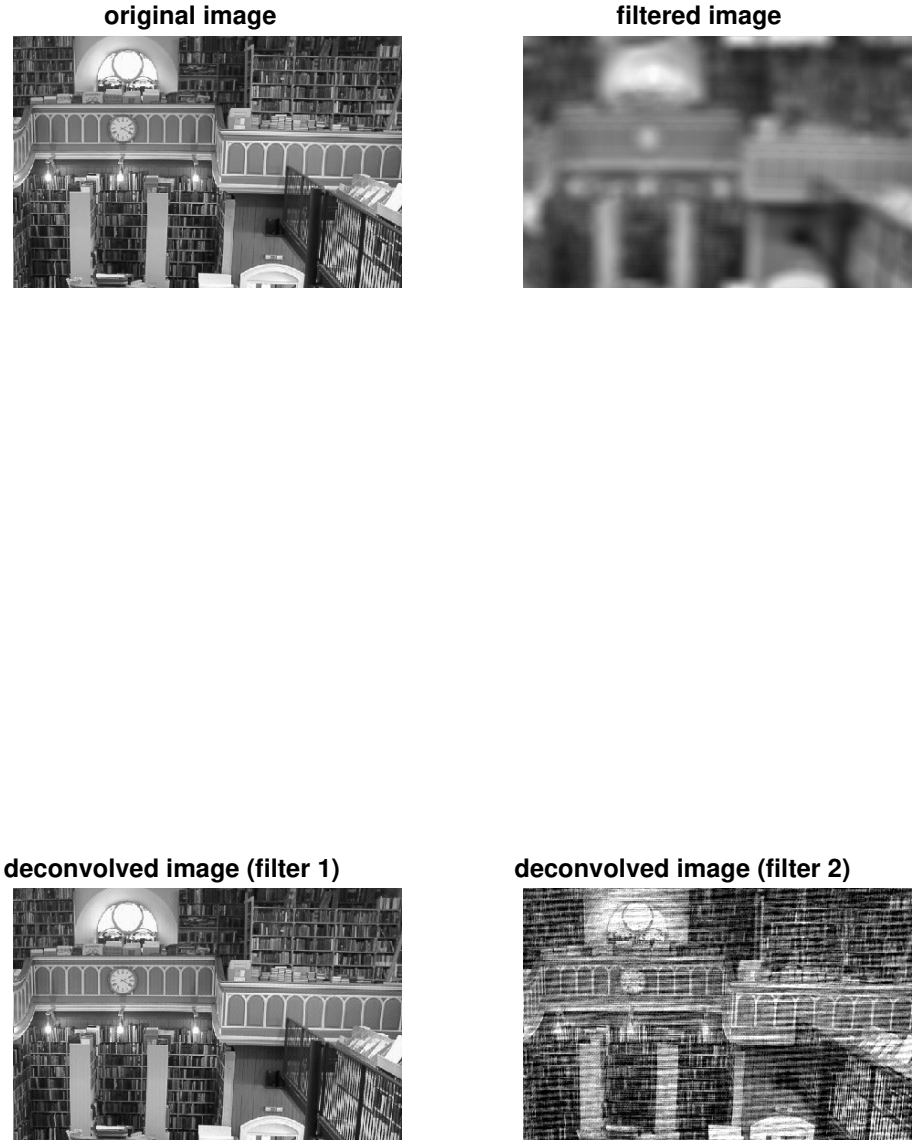

Figure 4: Resulting images in exercise 2