Assigment 3 - Harris Corner Detector & Optical Flow

Basak Eskili (11732540) **Lukas Jelinek (11896493)**

1 Introduction

In this assignment, we work on corner and optical flow detection by using Harris Corner model and Lukas-Kanade Algorihtm. Later on, we will implement a tracking function to create a video by using these two methods.

2 Harris Corner Detector

2.1 Question - 1

ⓐ The **harris_corner_detector.m** is implemented according to the given formula in the assignment description. It returns **H** matrix, which is called *cornerness*, also the coordinates of detected corners. See figure 1 and 2

Within the **harris_corner_detector.m** function, we calculate the threshold by the following formula:

$$threshold = mean(mean(H)) * constant$$

The figure 3 shows the comparison between different threshold constants. As can be seen, lower threshold values cause wrong detection, that is because the formula detects unnecessary points in the image. For the further experiment we picked the best value (95) by observing the outputs for different threshold values.

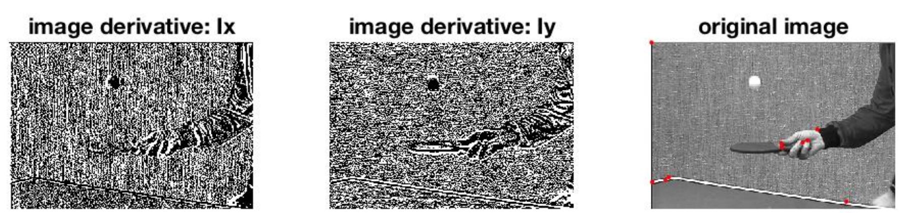

Figure 1: Harris Corner Detector - pingpong

ⓒ We want features to be detected despite geometric or photometric changes in the image: if we have two transformed versions of the same image, features should be detected in corresponding locations. Harris corner detector is rotation invariant. This can be also proven by taking into account eigen values of matrix Q(x,y). During the image rotation the eigen vectors are changing but eigen values stay same. Figure 6-8 shows the results for rotated images in addition to original version.

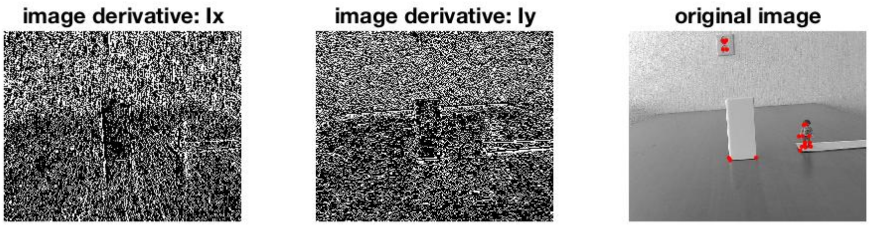

Figure 2: Harris Corner Detector - person toy

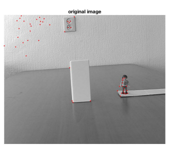

Figure 3: constant = 50

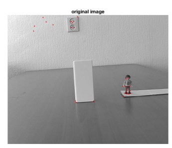

Figure 4: constant = 70

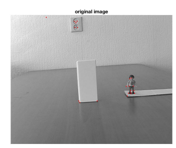

Figure 5: constant = 95

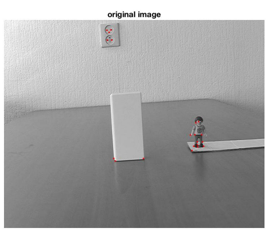

Figure 6: Original image

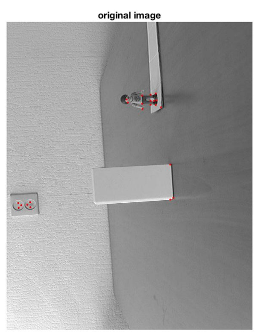

Figure 7: Rotated Image 1

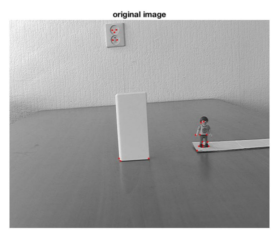

Figure 8: Rotated Image 2

2.2 Question - 2

1

$$H = min(\lambda_1, \lambda_2)$$

2 We are calculating Eigen value decomposition of the local patch only. We are interested in eigenvalues of our matrix Q(x, y) which is taking into account intensities inside of the window with center in pixel x,y. These intensities where created by convolution of image with gradient filter.

3 a) flat region

b) edge

c) corner

3 Optical Flow with Lukas-Kanade Algorithm

Question-1 In this part, we try to estimate the optical flow between these two pairs by using the Lucas-Kanade algorithm. We followed the steps described in the assignment and implemented **lucas_kanade.m** function. See figure 9-14.

Question 2

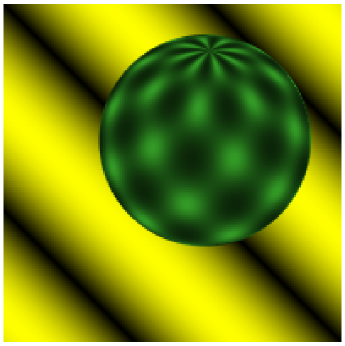

Figure 9: Image 1

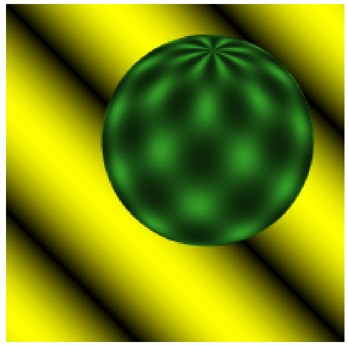

Figure 10: Image 2

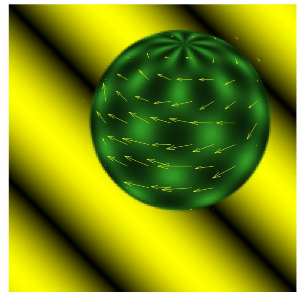

Figure 11: Optical Flow

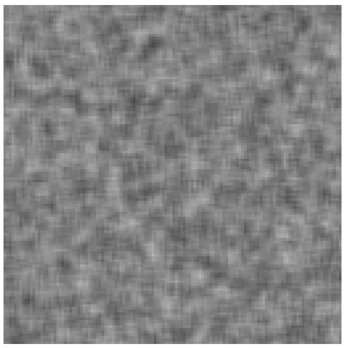

Figure 12: Image 1

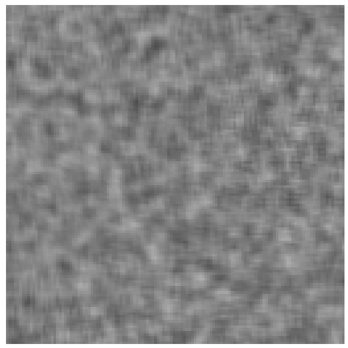

Figure 13: Image 2

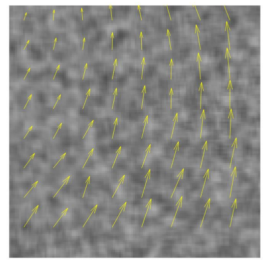

Figure 14: Optical Flow

The Horn–Schunck method of estimating optical flow is a global method which introduces a global constraint of smoothness to solve the aperture problem The Horn-Schunck algorithm assumes smoothness in the flow over the whole image. Thus, it tries to minimize distortions in flow and prefers solutions which show more smoothness. Lucas-Kanade is local and local methods are more robust under noise, while global techniques yield dense flow fields. Though Horn-Schunck algorithm gives a complete solution for optical flow, it takes high computational time because of the iterations and hence resulting in the mathematical complexity. This can be rectified in Lucas-Kanade algorithm by implementing the concept of Least Square method. However, Lucas-Kanade algorithm fails when the inverse does not exist. Flat regions give all the derivatives equals to zero, therefore we can not use Lucas-Kanade in this situation.

4 Feature Tracking

Question 1 In the last part, a basic feature tracking algorithm is implemented, **tracking.m** to observe the motion in the sequence of images and create a sample video. We first located the corner points with Harris Corner Detector, and tracked these points by running Lukas-Kanade algorithm for each image pairs sequentially. Then in order to visualize the motion, we make a video by combining the processed images.

Question 2 When we detect features we can loose information about context. In a case of tracking person in an image with multiple people it is better to continuously track one person in every time frame than detecting features which could be present on both people. The motion tracking without feature detection is more robust in this situation.

5 Conclusion

In this work we have implement Harris corner detector and optical flow tracking. Both method work and produce good results. The result of this work are two videos showing motion tracking. We have observed that Harris corner detector is a simple method which can be quickly computed. This is useful property to use it as a base step of more complicated feature detection.