UNIVERSITAT DE GIRONA

VISUAL PERCEPTION

LABORATORY REPORT

Corner Detection

Author
Mohit VAISHNAV

Supervisor
Dr. Rafael GARCIA

March 14, 2018

1 Introduction

This lab was meant to introduce us to the techniques of detecting edges in any image. Implementing those algorithms gave us step by step overview of working of corner detection. Here we have implemented *Harris Corner Detection* algorithm which works on image gradient.

2 Implementation

Using the data set provided us in the class which was of *Chessboard* but differ slightly we have to check our algorithm over that. We presented the result via two ways, i.e. *Harris Corner Detection* as well as *Harris and Stephens* which used determinant and trace of matrix instead calculating eigenvalues.

Our next task was to detect 81 most salient points or features in both the types of results derived above using both these methods.

3 Observations

Implementation of the method of *Harris and Stephens* requires the computation of eigenvalues. Computing eigenvalues was a difficult task in early nineties where matrix inverse and multiplication were in itself very difficult. To overcome this bottleneck approximation method was used instead computing the exact values. In this determinant and trace of matrix was utilized in a way to give the best estimation of eigenvalues.

This could be done either in hard coding way or using the existing *MATLab* function *eig*. Although they both perform the same action but while using the predefined functions of *MATLab* run time of the program drastically increases whereas hard coding this algorithm decreases the time by the faction of 1/500. It was because of the internal computations before calculating determinant and trace of any matrices. Once we have the eigenvalues we can observe this in a gray-scale image form using *MATLab* function *mat2gray*.

Next we have to find the 81 most salient points. While projecting these wrt to original image, it was observed that they were clubbed in a large groups together. A corner is defined in terms of the gradient change which is not just a location but an area which varies according to the gradient at each place wherever it appears. This was the reason that though there are 81 corners in principle on a chess board but each of those corners contribute

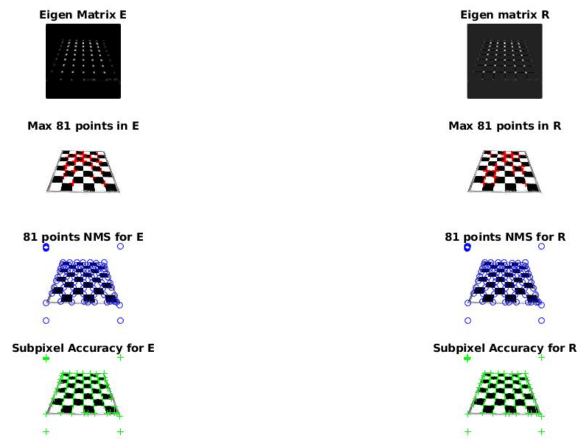

Figure 1: chessboard01

to many feature vectors hence the corners from all sides. To overcome this shortcoming we have used the technique of suppressing non maximal points around an area thereby trying to minimize the number of corners around a particular area and diversifying them. Our aim is to get exactly the same corners at each location which are present on a chess board.

4 Results

5 Difficulties

Basic structure of the code has already been provided in the lab sheet which was just needed to be redefined in such a way that could be used further for our purpose. The amount of difficulty was increasing as we were progressing in the implementation but the last two of them took much of the time. *Non Maximal Suppression* required help from the internet to get the commands and their usage after looking for a while whereas more rigorous thinking was required for the last part of the lab sheet. Here we were asked to compute the *Sub-Pixel Accuracy* where there was very little conceptual clarity which got clarified while in lab after having discussion with the supervisor.

Few implementations slips worth mentioning could be, accessing the pixels of an image. This looks trivial but it was completely different from

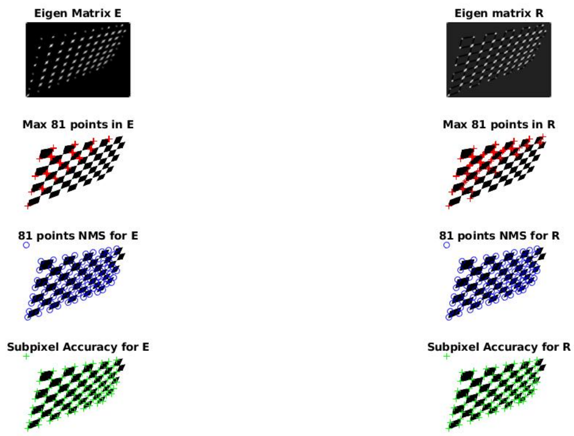

Figure 2: chessboard02

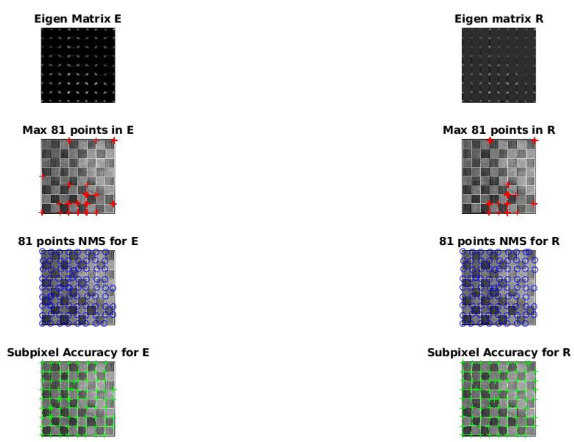

Figure 3: chessboard03

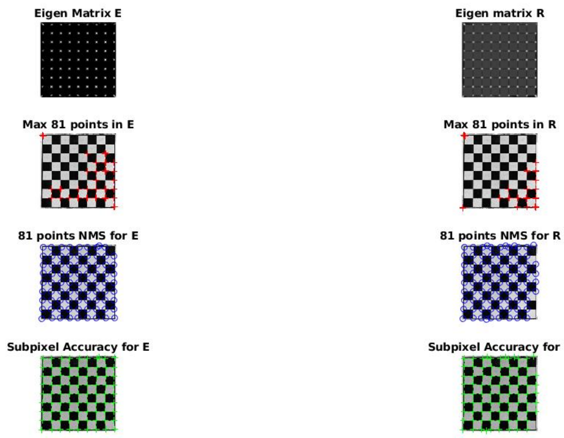

Figure 4: chessboard04

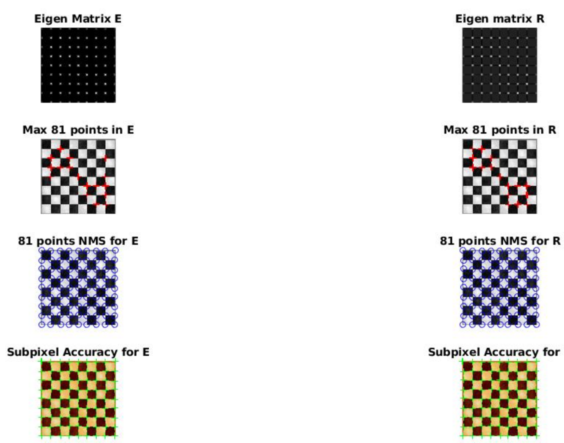

Figure 5: chessboard05

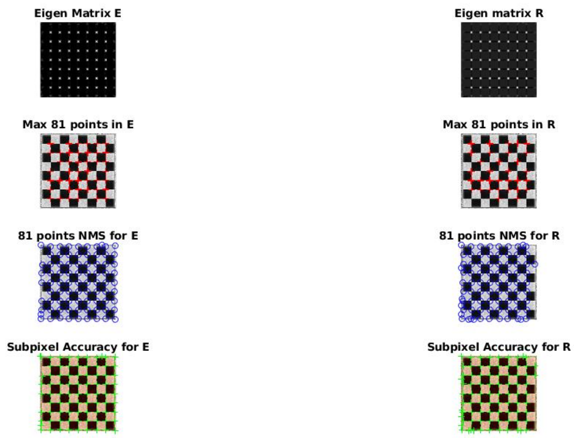

Figure 6: chessboard06

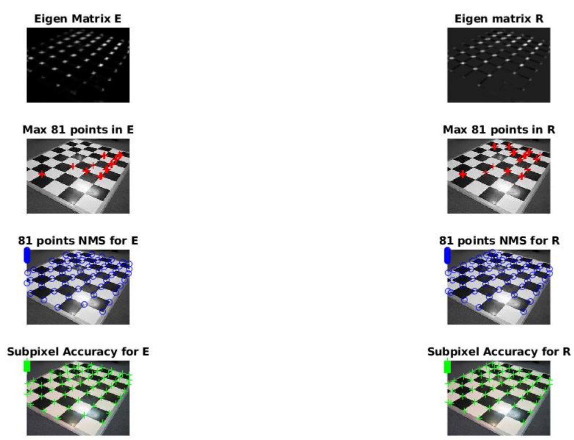

Figure 7: chessboard07

what our intuition says. Location (0,0) of the image is top left corner instead usually accepted bottom left corner. Hence to set the corner points in accordance with the image was little tricky and it took us some time.

6 Conclusion

Implementation of the *Harris Corner* detection technique was the first step towards understanding about the corners of an image. Understanding and extracting the corners of the image is one of the most basic feature for many of the state of the art algorithms like SIFT/SURF. So it is very important for us to have the knowledge from scratch. Many of its implementation could be image alignment, stitching and registration, motion detection, object recognition etc.