Assignment 3 Harris Corner Detector and Optical Flow

12055980, 12153605, 12202770

October 2021

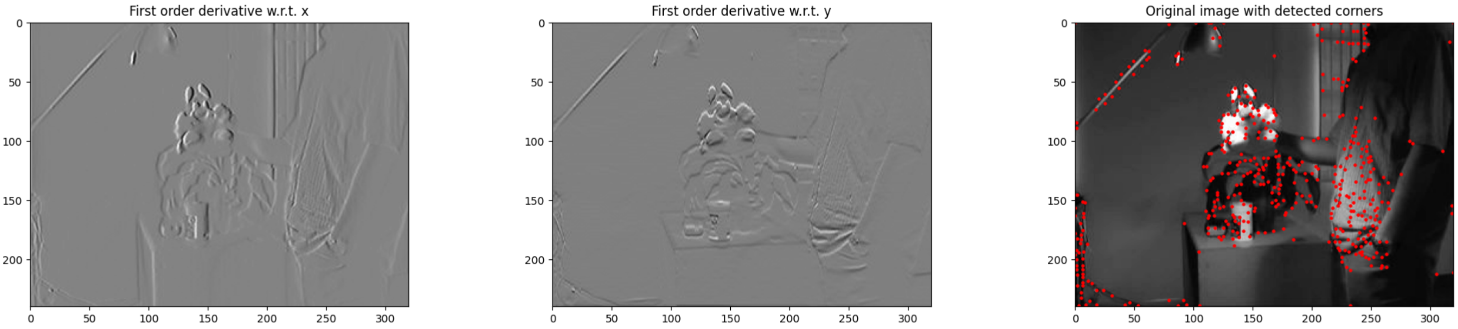
Introduction

This report will look at the Harris Corner Detector, Optical Flow and Feature Tracking. In the Harris Corner Detector, the algorithm will be programmed. In the Optical Flow part, the Lucas-Kanade Algorithm will be programmed. Ultimately in the Feature Tracking part, the Harris Corner Detector and the Lucas-Kanade algorithm will be combined to create a video of tracking the features in the frames.

1 Harris Corner Detector

Question 1

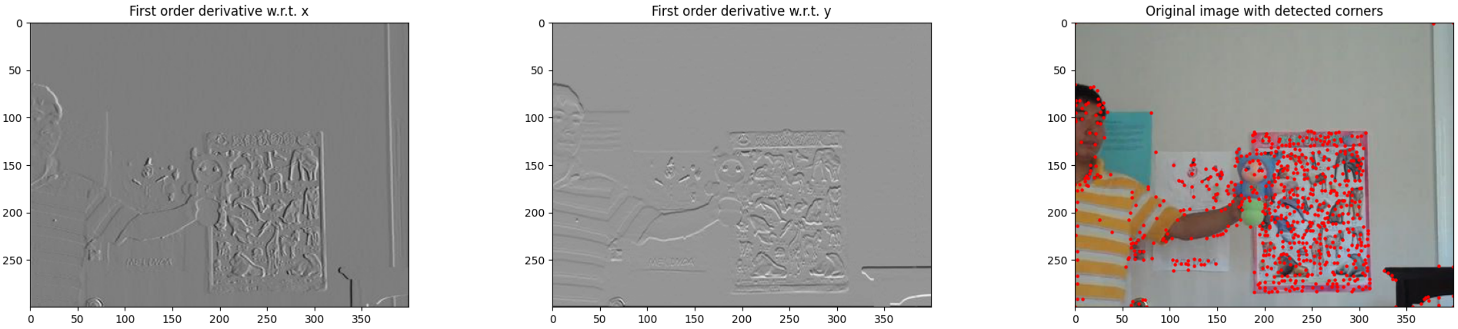
2 Setting a low threshold will result in a lot of features (sometimes in 'flat' areas), which are close to each other. To limit an abundance of features, the threshold was set to 10000 for both images.


First order derivative w.r.t. x
First order derivative w.r.t. y
Original image with detected corners
0
50
100
150
200
250
300

(a) (b) (c)

Figure 1: Results on the toy image

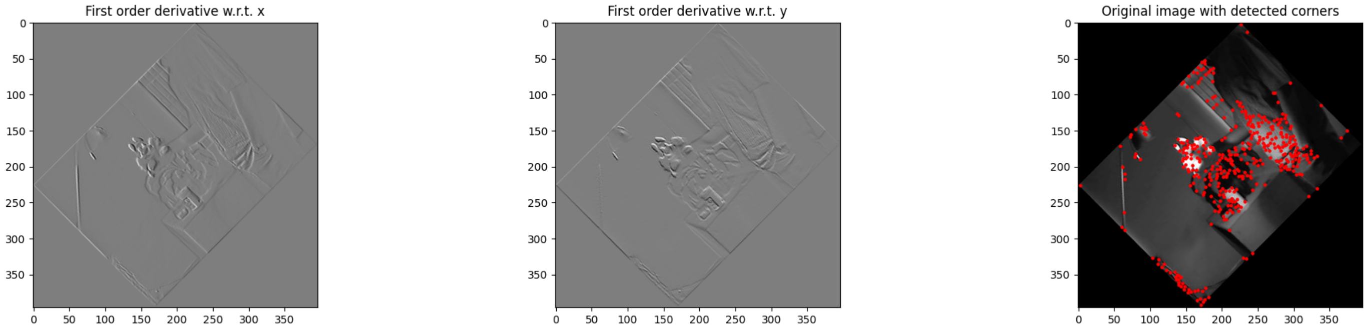
Figure 1 shows the results of the Harris Corner Detector applied to the toy image. As can be seen, no invalid corners are show in flat areas and even with a threshold of 10000, a great number of features remain.


First order derivative w.r.t. x
First order derivative w.r.t. y
Original image with detected corners
0
50
100
150
200
250
300
350

(a) (b) (c)

Figure 2: Results on the doll image

Figure 2 shows the results of the Harris Corner Detector applied to the doll image. Again there are no features in 'flat' areas. As can be seen, the highest concentration of features can be found in the poster on the wall next to the doll.

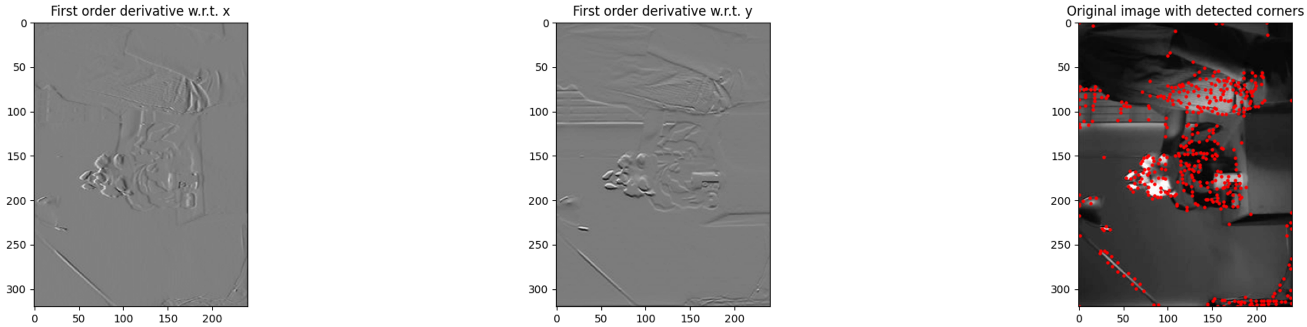


(a)

(b)

(c)

Figure 3: Results on the 45 degree rotated toy image



(a)

(b)

(c)

Figure 4: Results on the 90 degree rotated toy image

3

As can be seen in figures 3c and 4c, the Harris Corner Detector finds almost the same features after rotating the image as in the original image (figure 1c), where there is some loss in features in the lamp after rotating the image with 45 degrees. This means that the Harris Corner Detector is rotation-invariant. This can also be argued without looking at the images. The only filter used in the algorithm is a Gaussian filter, which is symmetric and thus rotation-invariant. Because this is the only filter used, the algorithm will also be rotation-invariant.

Question 2

1. Shi and Tomasi define their cornerness by: $H = min(\lambda_1 \lambda_2)$. Where $\lambda_1 \lambda_2$ are the eigenvalues like in the Harris Corner Detector.

2. It is Translation invariant, because λ does not change with changing location.
   It is Rotation invariant, because λ does also not change when rotating the image.
   It is not Scale invariant, because λ does change when we scale the image changes.

3. (a) This gives $H = min(\lambda_1 \lambda_2)$ with the eigenvalues $\lambda_1 \lambda_2$ near zero. So the value H will be zero.

   (b) This means that one λ is still near zero. Given the formula $H = min(\lambda_1 \lambda_2)$, H will still be zero. It will take the lowest valued λ.

   (c) This is the only time that H will not be zero. Because both λ_1 and λ_2 are big, H will also be big.

2 Optical Flow - Lucas-Kanade Algorithm

Question 1

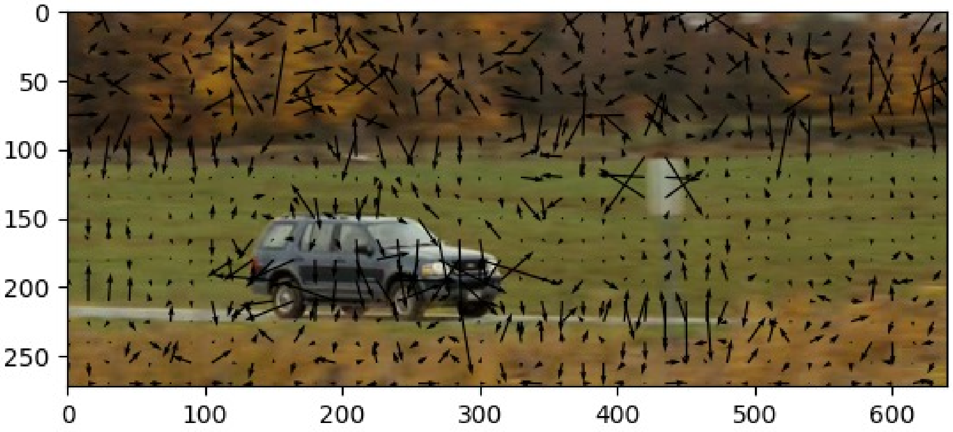


Figure 5: Optical Flow

Question 2

1. The Horn-Schunck method is a global method to estimate the optical flow. It uses a global constant to solve the aperture problem called smoothness.
   The Lucas-Kanade method assumes the optical flow is constant in a local neighbourhood of pixels that are considered. So it operates locally, thus around the center pixel.

2. The Horn-Schunck method does sense the optical flow when the local region is uniform, because it has the global smoothness constant.
   The Lucas-Kanade method cannot estimate the optical flow when the local region is uniform. This is because it does not have a global sense of the image.

3 Feature Tracking

Question 1

The Harris Corner Detector and Lucas-Kanade Algorithm are combined to create the tracking algorithm. The video's created by this algorithm are included. The files are called toy.mp4 and doll.mp4.

Question 2

Feature tracking is a lot less computationally expensive than detecting features for each and every frame. By using feature tracking, it is possible to track the changes in the different frames a lot easier than detecting features. Feature tracking also shows the way that each feature moves in the new frame, with detecting features this is not always possible.

Conclusion

In the end we learned how the Harris Corner Detector and the Lucas-Kanade Algorithm works. We also learned about the alternative for the Harris Corner Detector by Shi and Tomasi and the alternative for the Lucas-Kanade by Horn-Schunck. In the end we used these algorithms to create the feature tracking algorithm. By applying the algorithms on a sequence of images they also can be used for videos.