Computer Vision : Project 2

# Harris Corner Detection + SIFT Descriptor

Srijan Mallick
Alimpan Barik

Semester 2
M.Sc Big Data Analytics, RKMVERI

April 2021

# Contents

# 1 Introduction:

A **corner** can be defined as the intersection of two edges. It can also be defined as a point around which the image gradient has two or more dominant directions. One early attempt to detect these corners in an image was done by **Chris Harris** and **Mike Stephens** in their paper *"A Combined Corner and Edge Detector"* in 1988, so now the method is called **Harris Corner Detector**. It is rotation invariant, which means even if the image is rotated, we can find the same corners. However, it is not scale invariant. So, in 2004, **D.Lowe**, University of British Columbia, came up with a new algorithm, **Scale Invariant Feature Transform (SIFT)** in his paper, *"Distinctive Image Features from Scale-Invariant Keypoints"*, which extracts keypoints and computes its descriptors.

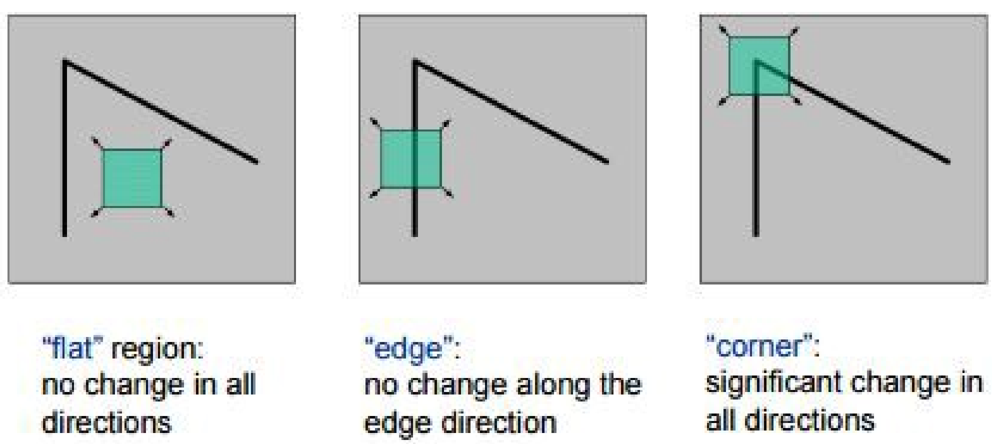



Figure 1.1: Properties of a corner point

# 2 Harris Corner Detection

## 2.1 Principle

Let $I$ be the given image. Then for a shift $[u,v]$, the change in the appearance of the window $w(x,y)$ is given by :

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u,y+v) - I(x,y)]^2$$

where $I(x,y)$ denotes the image intensity, $I(x+u,y+v)$ is the shifted intensity and $w(x,y)$ is the window function, which is usually taken as box kernel or Gaussian kernel.

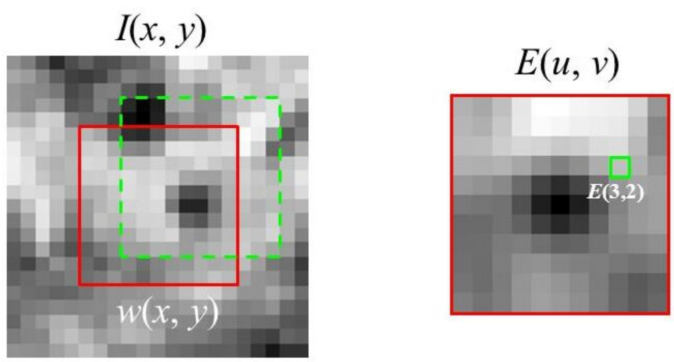



Figure 2.1

Our objective is to maximise the function $E(u,v)$ for corner detection. That means, we have to maximise the second term. Applying Taylor Expansion to the above equation followed by some algebraic manipulation, we get:

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

where

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Here $I_x$ and $I_y$ denote the image derivatives in $x$ and $y$ directions respectively.

After this, we compute the eigenvalues of $M$, which would help us decide whether a region is 'flat','edge' or a 'corner'. Let $\lambda_1$ and $\lambda_2$ be the eigenvalues of $M$.
So $det(M) = \lambda_1\lambda_2$ and $trace(M) = \lambda_1 + \lambda_2$

The cornerness score or the corner response is given by

$$R = det(M) - k(trace(M))^2$$

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

The above can be pictorially represented by :

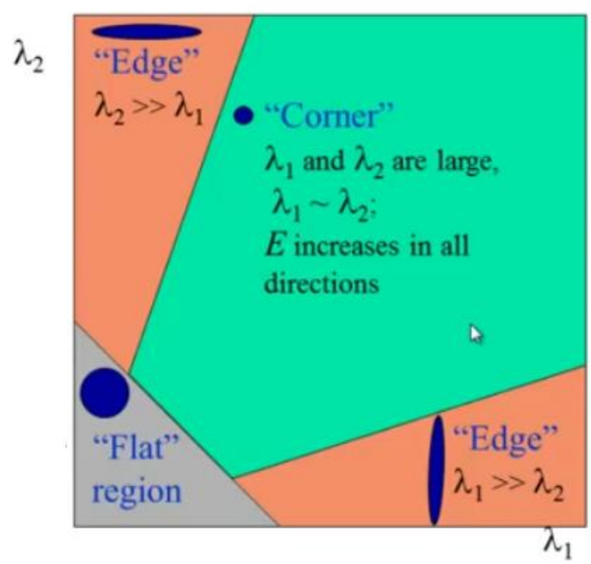



Figure 2.2

## 2.2 Algorithm

### 2.2.1 Smoothing the image

The purpose of this step is to reduce image noise by convolving the image with a Gaussian kernel.

### 2.2.2 Computing the Gradient of the image

We use the Sobel operator to compute the gradient of the image in the $x$ and $y$ direction.

### 2.2.3 Computing the Autocorrelation Matrix

Applying Sobel operator on the image gradients, we compute $I_x^2$, $I_y^2$ and $I_x I_y$, which is then used to form the autocorrelation matrix:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

### 2.2.4 Computing the Corner strength function

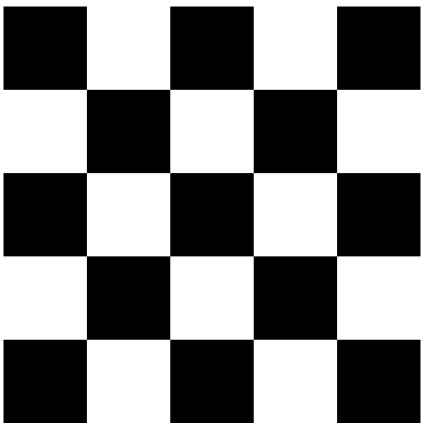
From the eigenvalues of the autocorrelation matrix M, we compute the corner response function. Among the many available corner response functions, we shall use the Harris corner response given by:

$$R = det(M) - k(trace(M))^2$$

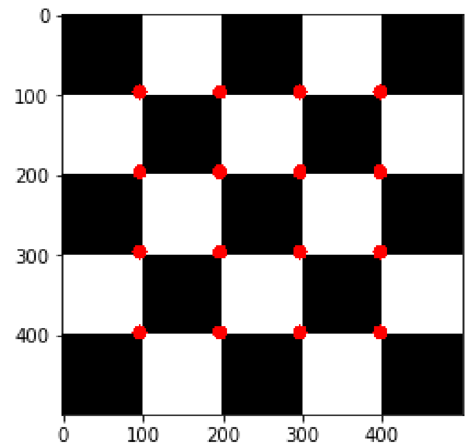
### 2.2.5 Non Maximum Suppression

The purpose of the non-maximum suppression step is to find the best interest point in each local neighborhood. This technique is typically used for refining edge-like structures, selecting points, or discriminating among simultaneous object detections.

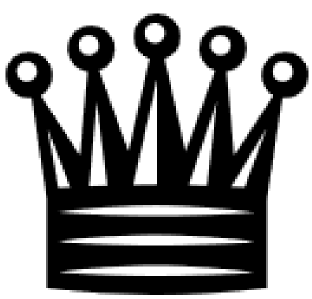
# 3 Some Examples of Image corners detected by Harris Method



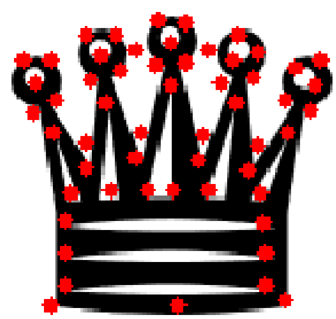
(a) Chess board




(b) Chess board with corners detected



(a) Queen (chess piece)



(b) Queen (chess piece) with corners detected

(a) Jet plane

(b) Jet plane with corners detected

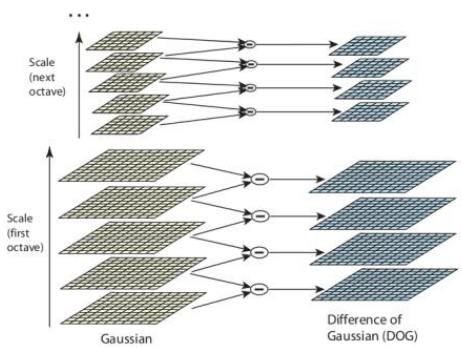
# 4 Feature Detection using SIFT

For any object in an image, interesting points on the object can be extracted to provide a "feature description" of the object. Scale Invariant Feature Trasform (SIFT) is a technique for detecting these (interest or) feature points in an image. The SIFT features are invariant to scale and rotation. There are mainly four steps involved in SIFT algorithm.

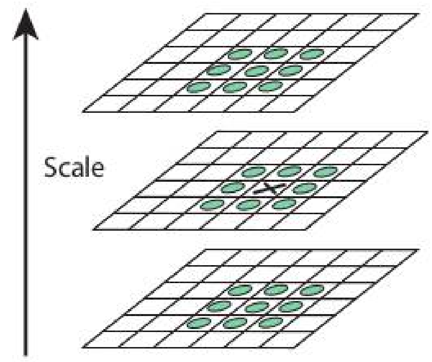
## 4.1 Algorithm

### 4.1.1 Scale-space Extrema Detection

In this step, Laplacian of Gaussian is to be found for the image with various $\sigma$ values. LoG acts as a blob detector which detects blobs in various sizes due to change in $\sigma$. But LoG is a little costly, so SIFT uses Difference of Gaussians which is an approximation of LoG. Difference of Gaussian is obtained as the difference of Gaussian blurring of an image with two different $\sigma$, let it be $\sigma$ and k$\sigma$. This process is done for different octaves of the image in Gaussian Pyramid. Then images are searched for local extrema over scale and space. Local extremas are potential keypoints.




(a) Difference of Gaussian




(b) Searching for local extrema over scale and space

### 4.1.2 Keypoint Localization

Once potential keypoints locations are found, they have to be refined to get more accurate results. DoG has higher response for edges, so edges also need to be removed. For this, a concept similar to Harris corner detector is used. Thus, low-contrast keypoints and edge keypoints are removed.

### 4.1.3 Orientation Assignment

Now an orientation is assigned to each keypoint to achieve invariance to image rotation. A neigbourhood is taken around the keypoint location depending on the scale, and the gradient magnitude and direction is calculated in that region. An orientation histogram with 36 bins covering 360 degrees is created.

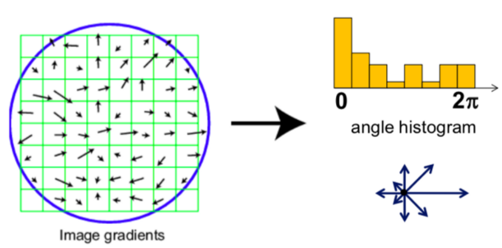



Figure 4.2: Orientation assignment

### 4.1.4 Keypoint Descriptor

Now keypoint descriptor is created. A 16x16 neighbourhood around the keypoint is taken. It is devided into 16 sub-blocks of 4x4 size. For each sub-block, 8 bin orientation histogram is created. So a total of 128 bin values are available. It is represented as a vector to form keypoint descriptor. In addition to this, several measures are taken to achieve robustness against illumination changes, rotation etc.

### 4.1.5 Keypoint Matching

Keypoints between two images are matched by identifying their nearest neighbours. But in some cases, the second closest-match may be very near to the first. It may happen due to noise or some other reasons. In that case, ratio of closest-distance to second-closest distance is taken.

## 4.2 Some Examples of SIFT applied images

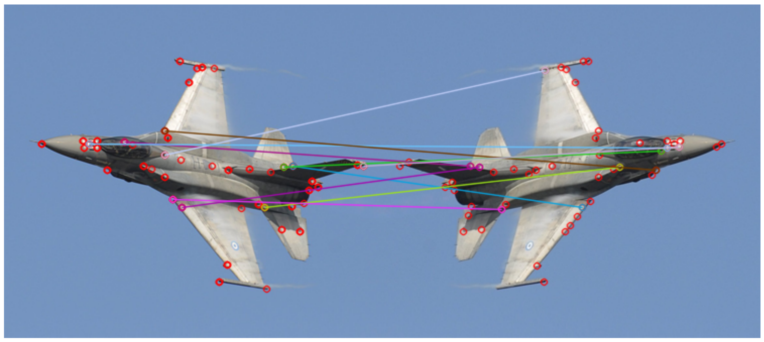


Figure 4.3

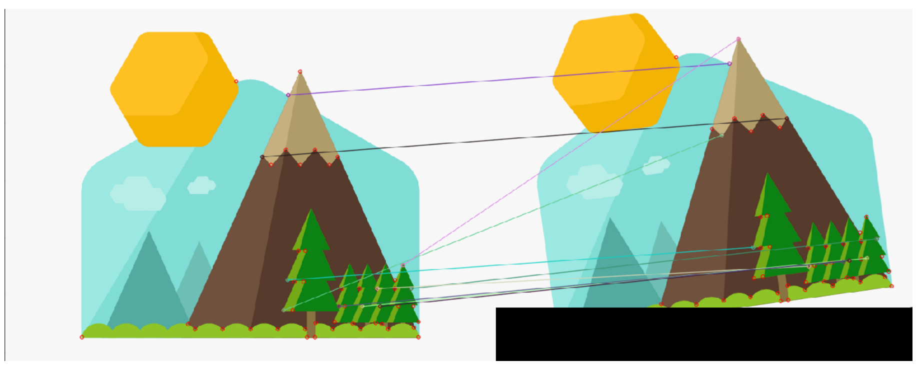


Figure 4.4

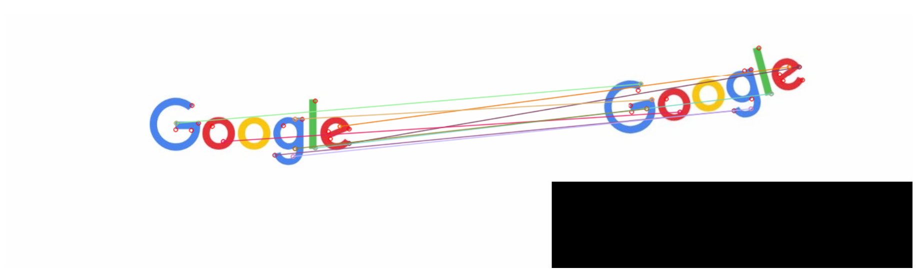



Figure 4.5

# 5 Conclusion

In this project, we saw how feature points are extracted from images. We saw the implementation of Harris corner detection algorithm, which is rotation invariant, and SIFT algorithm which is both scale and rotation invariant since it works with local descriptors. Thus, SIFT produces better and more efficient results while extracting feature points from images.

# 6 Acknowledgement

We are highly grateful to Tamal Maharaj, Dept. of Computer Science, for giving us the opportunity to work on this project. His guidance, supervision and support, throughout the project, has helped us immensely to complete it on time.

# 7 References

- Harris,C. & Stephens,M. *"A Combined Corner and Edge Detector" (1988)*
- Lowe,D. *"Distinctive Image Features from Scale-Invariant Keypoints" (2004)*, University of British Columbia
- Sanchez,J., Monzon,N. & Salgado,A. *"An Analysis and Implementation of the Harris Corner Detector"*