Homework 2

Thomas Buchegger
Introduction to Signal and Image Processing

April 10, 2019

Abstract

This report is based on Homework 2 in the course Introduction to Signal and Image Processing.

Contents

1 Used Methods

1.1 Linear Filtering

Linear filtering is used to modify a picture. This includes enhancing or removing features. Filtering is a so called neighbourhood operation, which means that the value of a certain pixel is based on its neighbourhood pixels.
An example for a linear filter would be a box filter. It calculates the arithmetic means based on its neighbourhood pixels for the new pixel. This leads to a blurring effect.

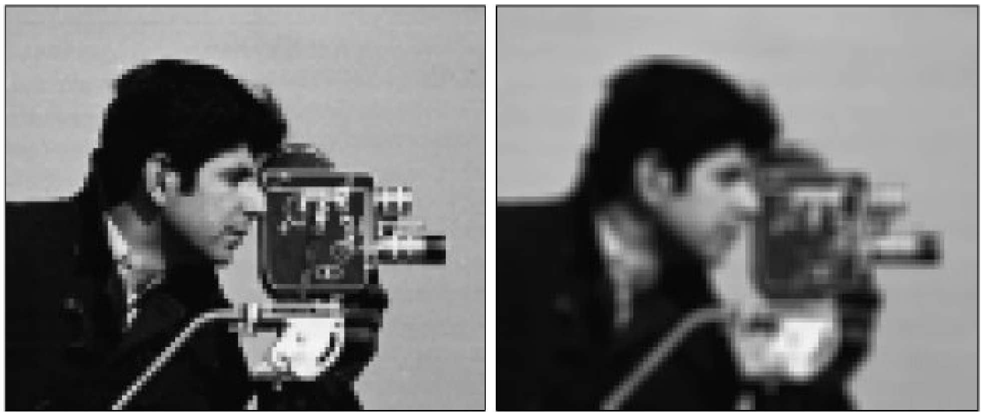

Figure 1: Filtering with a box-filter of size 3×3.

1.2 Finding edges

As the name suggests, the goal is to detect edges in images. One of the most popular edge detection algorithm is canny edge detection.
It consists of four steps:

1. Preprocessing with gauss for example

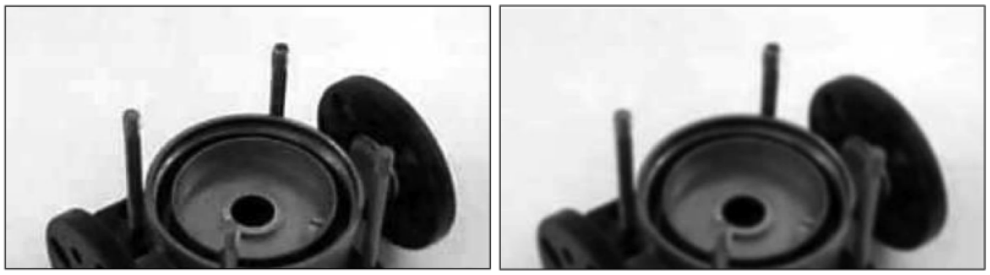

Figure 2: Before and after low pass filter

2. Gradient calculation

3. Nonmax suppression

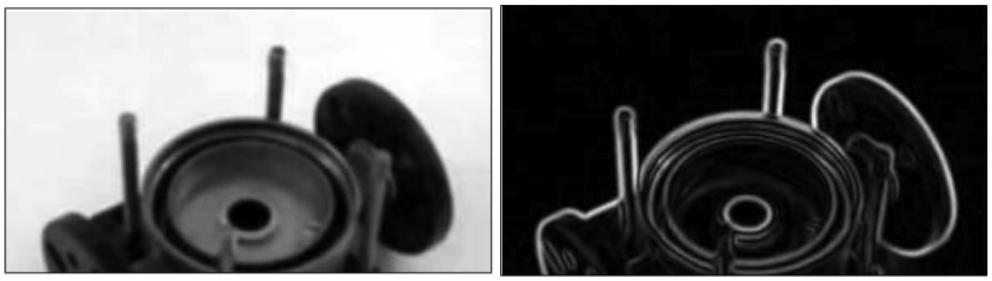

Figure 3: Before and after gradient calculation

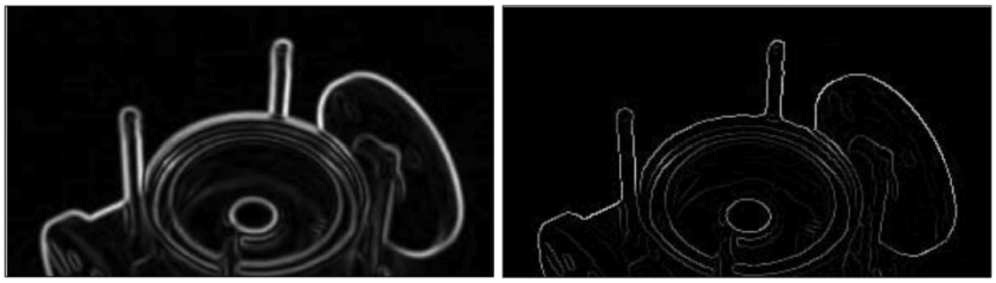

Figure 4: Before and after non max suppression

4. Hysteresis thresholding

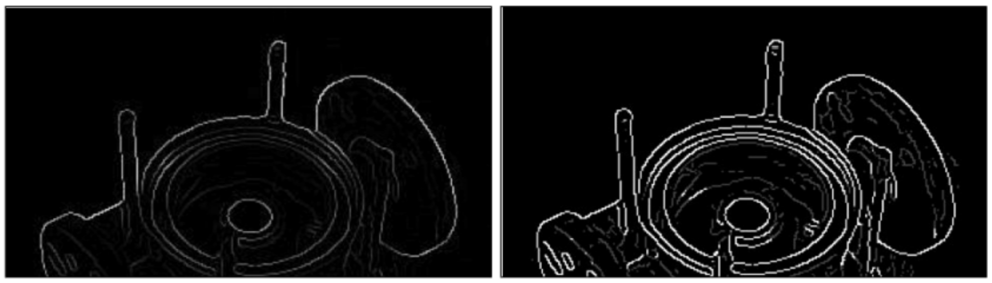

Figure 5: Before and after thresholding

1.3 Corner detection

The goal is to detect corners in images. A corner is defined as an intersection between two edges.

2 Exercises

2.1 Linear Filtering

2.1.1 Boxfilter

The function boxfilter(n) can be summarized as $np.ones((n, n))/(nxn)$.

2.1.2 myconv2

2.1.3 boxfilter with myconv2

2.1.4 gauss1d

2.1.5 gauss2d

2.1.6 gconv

2.1.7 More efficient convolution

2.1.8 Computation time vs filter size experiment

2.2 Finding edges

2.2.1 gradients

Gradient detection is done with a high pass filter. In our excercise this is done with once in x and once in y direction with the help of the *myconv*2 and *gauss*1*d* functions. The value for sigma is 1.

2.2.2 create edge magn image

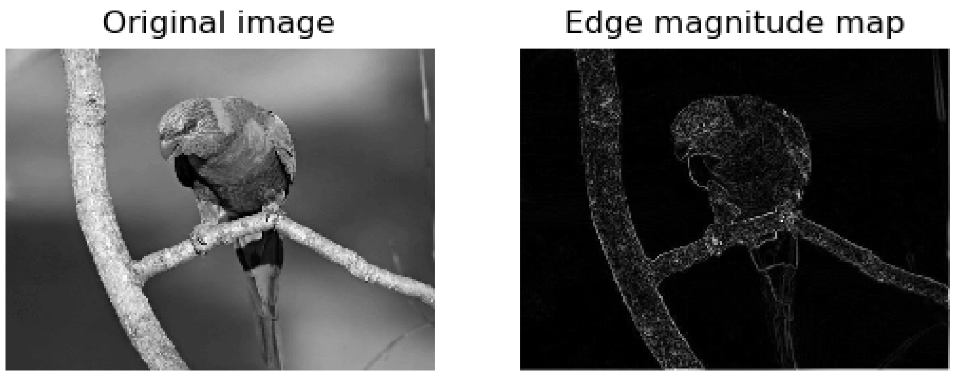

Figure 6: Original image and its magnitude map

2.2.3 make edge map

2.2.4 edge non max suppression

The non max suppression is the last step of the canny edge detector algorithm. This step is needed, bceause edges are usually larger than one pixel, so they get reduced to the pixel with the max. gradient.
This is done by iterating through the picture four times, once in each direction. The currently looked at pixel is then compared to its neighbours. If it has the strongest gradient in [x,y], then it survives. Else it's set to 0.

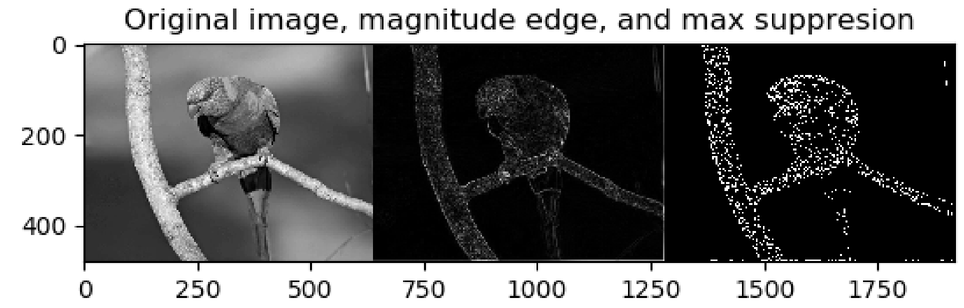

Figure 7: Comparison of original image, image with gradients and image with non max suppression

2.3 Corner detection

2.3.1 myharris

2.3.2 Evaluate myharris

2.3.3 Evaluate myharris rot 45

2.3.4 Evaluate myharris downscaled

2.3.5