Computer Vision 1, Assignment 3

Erik Stammes Max Bos Lars Rigter Jan Zuiderveld

March 6, 2019

Introduction

In this assignment we first build two algorithms for one downstream task: the Harris corner detector and optical flow estimation using the Lucas-Kanade algorithm. Together, they are combined to create a feature tracking algorithm, which can track features across sequences of images.

1 Harris Corner Detector

Question 1
See `harris_corner_detector.m` and `harris_demo.m` for the implementation of the Harris corner detector (Harris u. a. (1988)). Figure 1a and Figure 1b show the results of these functions on the person_toy/000001.jpg and pingpong/0000.jpeg images respectively. The algorithm works fairly well, finding most corners, but there are also some false positives and negatives. Figure 1c and Figure 1d show that the algorithm is not entirely rotation invariant. While the most important corners are picked up in all rotations, not all points are in each rotated image. This is due to the fact that the algorithm uses only two directions to calculate the first order Gaussian derivatives.
Question 2
The cornerness metric from Shi and Tomasi is defined as

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

In order to find the smallest eigenvalue, the matrix does not have to be decomposed entirely, the inverse power method can be used to find the smallest eigenvalue.

Internally, when using the Matlab `corner` function with method 'MinimumEigenvalue' (which corresponds to Shi and Tomasi's method), the following cornerness measure is computed:

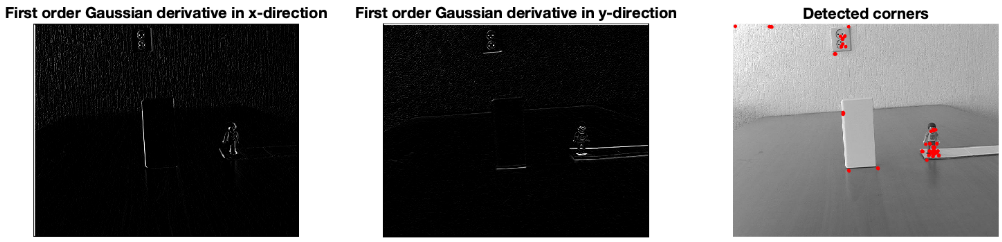
```
cornerness = ((A + B) - sqrt((A - B) .^ 2 + 4 * C .^ 2)) / 2;
```

One can find the internal function definitions by typing `edit functionname`. In this case, that would be `edit corner` and `edit cornermetric`.

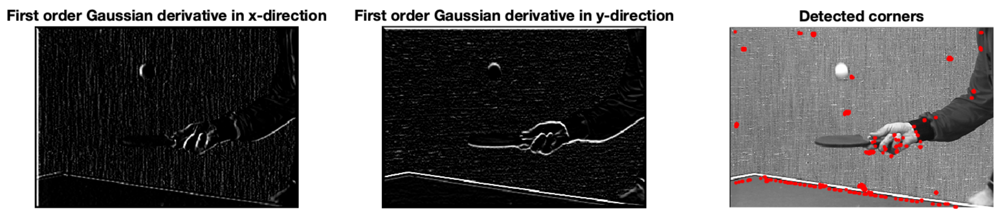
A certain minimum threshold is used to decide when a corner is accepted. In case when both eigenvalues are near zero e.g. below the threshold, the minimum is also close to zero so the corresponding region is uniform. If one of the two is close to zero, the minimum will also be close to zero and the corresponding region is an edge. If both eigenvalues are big, the minimum will be big as well and if it exceeds the threshold the corresponding region is a corner.

2 Optical Flow with Lucas-Kanade Algorithm

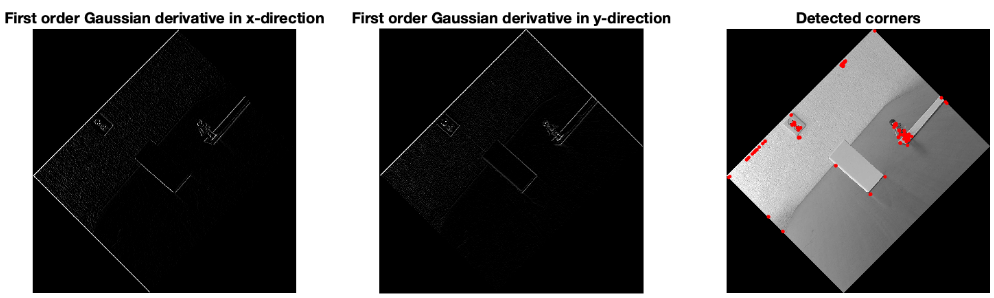
Question 1
See `lucas_kanade.m` and `lucas_kanade_demo.m` for the implementation of the Lucas-Kanade method for Optical Flow estimation. Figure 2a and Figure 2b show the results of these functions on two pairs of images, synth1.pgm, synth2.pgm; and sphere1.ppm, sphere2.ppm. The algorithm works pretty good, generally computing optical flow vectors in the right direction, but there are some situations (especially in the sphere



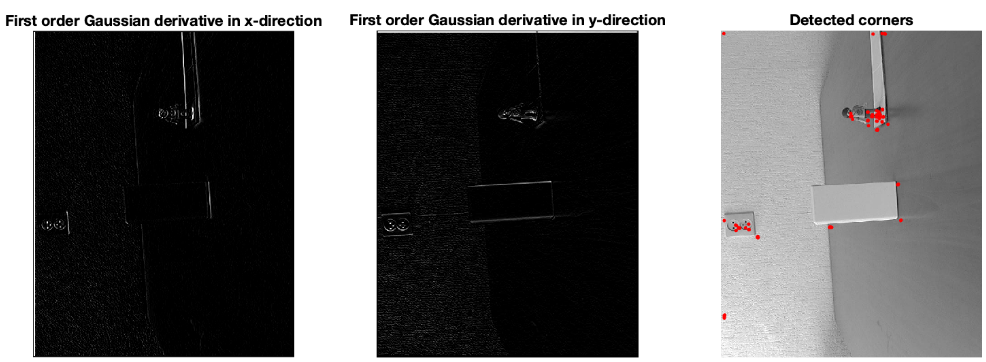
(a) First order Gaussian derivatives in x and y direction and Harris corner detected points on person_toy/000001.jpg



(b) First order Gaussian derivatives in x and y direction and Harris corner detected points on pingpong/0000.jpeg

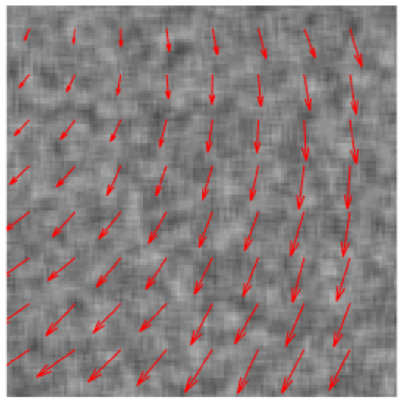


(c) First order Gaussian derivatives in x and y direction and Harris corner detected points on 45 degree rotated person_toy/000001.jpg

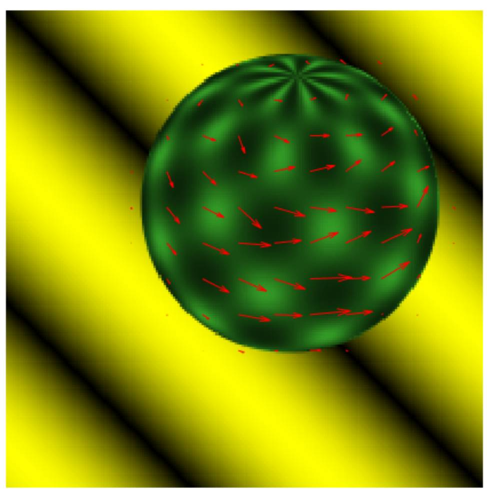


(d) First order Gaussian derivatives in x and y direction and Harris corner detected points on 90 degree rotated person_toy/000001.jpg

Figure 1



(a) Optical flow vectors of non-overlapping 15x15 pixel regions estimated using Lucas-Kanade Algorithm on the image pair synth1.pgm, synth2.pgm



(b) Optical flow vectors of non-overlapping 15x15 pixel regions estimated using Lucas-Kanade Algorithm on the image pair sphere1.ppm, sphere2.ppm

Figure 2

pair) where some incorrect estimations are made. Analysis is done on black and white converted images, because the separate r, g and b channels showed similar results. Out of bound windows are ignored.

Question 2

The Lucas-Kanade technique operates at a local scale by taking image patches of a certain size (in our case 15x15 pixels) and computing the flow in each of these patches (Lucas u. a. (1981)). The Horn-Schunck technique operates globally, minimizing an error function that integrates all pixel values and a regularization term (Horn und Schunck (1981)). This regularization term is a global constraint on the smoothness of the image, which makes it more invariant to the aperture problem.

Often local methods are more robust under noise, while global techniques yield dense flow fields Bruhn u. a. (2005).

The Horn-Schunk method deals with flat regions by calculating error functions over the entire image, and thus including global information on the derivatives. The Lucas-Kanade method deals less well with flat regions since it calculates optical flow based on a local patch. Because of these characteristics, Horn-Schunk behaves better under flat regions in an image than the Lucas-Kanade method.

3 Feature Tracking

Question 1

See `tracking.m` and `tracking_demo.m` for the implementation and a demonstration of the tracking feature points using a hybrid between Harris Corner and Lucas-Kanade. The demonstration generates videos including the tracking of the corner feature points. For each frame in the given sequence of images, the optical flow from the previous image is calculated with a window size of 16. For each corner point, the patch in which the point located is calculated, then the point position is updated by incrementing the x-position with the u-value for the belonging patch, times a constant, and the y-position with v-value for the belonging patch. The mentioned constant was found by tweaking.

The generated videos for the "person_toy" and "pingpong" images can be found in the source code under the names of `person_toy.avi` and `pingpong.avi`.

Question 2

We need tracking of initial corner points if we want to track the position of a certain object within a stream of images instead of just locating objects within each image. If you want to track initial corner points by feature detection for each frame, implementing functionality for comparing calculated corner feature points between two images and matching two corners as the same corner is more prone to errors than tracking corner points using an optical flow map. Using an optical flow map between two images, one can use the calculated velocity for the region in which the original corner point is located to calculate the new location of the corner point. Detecting features for each and every frame would result in loss of the location of the initial corner points, so the feature points then have to be matched using some similarity process if you still want to track initial corner points.

4 Conclusion

In this assignment we experimented with computer vision methods such as feature detection using the Harris Corner detector, finding motion changes between two consecutive images using the Lucas-Kanade algorithm for optical flow estimation, and combining the two mentioned methods for feature tracking within a sequence of images. The obtained results were insightful to us, we learned about the definition of cornerness from multiple resources, how global and local optical flow methods differ in advantages and drawbacks, and how one can track features using the optical flow between two consecutive images.

4.1 Contribution

Everyone contributed equally. While the questions were divided amongst the team members, we cooperated on most questions and all came together to discuss the final result and gave feedback where needed. We worked on this together in the same room at the same time at the university. We all agree with the final submission.

References

[Bruhn u. a. 2005] BRUHN, Andrés ; WEICKERT, Joachim ; SCHNÖRR, Christoph: Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. In: *International journal of computer vision* 61 (2005), Nr. 3, S. 211–231

[Harris u. a. 1988] HARRIS, Christopher G. ; STEPHENS, Mike u. a.: A combined corner and edge detector. In: *Alvey vision conference* Bd. 15 Citeseer (Veranst.), 1988, S. 10–5244

[Horn und Schunck 1981] HORN, Berthold K. ; SCHUNCK, Brian G.: Determining optical flow. In: *Artificial intelligence* 17 (1981), Nr. 1-3, S. 185–203

[Lucas u. a. 1981] LUCAS, Bruce D. ; KANADE, Takeo u. a.: An iterative image registration technique with an application to stereo vision. (1981)