Computer Vision: Second Assignment (part 2)

Students Giacomo Lugano (5400573), Claudio Tomaiuolo (5630055)

Abstract

Corner detection algorithm is an algorithm which helps in identifying the corners in an image. Corners are mainly formed by the combination of two or more edges. These corners may or may not define the boundary of an image [1].
Harris Corner Detector is a corner detection operator that is commonly used in computer vision algorithms to extract corners and infer features of an image [2]. In this paper, we present the application of Harries Corners Detection to find all corners present and recognised within an image.

Introduction

A corner is a point whose local neighborhood stands in two dominant and different edge directions. In other words, a corner can be interpreted as the junction of two edges, where an edge is a sudden change in image brightness. Corners are the important features in the image, and they are generally termed as interest points which are invariant to translation, rotation, and illumination [1].

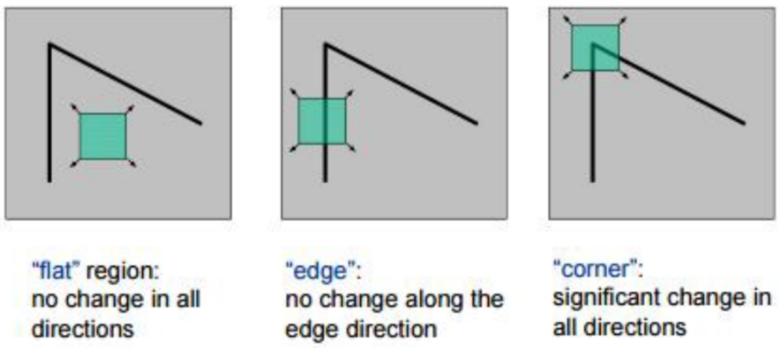

Figure 1: What is a corner.

In Figure 1, if we take the flat region then no gradient change is observed in any direction. Similarly, in the edge region, no gradient change is observed along the edge direction. While in corner region we observe a significant gradient change in all direction. Corners are considered good for patch matching and generally stable over the change of viewpoint [2].

Harris Corners Detection

A corners detector was first introduced by Chris Harris and Mike Stephens in their paper "A Combined Corner and Edge Detector" in 1988 (so now it is called the Harris Corner Detector), upon the improvement of Moravec's corner detector [3]. Compared to the previous one, Harris' corner detector takes the differential of the corner score into account with reference to direction directly, instead of using shifting patches for every 45-degree angles, and has been proved to be more accurate in distinguishing between edges and corners. Since then, it has been improved and adopted in many algorithms to preprocess images for subsequent applications [2].

HCD basically finds the difference in intensity for a displacement of (u,v) in all directions.

$$E(u,v) = \sum_{x,y} \underbrace{w(x,y)}_{\text{window function}} \, [\underbrace{I(x+u,y+v)}_{\text{shifted intensity}} - \underbrace{I(x,y)}_{\text{intensity}}]^2 \tag{1}$$

We have to maximize this function E(u,v) for corner detection. That means, we have to maximize the second term. Applying Taylor Expansion to the above equation and using some mathematical steps, we get the final equation as:

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \tag{2}$$

where

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix} \tag{3}$$

Here, I_x and I_y are image derivatives in x and y directions respectively.

Then comes the main part. After this, they created a score, basically an equation, which determines if a window can contain a corner or not.

$$R = \det(M) - k(\operatorname{trace}(M))^2 \tag{4}$$

where:

- det(M)=$\lambda_1\lambda_2$
- trace(M)=$\lambda_1+\lambda_2$
- λ_1 and λ_2 are the eigenvalues of M

So the magnitudes of these eigenvalues decide whether a region is a corner, an edge, or flat.

- When $|R|$ is small, which happens when λ_1 and λ_2 are small, the region is flat.
- When $R < 0$, which happens when $\lambda_1 \gg \lambda_2$ or vice versa, the region is edge.
- When R is large, which happens when λ_1 and λ_2 are large and $\lambda_1 \sim \lambda_2$, the region is a corner.

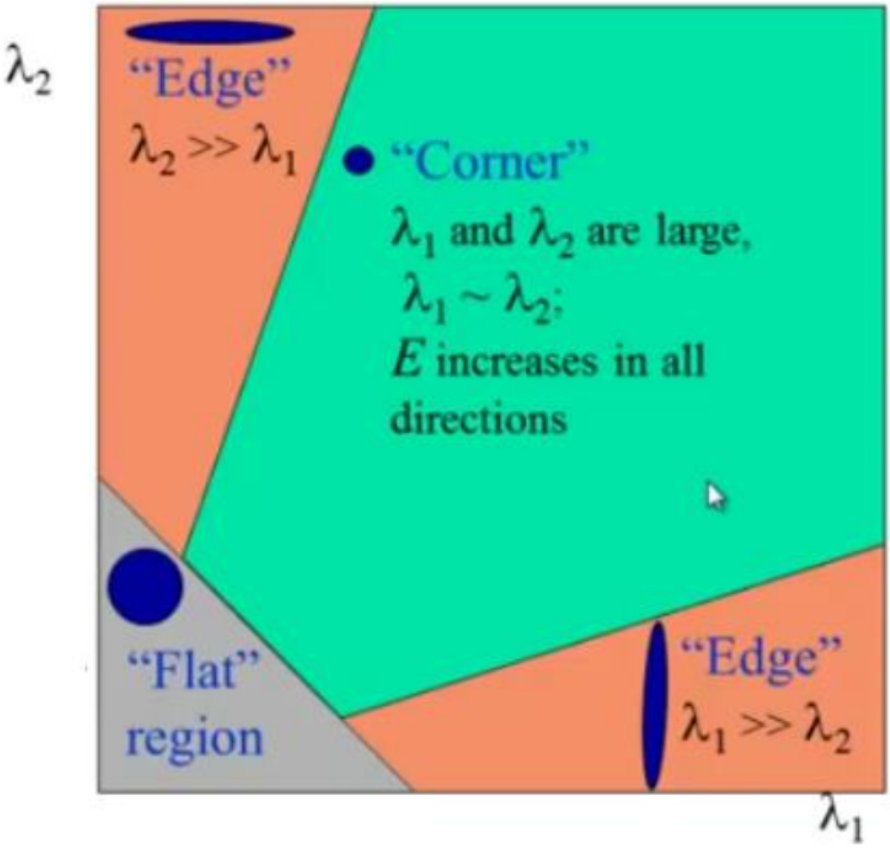

Figure 2: Representation of Corner, Edge and Flat regions

Results

Considering the following image:

Figure 3: Original image

we perform the partial derivatives, X (Figure 4) and Y (Figure 5), of the image, obtaining:

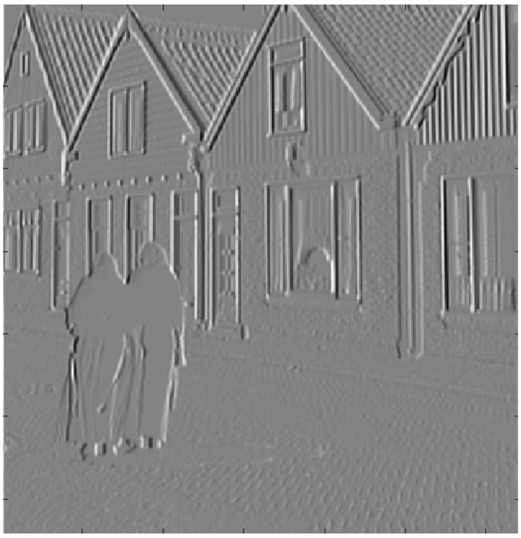

Figure 4: Derivative along X

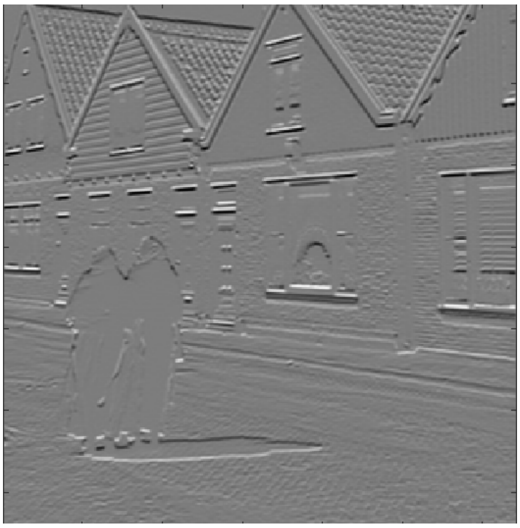

Figure 5: Derivative along Y

The Gaussian Filter used:

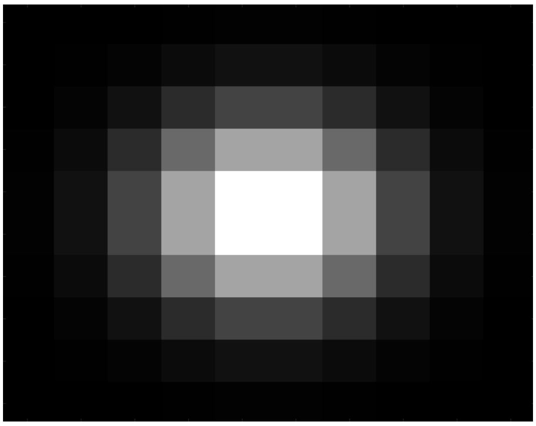

Figure 6: Gaussian Filter

Evaluating the M matrix coefficients and applying the corners response measure method, we obtained:

Figure 7: Corners response image

Applying the threshold for the corner regions 0.3*M, where M is the maximum value of R map, we obtained:

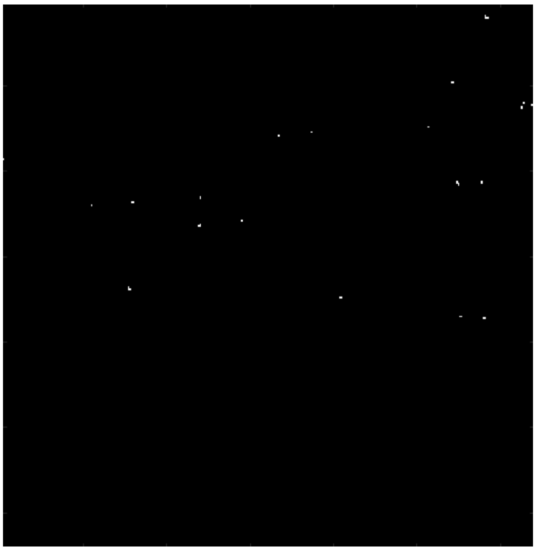

Figure 8: Corners response image with threshold applied

The Blob analysis consist of conversion of the corner region image in the logical image to calculate the centroids of the corners. Then, we use the function *regionprops()*, which returns the centroids in a structured array. The final result is:

Figure 9: Final image with corners detection

Conclusions

It can be deduced from Figure 9 that the algorithm of the Harris Corners Detection application works.

References

[1] P. Ram and S. Padmavathi (2016), *Analysis of Harris corner detection for color images*, 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), doi:

10.1109/SCOPES.2016.7955862.

[2] Wu Peng, Xu Hongling, Li Wenlin, Song Wenlong (2016), *Harris Scale Invariant Corner Detection Algorithm Based on the Significant Region*, International Journal of Signal Processing, Image Processing and Pattern Recognition, 9(3), 413-420.

[3] Chris Harris and Mike Stephens (1988), *A Combined Corner and Edge Detector*, Alvey Vision Conference. Vol. 15.