Universiteit van Amsterdam

&

Lab Assignment 3

October 5, 2021

Students:
Name: Nicole Sang-Ajang
UvAnetID: 10734155

Name: Haohui Zhang
UvAnetID: 14124262

Name: Xiaoyang Sun
UvAnetID: 13825453

Affiliation:
Master Artifical Intelligence

Lecturer:
Dr. Shaodi You

Course:
Computer Vision 1

Course code:
52041COV6Y

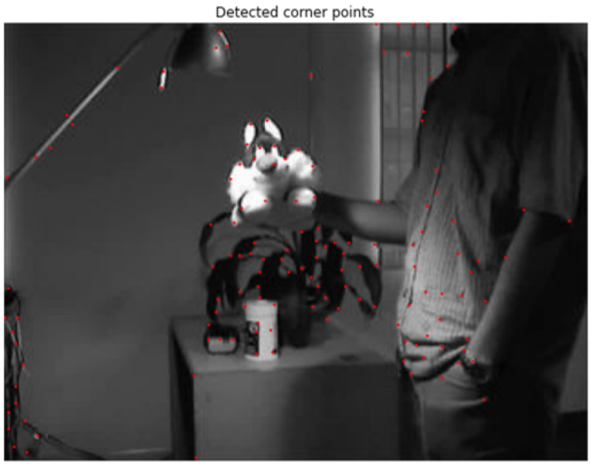
Introduction

[4] This assignment introduces the principle and algorithm in feature detecting and feature tracking mainly separate into two parts Harris Corner Detector and Optical Flow. In the first section, we implement the Harries Corner detector algorithm to detect corners to find features. We also compare it with a similar corner detection algorithm the Shi-Tomasi Corner detector. After that, we explore the Lucas-Kanade algorithm to compute the optical flow and compare it with Horn-Schunck method. At last, we combine Harries Corner detector and Lucas-Kanada algorithm to tract the corners in the real-world images.

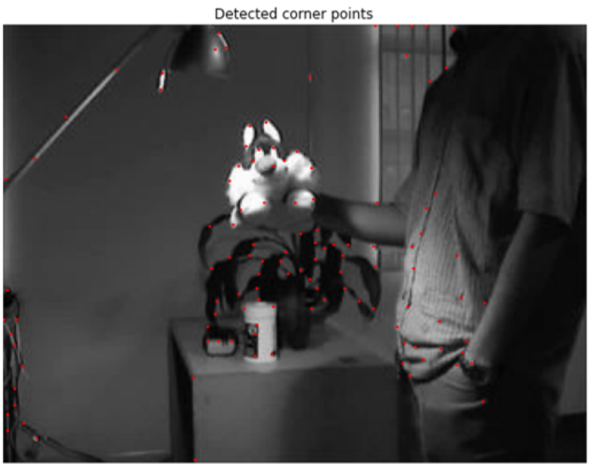
1 Harris Corner Detector

1.1

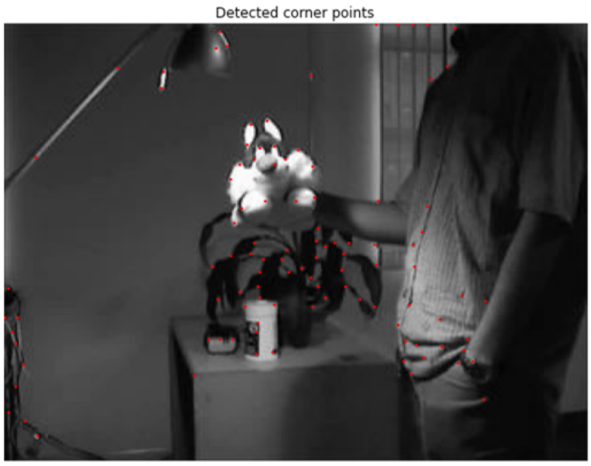
(a) T



(b) T



(c) T



(d) T

Figure 1: T

Question 1.1

Question 1.2

In Harris Corner Detector algorithm, We implemented 3 procedures. First is Gaussian Filter Scale and Kernel size k, after adjustment, we chose σ=1.5 and k=5. Second procedure is window size n, which refers to window size to search for local maxima, after adjust with n=3,5,7,9, we found n=9 shows the best result.

The image gradients Ix and Iy are shown in Figure 1 and Figure 2, as well as the original image with the corner points detected. From the gradient images we can see that only vertical or horizontal textures remain in Ix and Iy respectively.

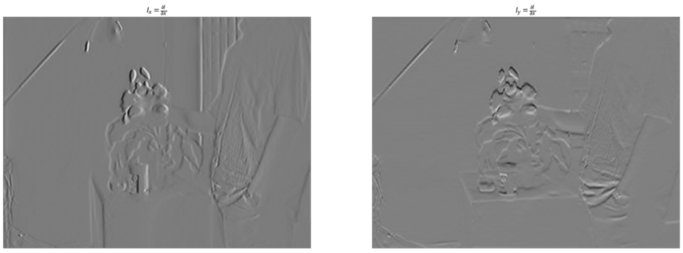


Figure 2: Toy image derivatives Ix and Iy

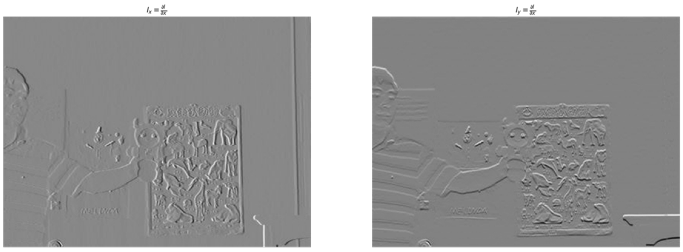


Figure 3: Doll image derivatives Ix and Iy

Figure 3 shows the influence of different thresholds in our experiment on Harris corner detection. Take toy as an example, we start the experiment from threshold 10000. Obviously, it can be seen that the effect of the threshold of 10000 in the experiment is not ideal, and the corner points of the clothes and on the window are detected. With the increase of the threshold, the performance of Harris corner detection improved. Therefore, for toy, the threshold of 150000 is a ideal parameter.

For dolls, since the composition of the posters on the wall is more complicated, we start the experiment with a threshold of 50000. Similar to the experiment process of toy, the effect of the experiment can be observed by changing the threshold. When the threshold exceeds 100000, some unnecessary corners in other areas disappear, but when the threshold exceeds 150,000, the existing corners disappear. In the range of [100000,150000], we chose 100000 as a reasonable threshold.

Question 1.3 Harris corner point detection is a rotation invariant algorithm. That is because when the image is rotated, the shape remains the same, which means that the feature values remain the same. The only change in rotation is the direction of the gradient.

As can be seen from the diagram, when the image is rotated by 90 degrees, the result is exactly the same as when the image is not rotated. However, when the image is rotated by 45 degrees, there are some deviations from the original results.There are some new false positives along the edge of images because of 0 padding.[2]

The rotated image has some blank areas at the edges of the image, so that the pixels of the neighbours at the edges of the image are changed by the rotation. The pixels of the neighbours at other locations have not been changed by the rotation and therefore the result remains the same as the original image.

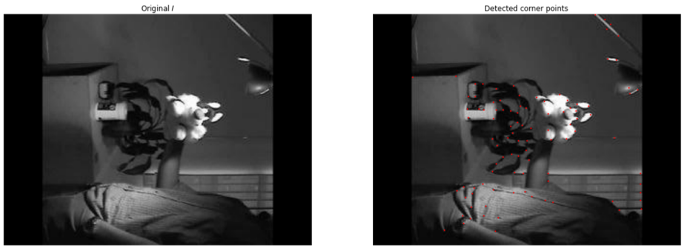


Figure 4: Toy image rotated 90 degrees, threshold remains 150000

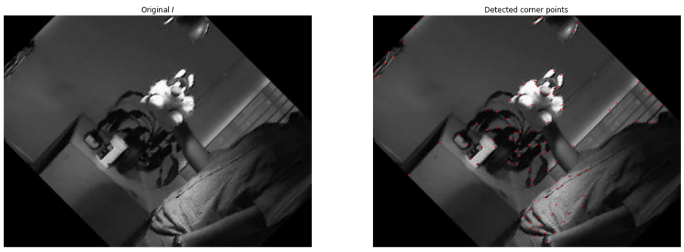


Figure 5: Toy image rotated 45 degrees, threshold remains 150000

1.2

Question 2.1 Shi-Tomasi corner detection is based on much the same principle as Harris corner detection, except that the final discriminant is chosen by selecting the smallest of the features.

$$R = min(\lambda 1, \lambda 2) \tag{1}$$

If the scoring exceeds the threshold, we consider it a corner point. We can plot it into the λ_1, $\lambda_1 2$ space and we get the following diagram.Only if 1 and 2 are both greater than the minimum value are they considered corner points (green areas).

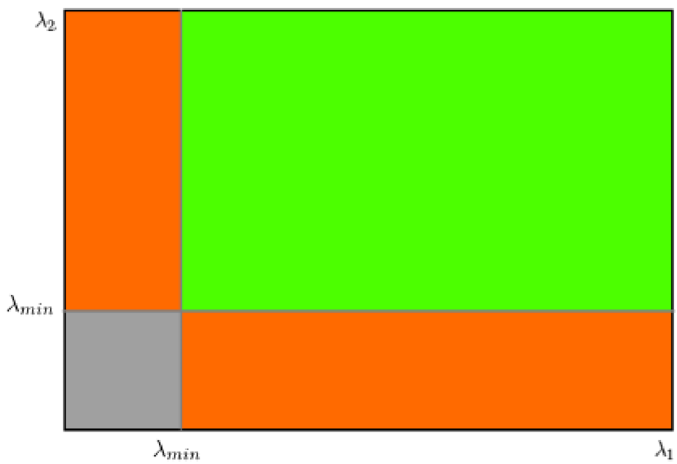


Figure 6: Shi-Tomasi Corner Detection Space

Question 2.2 The Shi-Tomasi algorithm satisfies translation invariance, rotation invariance, but not scale invariance. Translation and rotation do not change the shape of the image, which means that the eigenvalues remain the same as before. The only change occurs in the direction of the derivatives. However, as shown in Figure 6, when the scale changes, the edge is detected as a corner with the same window size, which means that the feature values are different from the original scale.[3]

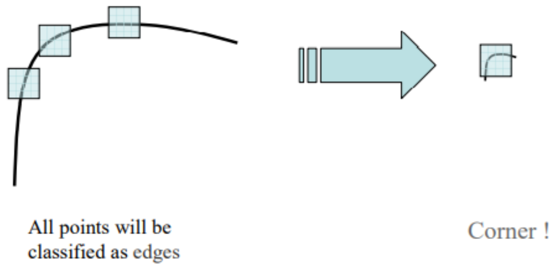


Figure 7: Shi-Tomasi Algorithm Scale Invariance[3]

Question 2.3

(a) When both eigenvalues are near 0, taking the minimum of the two, the cornerness H would be near 0 as well, it will not be classified as a corner.

(b) When only one of them is close to zero whereas the other one is big, the cornerness will still be very small, because of the algorithm, the minimum is taken. Thus, it will not be classified as a corner either.

(c) when both the eigenvalues are big, the cornerness will be big too, indicating a corner point.

2 Optical Flow - Lucas-Kanade Algorithm

2.1

This section is named optical flow, which is defined as the apparent motion of pixels, surfaces, corners or regions between two frames as a consequence of object movements captured by the camera. There are multiple methods to compute the optical flow and in the first part, we will focus on the implementation of the Lucas-Kanade Algorithm. The constraint of this algorithm is that the optical flow must be constant in a local neighborhood of a pixel. Following by the given mathematical formula, we applied the algorithm on the given coke and car images.

This algorithm is separated into two main sections. The first section is the pre-processing of the 2 frames. According to the request, the input images must be divided into non-overlapping regions. Givem a fixed region width, we must check whether the frame can be properly divided. If not, we just crop the image and remove the extra region which border with is not satisfied. Meanwhile, because the optical flow assume brightness constancy which is the image values remain constant over time, we converted the colored images to gray-scale images.

The second section mainly focusing on computing the local image flow vector v. We first figure out the partial derivatives I_x and I_y of image I for each pixel and difference of partial derivatives I_t b. We can get the

v through the resolution of:

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

The results are as followed.

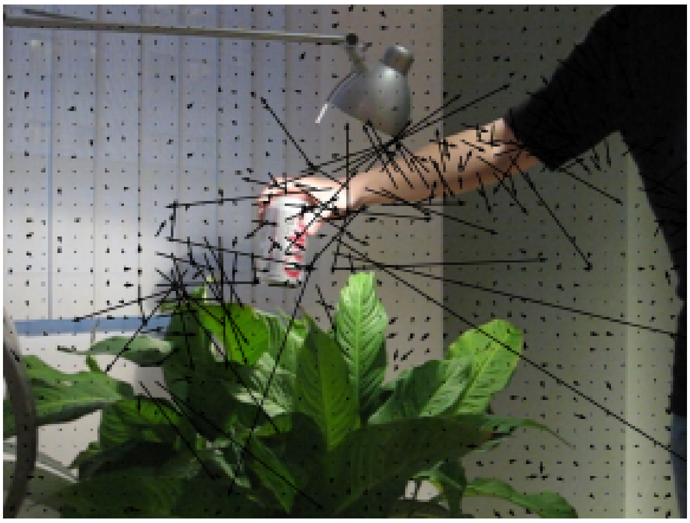


Figure 8: Coke motion tracking

Figure 9: Car motion tracking

2.2

Lucas-Kanade is only one of the differential optical flow estimate methods and there are more methods in the literature. In this section, we will compare the Horn-Schunck method with Lucas-Kanade.

Question 2.1 One of constraint of Lucas-Kanade is spatial consistency, that is, adjacent pixels in the previous frame are also adjacent in the next frame. The same optical flow and movement is shared by neighboring pixels. This is a unique assumption of the LK optical flow method and that's why Lucas-Kanade algorithm is a local method.

On the contrast, Horn-Schunck optical flow algorithm uses a global method to estimate the dense optical flow field of an image. Different from the LK optical flow constraint, the HS algorithm adds a constraint of smoothness. The pixels belonging to the same object in the scene form the optical flow field vector should be very smooth, and only at the boundary of the object will there be a sudden change in optical flow. This boundary parts only occupies a small part of the image. On the whole, the optical flow field of the image should be smooth and that is reason why the Horn-Schunck algorithm is a globel differential method.

Question 2.2 The Lucas-Kanade algorithm cannot deal with the flat regions. The gradient of the image will vanish and be close to 0, and matrix A only contains 0. In this case, $(AA^T)^(- 1)$ cannot be calculated. As for Horn-Schunck method, it is possible to compute the optical flow for flat region for that Horn-Schunck is a global method. All pixels in the image will make a contribution to the optical flow of one pixel and fill the missing motion information in local regions. The filling-in effect may produce more reliable information in flat regions by averaging the less reliable information from all surrounding high gradient areas. However it will also cause false motion detection. [1]

3 Feature Tracking

3.1

Question 1

Question 2

3.2

Question 2

4 Conclusion

This assignment gave us an in-depth understanding of processing feature detection and optical flow tracking. First, we implement the Harris Corner detector to detect interests points for our feature tracking task. To estimate the optical flow of two frames, we implement the Lucas-Kanade algorithm, w, which only operates locally and may not distribute the optical flow in a flat area. Finally, we combining those two methods to perform video tracking.

References

[1] Weickert Bruhn. *Lucas/Kanade Meets Horn/Schunck: Combing Local and Global Optic Flow Methods.* International Journal of Computer Vision, 2005.

[2] Shengli Cai and Huiqing Zhang. “Image rotation measurement based on Harris corner detection”. In: *Computer Measurement and Control* 19.001 (2011), pp. 30–32.

[3] “courses.cs.washington.edu,courses”. In: *courses.cs.washington.edu/courses/cse576 /06sp/notes/HarrisDetector.pdj* ().

[4] Richard Szeliski. *Computer vision: algorithms and applications.* Springer Science & Business Media, 2010.

Appendices

A Results