Computer Vision 1
Assignment 2

Nichita Diaconu (11737980), Radu Sibechi(11808527)

1 Harris Corner Detector

The file main_harris.m is the main script for this exercise.

Question 1

Fig.1 On the left person_toy convolved with derivative on X-axis of gaussian. On the right person_toy convolved with derivative on Y-axis of gaussian.

Fig.2 On the left pingpong convolved with derivative on X-axis of gaussian. On the right pingpong convolved with derivative on Y-axis of gaussian.

The detected edges are more visible in the case of the table edges on the pingpong example. Nonetheless, the differences are very subtle to the human eye if we apply the derivative filter [-1/2 0 1/2]. Regardless of this, they are exact and useful further on.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

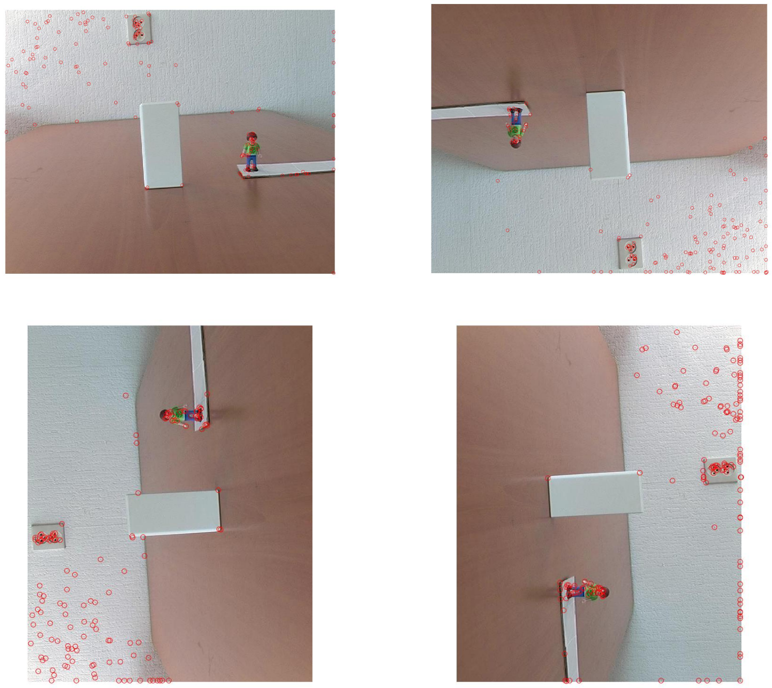

Fig.2 Harris corner detector on person_toy using Harris cornerness measure, using the covariance values IxIx; IyIy; IxIy at different angles.

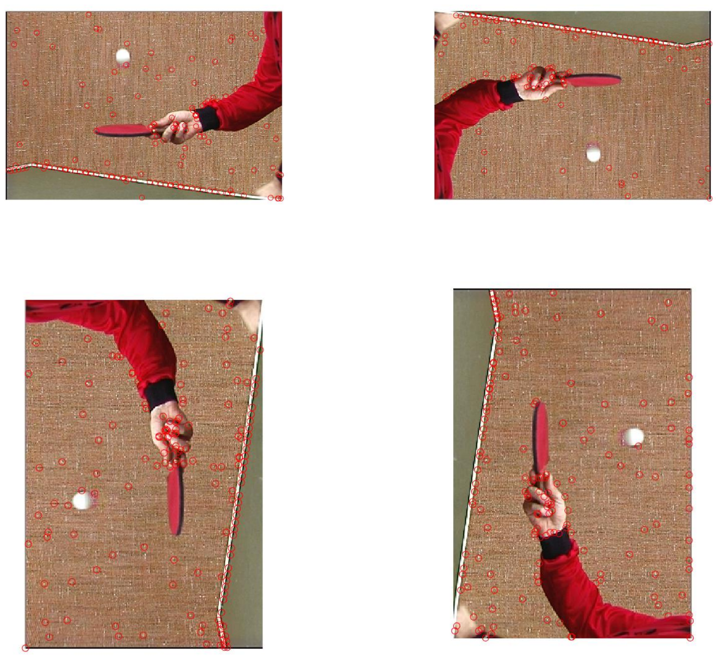

Fig.3 Harris corner detector on pingpong using Harris cornerness measure, using the covariance values IxIx; IyIy; IxIy at different angles.

We conclude based on the images in Fig.2 and Fig.3 that the Harris corner detection method is rotation invariant mostly. It does not preserve all the features throughout the rotations, bust most of the important feature (the toy eyes, the hand of the player) are still there, regardless of the rotation. The covariance matrix of the derivatives differs from angle to angle and therefore it makes some features have a higher cornerness than others in different rotation angles.

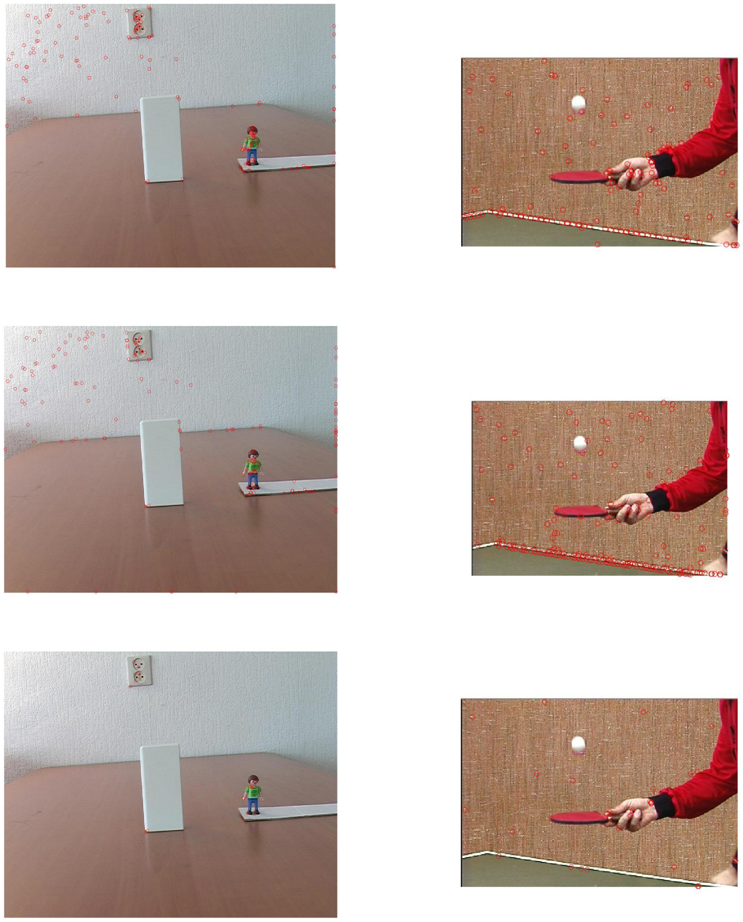

Fig.4 Harris corner detector on person_toy (left) and pingpong(right) using Harris cornerness measure with derivative covariences (row 1), Harris cornerness measure with eigenvalues (row 2), Shi Tomasi with eigenvalues (row 3). $\lambda_1 \lambda_2$ at different angles.

Question 2

1. Cornerness is defined as $min(\lambda_1, \lambda_2)$ of the eigenvalues of the derivative covariance matrix.
2. I think to answer this question, I would have to explain why we can use the derivative covariance values directly in the Harris measure instead of computing the eigenvalues. The eigenvalues would tell how how much the derivatives vary on the 2 axes on which they vary the most. The Harris measure can be explained as: the product of how much the derivatives (on X and Y axis) vary independently - the square of how much they vary together and there is some more stuff which might not be so relevant for this argument. Since we could always say that the derivatives on X and Y vary separately a lot more than they vary together with a minimum value of some threshold, we can use this measure. In the Shi-Tomashi, since we look at the minimum eigenvalue, we actually look at the minimum of the the two maximum variance components and that actually tells us how much the derivatives vary by themselves on the axis that they vary together the least (the eigenvectors are orthogonal since the covariance is symmetric). We cannot replace this with any of the 3 components in the Harris measure, IxIx, IyIy or IxIy.
3. (a) cornerness would be small, close 0 (no edge whatsoever)

(b) cornerness would be small, close 0 (probably an edge)
(c) cornerness would be high (probably a corner)

Experiments details, threshold We can also see in Fig.4 how Shi-Tomasi detect more accurate features. In our experiments we did not use a threshold, but rather the number of corners to be considered based on the cornerness value of each method. This variable is called top_N and was set to 250. This means that the algorithm will pick the top 250 corners based on their cornerness values. Nonetheless, we can see that, especially in the case of Shi-Tomasi, there are not 250 corners detected. This is because allowing for the next smaller cornernass value and its corresponding corners to be detected, there would be more than 250 corners. This means that the histogram of cornernerness values in probably very skewed in the case of Shi-Tomasi method.

2 Lucas Kanade

The file main_l_k.m is the main script for this exercise.

Question 1

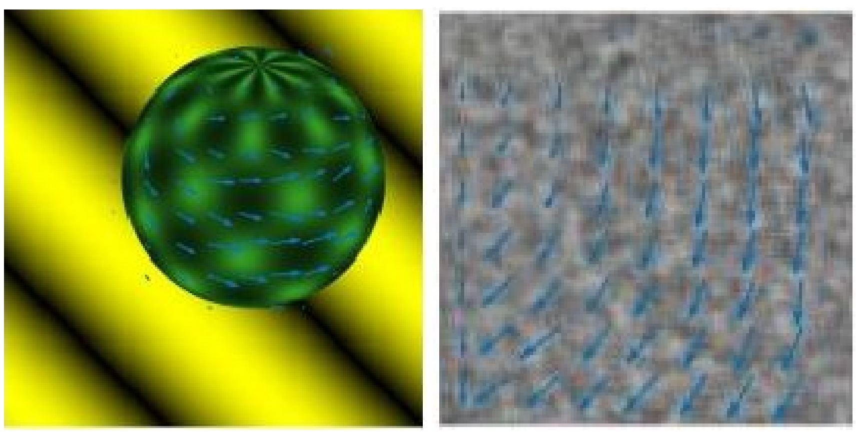

Fig.5 Flows of the sphere (left) and synth(right)

Question 2

1

Lucas and Kanade operates at a local scale, as it assumes that the flow is essentially constant in a local neighbourhood of the pixel under consideration and solves the basic optical flow equations for all the pixels in that neighbourhood. Horn-Schunck method operates at a global scale, by introducing a global constraint of smoothness in the flow over the whole image. As such, it tries to minimize the distortions in flow and prefers solutions which show more smoothness.

2

For Lucas and Kanade, the resulting linear system can be solved provided that its system matrix is invertible. This is not the case in flat regions, where the image gradient vanishes Therefore, LK approach, can not properly solve the aperture problem neither at edges nor flat regions.

Horn-Schunck solves this problem by imposing spatial smoothness to the flow field. Adjacent pixels should move together as much as possible.

3 Feature Tracking

The file main_track.m is the main of the tracking Exercise. If Matlab crashes while running this script, matlab should be opened with opengl rendering with the command (on Linux): matlab -softwareopengl

Question 1

The videos are names : person_toy.avi and pingpong.avi. They have been generated using the main_track.m script.

Question 2

Feature detection at every frame could suffer from not always finding the same features. Moreover for every frame we would have to match each feature with one in the new frame and that has $O(n^2)$ complexity. This complexity comes on top of the complexity of detecting the features in the image.