Lab Report 3

Longxiang Wang, Jordan Earle, Ruihan Sun

Introduction

Corner detection and optical flow are important for many computer vision tasks, such as motion tracking ,image alignment, object recognition, etc. In the first part, Harris corner detector and its variant called Shi-Tomasi corner detector are explored. In the second part, Lucas-Kanade is implemented to do optical flow estimation. Finally, corner detector and optical flow estimation are utilized to achieve feature tracking across different frames.

Harris Corner Detector

Question 1

The cornerness is Harris corner detection algorithem is defined as

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

where λ_1 and λ_2 are 2 eigenvalues of $Q(x, y)$, k is an empirical coefficient, which takes the value of 0.04 in our case. $Q(x, y)$ is given by

$$Q(x, y) = \sum_W \begin{bmatrix} I_x(x, y)^2 & I_x(x, y)I_y(x, y) \\ I_x(x, y)I_y(x, y) & I_y(x, y)^2 \end{bmatrix} \tag{2}$$

where I_x and I_y represents gradient of image w.r.t. x and y, W represents a window centered at point (x, y). The implementation is shown in **harris_corner_detection.m**.

The result of Harris corner detection algorithm is shown in Figure 1 and Figure 2.

The threshold is also varied to show its impact on corner detection. As one can see, the higher the threshold value is, the less interest points are detected.

The original image is rotated 45 and 90 degrees respectively to show Harris corner detection algorithm's rotation invariance property. As seen in Figure 4, most of the interest points are detected in both angles. Note that there are artifacts introduced due to rotation, which can safely be neglected in this case.

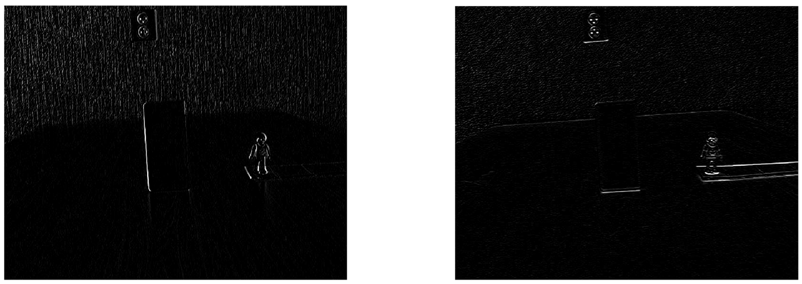
The reason why Harris corner detection is rotation invariant can be explained by examining the struct matrix H. Since H is symmetrical, it can visualize as an ellipse with axis lengths determined by the eigenvalues. Obviously, rotation does not change the shape of the ellipse, thus H's eigen values will remain constant. Therefore, Harris corner detection is rotation invariant.

Question 2

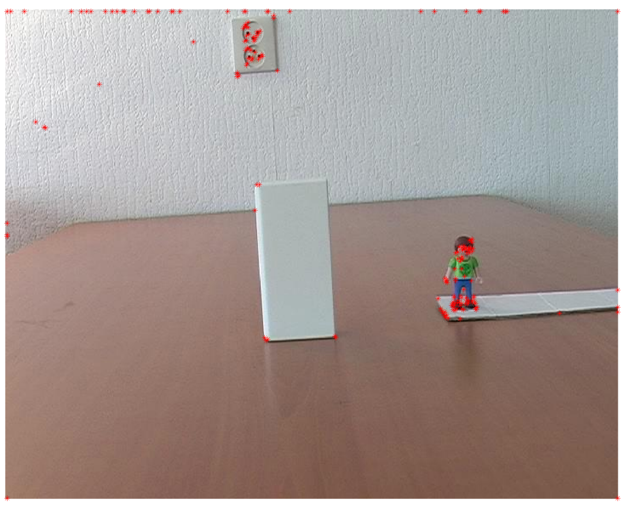
In Shi-Tomasi algorithm, the correctness is defined in [2] as:

$$H = \min\{\lambda_1, \lambda_2\} > \lambda_{min} \tag{3}$$

where λ_1 and λ_2 are 2 eigenvalues of Q(x,y) and λ_{min} is some predefined threshold.

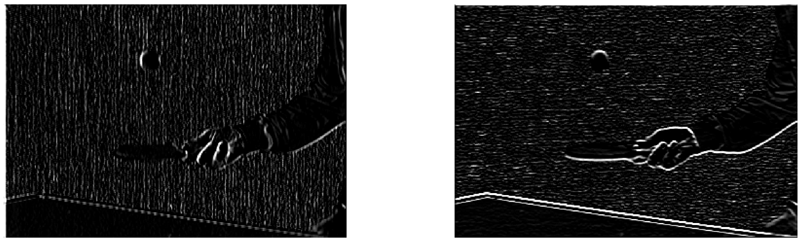


(a) I_x, I_y

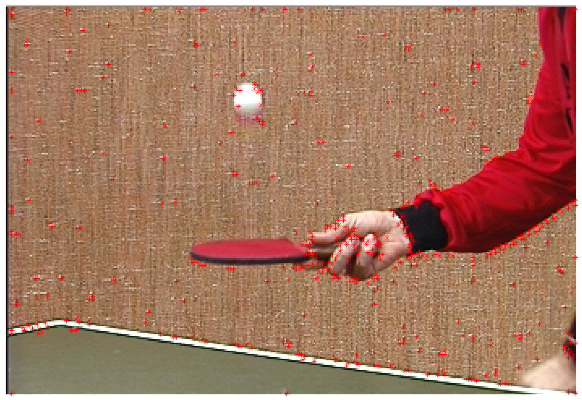


(b) interest points

Figure 1: Harris corner detection for person_toy/00000001.jpg with threshold = 0.01



(a) I_x, I_y



(b) interest points

Figure 2: Harris corner detection for pingpong/0000.jpeg with threshold = 0.01

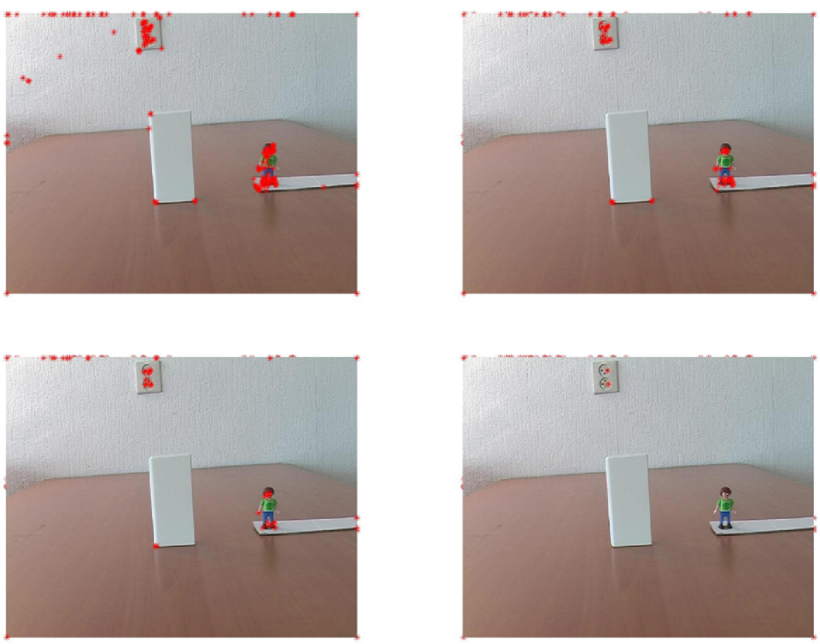


Figure 3: From left to right, top to down, the threshold values are $0.01, 0.05, 0.1, 1$ respectively

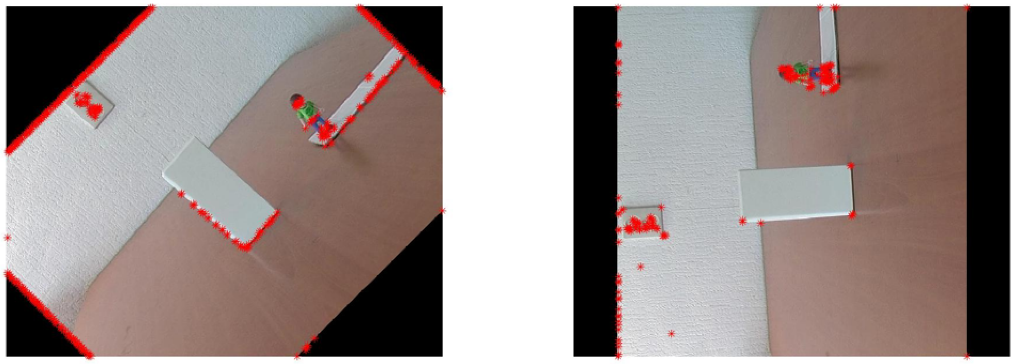


Figure 4: From left to right, the rotation angles are $45, 90$ respectively

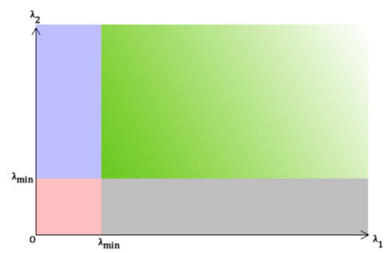


Figure 5: Relationship between eigenvalues and corners

Eigenvalue decomposition is not needed to be calculated at all since this requires taking square root so that computationally expensive. Similarly to Harris corner detection algorithm, one can approximate using matrices A and C, which contain pixel-wise eigenvalue information for the original image, and choose element-wise minimum of A and C as the result for $\min\{\lambda_1, \lambda_2\}$. As a reminder, $A = \sum_w I_x(x, y)^2$ and $C = \sum_w I_y(x, y)^2$.

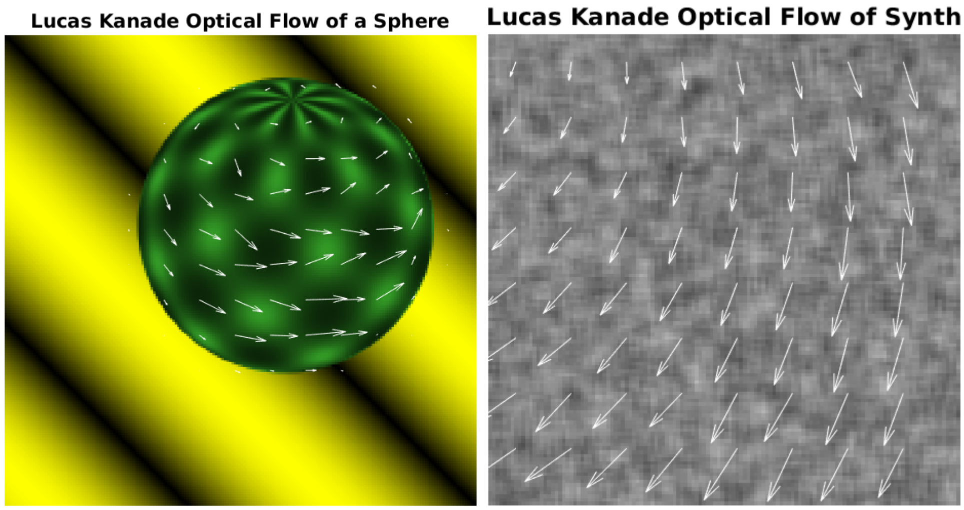
When both eigenvalues are near 0 (pink area in Figure 5), Shi and Tomasi will classify it as flat area of the image because there is no strong intensity change in original image. Furthermore, given one big eigenvalue and one near 0 eigenvalue (purple and gray areas in Figure 5), the algorithm treats it as an edge in the image since it indicates some unidirectional texture pattern. Finally, if both eigenvalues are big (green area in Figure 5), the algorithm predicts it as a corner. That is, both directions have strong intensity changes, which happens with high probability when an corner exists.

Optical Flow with Lucas-Kanade Algorithm

In order to determine the flow of an object, a Lucas-Kanade optical flow algorithm was implemented on a sphere object (sphere1 and sphere2) and on a synthetic object (synth1 and synth2). The algorithm attempts to find the solution to the over-constrained system of equations found from the equation 4 over every pixel in the patch where I_x is derivative of the intensity matrix in the x direction, I_y is derivative of the intensity matrix in the y direction, I_t is derivative of the intensity matrix in time (between frames, and V_x and V_y represent the velocity in the x and y direction respectively (the flow).

$$I_x V_x + I_y V_y = -I_t \tag{4}$$

The results can be seen in figure 6 and it shows that the ball is rotating anti-clockwise (left to right) as seen by the flow (arrows) and the synth object appears the rotating clockwise around an unseen center point.



(a) Ball image optical flow (b) Synthetic image optical flow

Figure 6: Sphere and synthetic image showing its flow as derived with the Lucas Kanade Optical Flow algorithm

In the Lucas Kanade method the image is divided into patches and the flow of all pixels in the patch are assumed to be the same, found by solving the over-constrained system of equations. The Horn-Schunck approach estimates the flow of every pixel in the image (global scale) by first assuming a smoothness in flow over the whole image, and then minimizing the global energy function constructed to represent the flow. Therefor where the Lucas Kanade finds a velocity for the whole patch, assuming they all move at approximately the same velocity, the Horn-Schunck will find one for each pixel. The Horn-Schunck is densities to noise when compared to the Lucas Kanade.

In flat regions, the Lucas Kanade algorithm would not be able to determine the flow for the patches in the flat region. This is because the gradient in those regions for the x and y directions become negligible in flat regions. For global energy based flow methods such as the Horn-Schunck, depending on the level of smoothing can be reliable estimators of the flow[1].

Feature Tracking

Question 1

In this section, Harris corner detector was used to extract feature points. Then, optical flow estimation was utilized to track these points. See videos under the folder "result_tracking".

Question 2

Even though one can detect features in separate frames, feature tracking is still needed to match and identify the same objects across different frames. For example, in human identification application, feature tracking can be used to track the same person in a video.

Further Considerations for Improving Feature Tracking

In order to achieve better results for feature tracking, one can smooth the image before applying Lucas-Kanade Algorithm. Since the algorithm assumes consistent optical flow in a local neighborhood, smoothing the image (e.g., averaging local pixels) makes the algorithm more robust to noise.

Another consideration is for the scaling factor when updating the positions of the interesting points between two frames. One may calculate I_t(time difference of 2 adjacent frames) at interesting points and use these as step sizes. That is,

$$x' = x + I_t V_x$$
$$y' = y + I_t V_y$$

where x are horizontal coordinates of interesting points, V_x is the horizontal velocity for optical flow, similarly for y and V_y.

Conclusion

In the first part of the experiment, Harris corner detection was explored. It was found that Harris corner detector can find corners using image gradients to construct matrix H, and use its eigen values to find interest points by calculating its cornerness values. An important hyper parameter threshold was experimented to show its impact on interest points. Higher threshold leads to less detected interest points. The algorithm's rotation invariance was also made evident by plotting different angles and examine the ellipse property of the struct matrix H. In the end, a modified version of corner detection by Shi and Tomasi was discussed. This new algorithm has a different definition or cornerness, and it does not require eigenvalue decomposition thus makes it more computationally convenient.

Next the Lucas Kanade method for determining the flow of an image through patches was used. An example of 2 images of a sphere and a synthetic object was demonstrated to show the optical flow. After this, other methods for finding the flow were examined, such as the Horn-Schunck algorithm, which uses a global energy function to find the flow of every pixel. It was found that for flat images, the Lucas Kanade method would not work well as the derivatives would be negligible, while depending on the smoothing result, the Horn-Schunck algorithm could give reliable estimators.

References

[1] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods". In: *International Journal of Computer Vision* 61.3 (2005), pp. 1–21. ISSN: 09205691. DOI: 10.1023/B:VISI.0000045324.43199.43.

[2] Jianbo Shi and Carlo Tomasi. "Good features to track". In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (1994), pp. 593–600.