University of Girona

Visual Perception

Lab 2

Corner Detection

Author:
Di MENG

Supervisor:
Prof. Rafael Garcia

March 26, 2017

1 Introduction

The objective of this lab is to grab the technology of corner detection. Corner detection is an approach used to extract certain kinds of features and salient points of an image, which can be used in matching problem and many other applications. There are many methods put forward to detect corners, such as Moravec, Harris, Hessian and Laplacian. Harris corner detector is implemented in this lab assignment.

2 Algorithm analysis and implementation

A corner can be defined as the intersection of two edges and can also be defined as a point for which there are two dominant and different edge directions in a local neighbourhood of the point. The basic idea for detecting a corner is to observe the appearance change in neighborhood of a patch. It shows no change in flat region, no change along edge direction where there's a edge and it appears significant change when there's a corner. Harris corner detector gives a mathematical approach for determining which case holds.

This method is based on the use of the image gradient. Without loss of generality, the image is converted to gray scale which is 2 dimensional firstly. The derivatives along horizontal(x) and vertical(y) directions of the image are computed. Then the products of derivatives are computed prepared for the auto-correlation matrix. The squared derivatives are smoothed by a Gaussian filter. Thus, the auto-correlation matrix M can be defined as follows:

$$\mathbf{M} = \begin{pmatrix} \sum_W \langle I_x^2 \rangle & \sum_W \langle I_x I_y \rangle \\ \sum_W \langle I_x I_y \rangle & \sum_W \langle I_y^2 \rangle \end{pmatrix}$$

Where W is a 3×3 neighborhood around the point, Ix and Iy are the derivatives for each pixel respectively.

M matrix can be used to derive a measure of "cornerness" for every pixel.

The pixel is considered to belong to a corner when the rank of M is 2. A matrix E can also be computed from M, which contains for every point the value of the smaller eigenvalue of M. The salient points can be detected by selecting the higher values in matrix E.

There is a more efficient way to compute matrix R proposed by Harris and Stephens which is the approximation of E.

$$R = \det(M) - k \cdot tr(M)^2$$

Where k is a constant with value 0.04, det is the determinant and tr is the trace(sum of the diagonal elements). Instead of computing every pair of eigenvalues from matrix M for each pixel, this way saves more time but the accuracy is lower compared with matrix E.

3 Comparison of matrix E and R

3.1 Results

The results of selecting salient points(corners detection) based on matrix E and R are compared. The figure 1 to 4 show the matrix E and R of each image and the corners detected from matrix E and R respectively. The matrix E and R are represented in the form of images which can show the difference more visually.

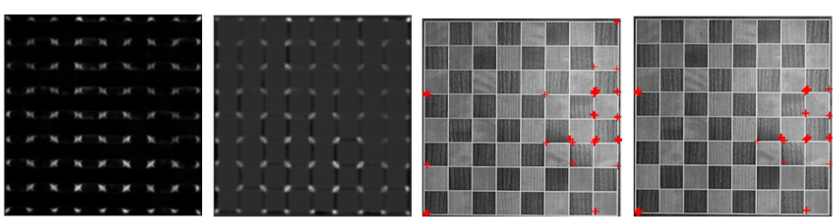

Figure 1: Chessboard03 - Matrix E, Matrix R, 81 most salient points from E, 81 most salient points from R

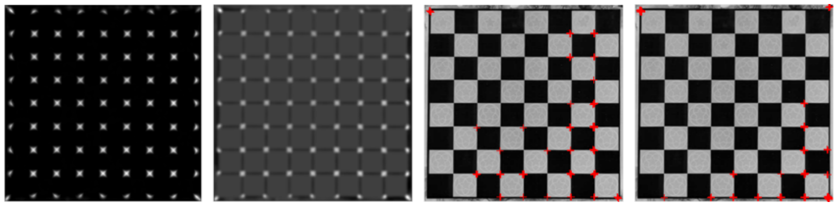

Figure 2: Chessboard04 - Matrix E, Matrix R, 81 most salient points from E, 81 most salient points from R

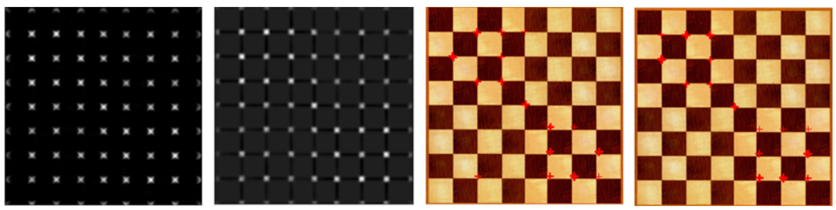

Figure 3: Chessboard05 - Matrix E, Matrix R, 81 most salient points from E, 81 most salient points from R

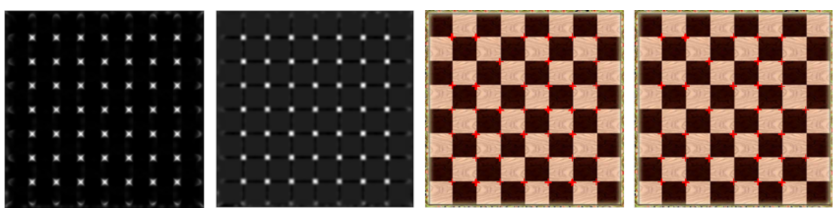

Figure 4: Chessboard06 - Matrix E, Matrix R, 81 most salient points from E, 81 most salient points from R

From the above four figures we can see that matrix E appears much contrast and more distinctive than matrix R. The pixels in the position of corners show higher values in matrix E than R compared with the background. And the 81 most salient points detected from matrix E appear more precise than R. Still both of the results are not as well as expected in this step.

3.2 Time consumption

The time consumption is compared according to computing the matrix E and R. The table below shows the time consumed when computing matrix

E and R of four images.

	Matrix E	Matrix R
Chessboard03:	0.384015 s	0.260196 s
Chessboard04:	0.374028 s	0.224866 s
Chessboard05:	0.271856 s	0.226587 s
Chessboard06:	0.304335 s	0.221217 s

From the table we can see that computing matrix R costs less time than computing matrix E. But the gap is not that much based on the advanced technology nowadays. Matrix E shows a significant advantage with higher accuracy compared with matrix R.

4 Improvement of the algorithm

4.1 Non-maximal suppression

As we can see from the result above, the corners detected are grouped together which are not as spread out as we expected. It is because the "corner" we thought which is a point is not located only on one pixel. More precisely observing, an area of pixels with high values from matrix E are supporting one corner in the image. What we want is the local maximum, so the idea of non-maximal suppression is brought up to maximize the distance between the salient points detected.

The way of implementing this function is that sorting the values in matrix E and going through the positions of these values in order. The neighbor pixels of the salient point in a certain window size are set to be zero. When the next salient point is being checked, it will be selected if its value is non-zero. Therefore, the salient points selected are all the local maximums and spread out in the image which satisfy the content of the image.

A function is implemented for this section. The input is the matrix E(or matrix R), the distance to maximize between points and the number of the salient points user wants to detect. The output is the salient points saved in a structure.

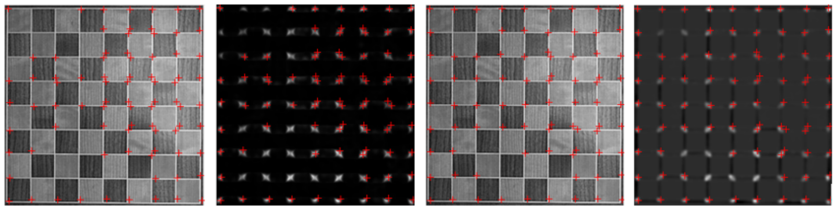

Figure 5: Chessboard03: Distance=11, From left to right - 81 most salient points form matrix E shown on original image, matrix E image, 81 most salient points form matrix R shown on original image, matrix R image

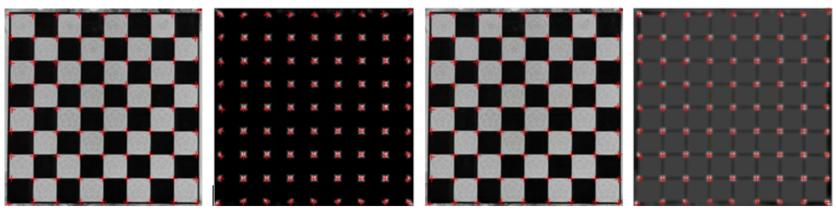

Figure 6: Chessboard04: Distance=11, From left to right - 81 most salient points form matrix E shown on original image, matrix E image, 81 most salient points form matrix R shown on original image, matrix R image

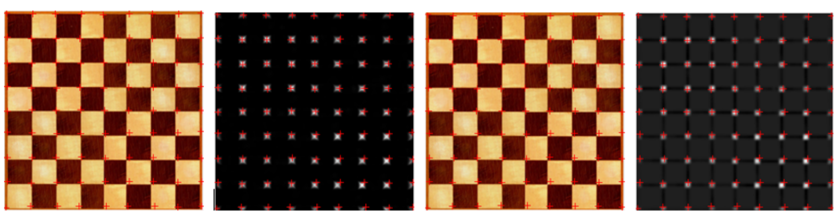

Figure 7: Chessboard05: Distance=11, From left to right - 81 most salient points form matrix E shown on original image, matrix E image, 81 most salient points form matrix R shown on original image, matrix R image

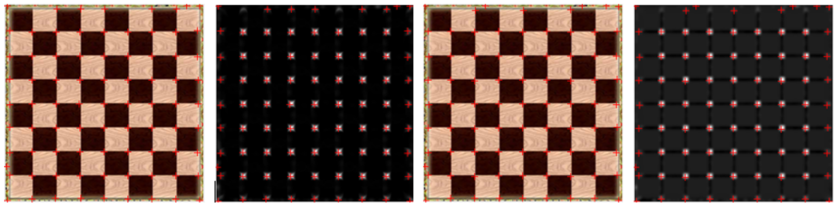

Figure 8: Chessboard06: Distance=11, From left to right - 81 most salient points form matrix E shown on original image, matrix E image, 81 most salient points form matrix R shown on original image, matrix R image

The four figures above show the corner detection using non-maximal suppression. We can see that the results are much better than before, the salient points are well located in the positions of corner by checking them in the image of matrix E and R. And the distance to maximize the salient points matters. For instance, image Chessboard03 could show better result when using distance 15.

4.2 Subpixel accuracy

When working with the images on a digital system, the smallest part is one pixel. The information between pixels can not be accessed. But several applications such as camera calibration, tracking and 3d reconstruction require higher accuracy than a camera can provide. So mathematical technique is developed for increasing the accuracy of detecting corners.

The basic idea of finding the subpixel accuracy is that based on the salient points detected and its neighbors, a surface function is built. The maximum which is the more precised subpixel can be found where the slope is zero in the quadratic equation. The more detailed information can be found in [1].

The way of implementing this method is that going through all the salient points detected, the parameters of the surface function are computed based on the 3x3 neighborhood. After obtaining the model(function) of the surface, the maximum can be computed, but coordinates are relative to the local center pixel. So the subpixels can obtained by converting the coordinates according to the image origin. The function built is in infinite range which could make mistakes, so the maximum computed is constrained in the range of 3x3 centered at the salient points.

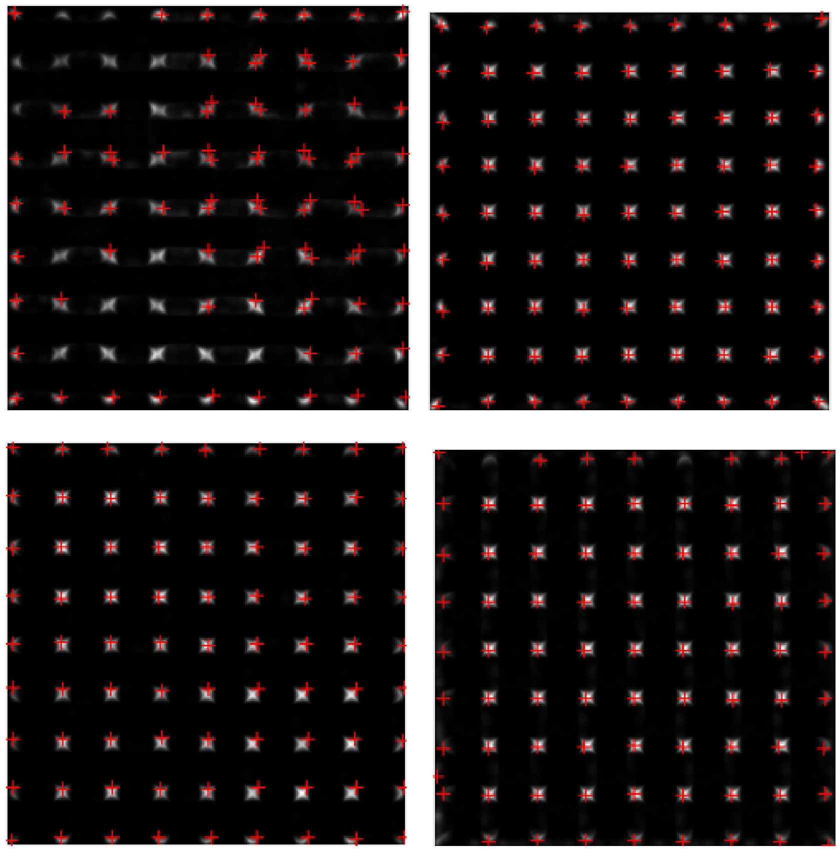

Figure 9: The salient points with subpixel accuracy shown in matrix E image. In order - Chessboard03, Chessboard04, Chessboard05, Chessboard06

From the figure above, we can hardly see the subpixel accuracy. The tabular shown below extracts 10 points' x coordinates for better observation of the subpixel accuracy.

1	2	3	4	5	6	7	8	9	10
66.1111	158.6739	159.6705	66	160.1467	128	190.1976	96.4164	159.4140	66
66	159	59	66	159	128	190	96	159	66

Figure 10: Subpixel accuracy of 10 points' x coordinates

5 Conclusion

Harris corner detector is analyzed and implemented in this lad assignment. The difference between matrix E and R is compared. For more precised results, non-maximal suppression and subpixel accuracy are implemented to improve the accuracy.

6 Reference

[1] Changming Sun. Fast optical flow using 3D shortest path techniques. Image and Vision computing, vol.20, no.13/14, pp.981/991, December 2002.