Assignment 3
Harris Corner Detector & Optical Flow

Nicole Ferreira Silverio
10521933

Cor Zuurmond
10580395

1 Introduction

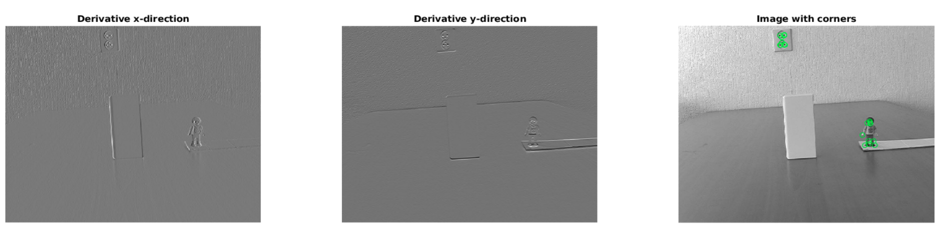
For the course Computer Vision 1 at the Universiteit van Amsterdam (UvA) weekly assignments regarding computer vision have to be submitted. This week the assignment was on 'Harris Corner Detector & Optical Flow'.

For this assignment two different subjects are reviewed and in the last part combined; Harris corner detection and optical flow which combined can be used for 'feature tracking'. In the coming sections of the report each partial subject will be discussed in more detail by answering the questions that were provided in the assignment with references to the provided literature. Firstly, in section 2 Harris corner detector will be elaborated, secondly in section 3 optical flow and lastly feature tracking in section 4.

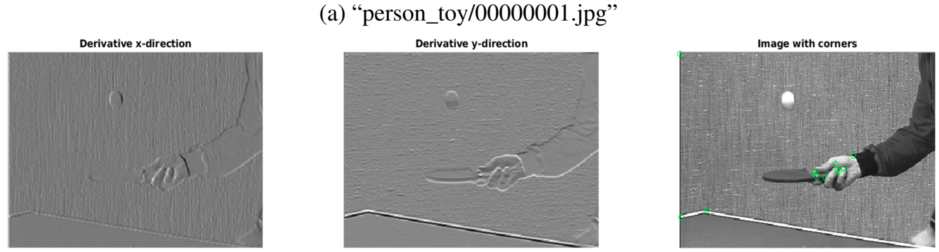
2 Harris corner detector

2.1 Question 1

The Harris corner detector is implemented following the formulas and hints from the assignment. Below in figure 1 the results of "person_toy/00000001.jpg" and "pingpong/0000.jpeg" are shown.



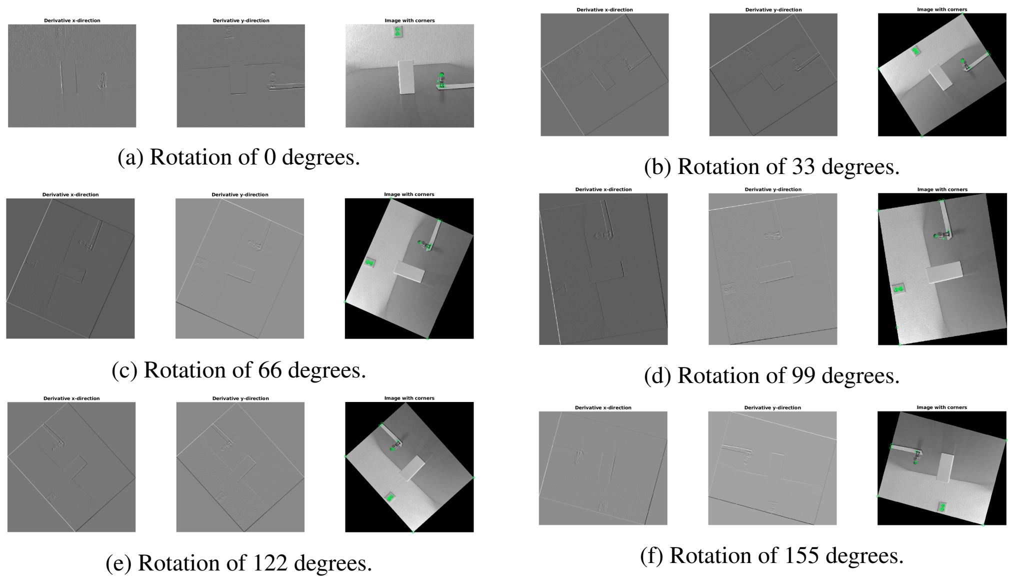
(a) "person_toy/00000001.jpg"



(b) "pingpong/0000.jpeg"

Figure 1: Corners and gradients of "person_toy/00000001.jpg" and "pingpong/0000.jpeg".

In theory the Harris corner detector is rotation invariance. In practice, however, images are discrete signals, therefore rotations may create slight difference between neighboring pixels. In figure 2 corners are detected for different rotations of “person_toy/00000001.jpg”. As you can see the corners on the little Playmobil guy and the power socket are almost in the same locations for all the rotations. The falsely detected corners on the corners of the image and on the platform of the Playmobil guy near the edge of the image occur due to the zero padding when rotating.



(a) Rotation of 0 degrees.

(b) Rotation of 33 degrees.

(c) Rotation of 66 degrees.

(d) Rotation of 99 degrees.

(e) Rotation of 122 degrees.

(f) Rotation of 155 degrees.

Figure 2: Corners and gradients of “person_toy/00000001.jpg” rotated at different angles.

2.2 Question 2

Shi and Tomasi propose a method to define features of which motion can be tracked well[2]. They say that if both eigenvalues in equation 1 are large that “can represent corners, salt-and-pepper textures, or any other pattern that can be tracked reliable”.

$$H = \lambda_1\lambda_2 - 0.04(\lambda_1 + \lambda_2)^2 \tag{1}$$

They say that in practice the matrix Q is “usually well conditioned” if the smaller eigenvalue is “sufficiently large”, giving the following criteria:

$$min(\lambda_1, \lambda_2) > \lambda$$

where λ is a predefined threshold. Shi and Tomasi use a window, therefor eigen decomposition is done over patches. For the following cases the ‘cornerness’ would be defined as:

- Both eigenvalues are near 0: no corner
- One eigenvalue is big and the other is near zero: no corner
- Both eigenvalues are zero: corner

(Assuming the threshold λ is positive and not smaller than ‘large’ and larger than ‘near zero’.)

3 Optical Flow with Lucas-Kanade Algorithm

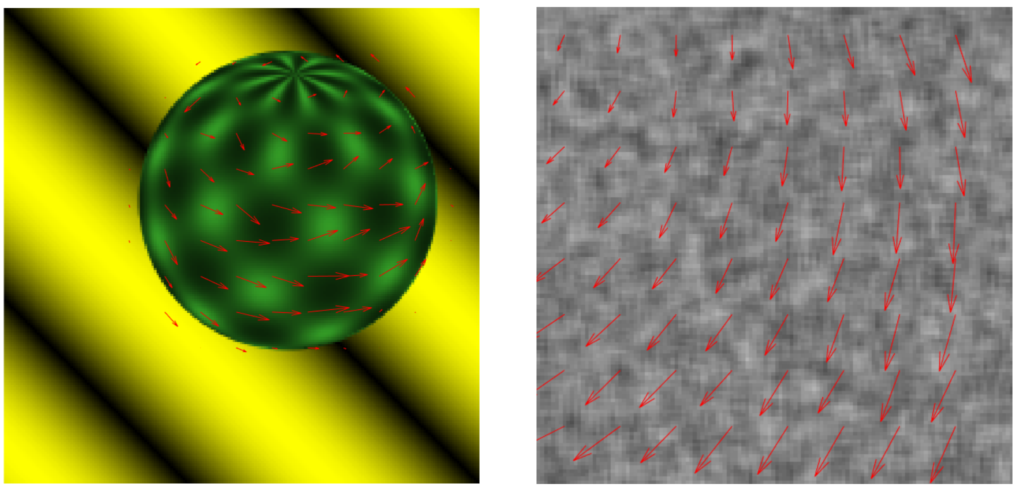
3.1 Question 1

The implementation of the Lucas-Kanada algorithm is as follows:
The input of our function is expected to be a grayscale image, thus color images should be converted to grayscale before being fed to the lucas-kanade function. Then it is checked whether or not interest

points (as row and column coordinates) are given. If no interest points are fed to the function, centrepoints of (non-overlapping) 15x15 windows are calculated, since it is the case that later on the image will be split into non-overlapping windows of size 15x15 pixels. If interest points are fed to the function, the image will be padded. Then, the gradients of the complete image will be calculated.

After all this preprocessing, it is again checked whether or not the function received interest points. If there were no interest points given, the image will be split up into non-overlapping windows of 15x15 pixels of which the gradients are retrieved (since the image gradients were calculated beforehand). Then matrix A, given in the formula in the exercise, is computed with use of the correct window gradients, vector b is calculated by subtracting the corresponding window of the second image from the window of the first image and then v is calculated by formula number 20 in the exercise, also shown in equation 2. If the interest points were given, the window around that point was calculated and the corresponding gradients were retrieved and again A, b and v were calculated according to the same formula as mentioned before (see equation 2).

$$v = (A^T A)^{-1} A^T b \tag{2}$$



(a) Optical flow in sphere image

(b) Optical flow in synth image

Figure 3: Optical flow visualized with red arrows on images 'sphere' and 'synth'

3.2 Question 2

3.2.1 Question 2.1

The Lucas-Kanade algorithm is an example of an optical flow algorithm that operates locally, while the Horn-Schunck method is an example of a global operating algorithm. The Lucas-Kanade method assumes that the optical flow of the neighboring pixels of the pixel that is taken into consideration is constant. Thus, in a local predefined window around the center pixel, all pixels have the same optical flow according to this algorithm. Here, only a local part of the image is taken into consideration at a time, thus this algorithm operates locally. The Horn-Schunck method takes brightness into consideration; it is assumed that the velocity of brightness corresponds to the movement of surfaces in an image and that the brightness pattern varies smoothly overall in the image [1]. Thus the Horn-Schunck method operates globally, since it is assumed that the optical flow is smooth over the complete image, where the optical flow in an image in computed by a global functional, which is then minimized.

3.2.2 Question 2.2

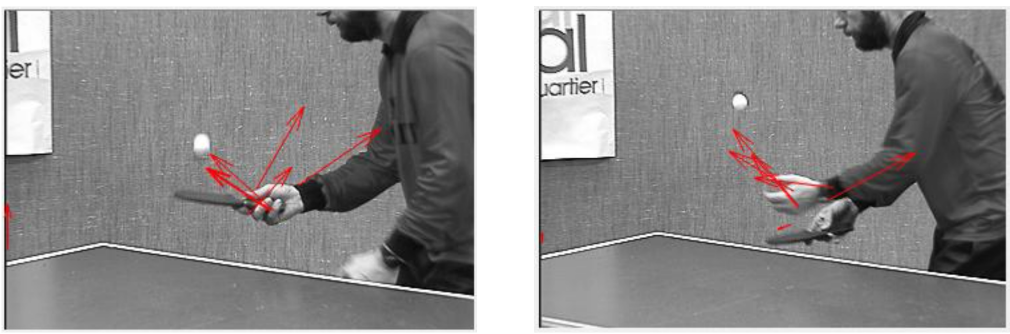
The Horn-Schunck algorithm can deal with flat regions since it takes into account movement of the global image by taking into account movement from nearby regions (pixels), which are not flat, and derives the movement in the flat region from these nearby regions, even though the gradient of the flat region is zero. The Lucas-Kanade algorithm operates locally, thus it can not take into account the movement in the whole image. Because of this, if there is a flat region where (almost all) the gradients are zero, the algorithm can not cope with this and thus can not find movement in a flat region.

4 Feature tracking

4.1 Question 1

Feature tracking is done by combining the Harris corner detector with Lucas-Kanade algorithm for optical flow. The video for "person_toy" is accompanied with this report in the zipfile as "person_toy.avi" and for "pingpong" as "pingpong.avi".

Interesting to note is that in the "ping pong" movie, the hands of the ping pong player overlap each other near the end of the video. From the beginning the interest points are on the right hand which is more in the back, when the left hand overlaps the right hand the interest points 'jump' from one hand to the other, as can be seen in the video and the accompanying screenshots of the video in figure 4. In the ideal situation this should not happen, though it can happen with the Lucas-Kanade algorithm.



(a) Interest points on right hand (b) Interest points on left hand

Figure 4: Screenshots of video pingpong.avi

4.2 Question 2

Feature tracking is needed to keep track of points of interest over time. Simply applying the Harris corner detector on each frame will result different points of interest per frame and thus will not show the correct movement of the interest points. Next to this, if the same interest points would be found in consecutive images, it would take more time and more effort to find the interest points in each consecutive image, connect the corresponding interest points in each image and then calculating the difference of location of each interest point to detect the optical flow.

5 Conclusion

Harris corner is implemented and shown qualitatively accurate results. The threshold has to be adjust to filter for the most "cornery" corners. Shi and Tomasi suggested a slightly different approach, which thresholds the lowest eigenvalue.

In the second part of the exercise the Lucas-Kanade algorithm is implemented and the optical flow in the images sphere and synth is calculated. The Lucas-Kanade algorithm operates locally, thus after examining literature about the Horn-Schunck method, which operates globally, it was found that the Lucas-Kanade algorithm will probably perform worse on flat regions in an image than the Horn-Schunck method.

At last, the Harris corner detector and the Lucas-Kanade algorithm were combined to keep track of features from frame to frame, of which the results can be seen in the provided movies.

References

[1] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[2] Jianbo Shi and Carlo Tomasi. Good features to track. In *CVPR*, 1994.