Computer Vision

Project 2

Harris Corner Detection And SIFT

Dristanta Das
Pratik Karmakar

Table of Contents

1 **Harris Corner Detection** 1

2 **Harris Corner Detector Implementation:** 1

3 **Harris Corner Detection algorithm:** 2

3.1 Step 1. Smoothing the Image 2
3.2 Step 2. Computing the Gradient of the Image 2
3.3 Step 3. Computing the Autocorrelation Matrix 3
3.4 Step 4. Computing the Corner Strength Function 3
3.5 Step 5. Non-Maximum Suppression 3
3.6 Step 6. Selecting Output Corners 3

4 **Examples of Images with Corners detected by Harris Method:-** 5

4.1 Example 1:Image of a House 5
4.2 Example 2:Chessboard 5
4.3 Example 3:Eifel Tower 6

5 **SIFT-Scale Invariant Feature Transform:-** 6

6 **SIFT Algorithm:-** 6

6.1 Extreme Detecting in Scale Space:- 6
6.2 Keypoints Locating:- 7
6.3 Keypoints Orientation Detection:- 7
6.4 Key-points's Descriptor Generation :- 9

7 **Example of Images on which SIFT is applied and the results:-** 10

7.1 Image of a House 10
7.2 Image of Eifel Tower 10
7.3 Image of Banksy Drawing 1 11
7.4 Image of Banksy Drawing 2 11
7.5 Banksy Drawing 3 12
7.6 Image in Atma Vikas 12
7.7 Image of the book "The Burning Forest by Nandini Sundar" 13

8 **Discussion:** 14

9 **References:** 14

1. Harris Corner Detection

The Harris corner detector is a standard technique for locating interest points on an image.Despite the appearance of many feature detectors in the last decade, it continues to be a reference technique, which is typically used for camera calibration, image matching, tracking or video stabilization .

2. Harris Corner Detector Implementation:

The idea behind the Harris method is to detect points based on the intensity variation in a local neighborhood: a small region around the feature should show a large intensity change when compared with windows shifted in any direction.This idea can be expressed through the autocorrelation function as follows: let the image be a scalar function I: $\Omega \rightarrow \mathbf{R}$ and ha small increment around any position in the domain, x $\in \Omega$ .Corners are defined as the points x that maximize the following functional for small shifts h,

$$E(h) = \sum w(x)(I(x+h) - I(x))^2 \tag{1}$$

i.e. the maximum variation in any direction. The function w(x) allows selecting the support region that is typically defined as a rectangular or Gaussian function. Taylor expansions can be used to linearize the expression I(x+h) as I(x+h)$\simeq I(x) + \nabla I(x)^\top h$, so that the right hand of first equation becomes,

$$E(h) \simeq \sum w(x)(\nabla I(x)h)^2 dx = \sum w(x)(h^\top \nabla I(x) \nabla I(x)^\top h) \tag{2}$$

This last expression depends on the gradient of the image through the autocorrelation matrix, orstructure tensor, which is given by

$$M = \sum w(x)(\nabla I(x)\nabla I(x)^\top) = \begin{bmatrix} \sum w(x) {I_x}^2 & \sum w(x) I_x I_y \\ \sum w(x) I_x I_y & \sum w(x) {I_y}^2 \end{bmatrix} \tag{3}$$

The second moment matrix M:

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

Can be written(without the weight):

Each product is a rank 1 2 × 2

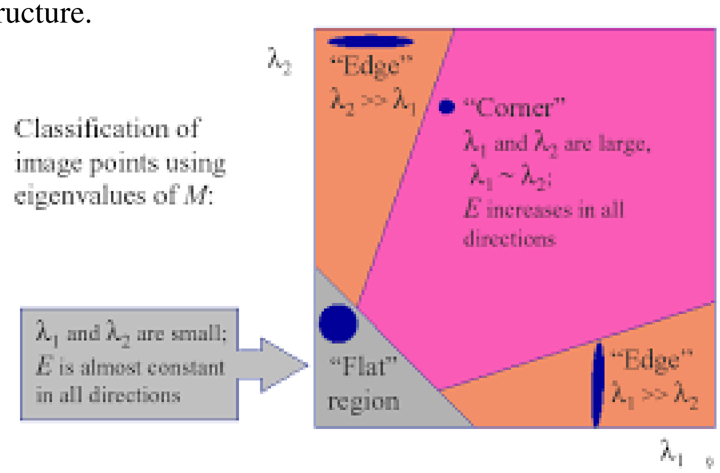
$$\mathrm{M} = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \left(\begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \quad I_y] \right) = \sum \nabla I (\nabla I)^T$$

The largest eigenvalue of M corresponds to the direction of largest intensity variation, while the second one corresponds to the intensity variation in its orthogonal direction. Analyzing their values, we may find three possible situations:

• Both eigenvalues are small, $\lambda_1 \approx \lambda_2 \approx 0$, then the region is likely to be a homogeneous region with intensity variations due to the presence of noise.

• One of the eigenvalues is much larger than the other one, $\lambda_1 >> \lambda_2 \approx 0$ then the region islikely to belong to an edge, with the largest eigenvalue corresponding to the edge orthogonal direction.

• Both eigenvalues are large, $\lambda_1 > \lambda_2 >> 0$ then the region is likely to contain large intensity variations in the two orthogonal directions, therefore corresponding to a corner-like structure.



3. Harris Corner Detection algorithm:

3.1 Step 1. Smoothing the Image

The purpose of this step is to reduce image noise and aliasing artifacts through the convolution with a Gaussian function. This step was not included in the original proposal, but it improves the performance of the method.

3.2 Step 2. Computing the Gradient of the Image

$$I_x = \frac{\partial I}{\partial x} \quad , \quad I_y = \frac{\partial I}{\partial y} \tag{4}$$

$$I_x I_y = \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \tag{5}$$

3.3 Step 3. Computing the Autocorrelation Matrix

The products of the derivatives are calculated at each position and the coefficients of the matrix are convolved with a Gaussian function.

$$M = \sum w(x)(\nabla I(x)\nabla I(x)^{\top}) = \begin{bmatrix} \sum w(x){I_x}^2 & \sum w(x)I_xI_y \\ \sum w(x)I_xI_y & \sum w(x){I_y}^2 \end{bmatrix} \tag{6}$$

3.4 Step 4. Computing the Corner Strength Function

The autocorrelation matrix is symmetric and positive semidefinite, yielding two real non-negative eigenvalues. Analyzing these eigenvalues, we may define corner response functions that are invariant to in-plane rotations. We implement the following standard functions

Harris and Stephens	$R_H = \lambda_1\lambda_2 - k(\lambda_1+\lambda_2)^2$
Shi and Tomasi	$R_{ST} = \lambda_{min} \quad if \quad \lambda_{min} > \tau, \quad 0 \quad Otherwise$
Harmonic Mean	$R_{HM} = \frac{2\lambda_1\lambda_2}{\lambda_1+\lambda_2}$

3.5 Step 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.The maxima of the corner strength function contain the interest points. However, many of these points will be located close to each other, thus, it is necessary to select the best candidates.

Algorithm 1 Non-Max Suppression

procedure NMS(B,c)
    $B_{nms} \leftarrow \emptyset$ Initialize empty set
    for $b_i \in B$ **do** => Iterate over all the boxes
        $discard \leftarrow \text{False}$ Take boolean variable and set it as false. This variable indicates whether b(i) should be kept or discarded
        for $b_j \in B$ **do** Start another loop to compare with b(i)
            if $\text{same}(b_i, b_j) > \boldsymbol{\lambda}_{\mathbf{nms}}$ **then** If both boxes having same IOU
                if $\text{score}(c, b_j) > \text{score}(c, b_i)$ **then**
                    $discard \leftarrow \text{True}$ Compare the scores. If score of b(i) is less than that of b(j), b(i) should be discarded, so set the flag to True.
        if not $discard$ **then**
            $B_{nms} \leftarrow B_{nms} \cup b_i$ Once b(i) is compared with all other boxes and still the discarded flag is False, then b(i) should be considered. So add it to the final list.
    return B_{nms} Do the same procedure for remaining boxes and return the final list

3.6 Step 6. Selecting Output Corners

The ways we can select corners are,

• The simplest one is to select all the corners detected. In this case, the sorting is given by the non-maximum suppression algorithm, i.e. sorted by rows and then by columns.

• In the second strategy, all the corners are sorted by their corner strength values in descending order. This is interesting for applications that need to process the more discriminant features first.

•Another alternative is to select a subset of the corners detected. The user specifies a number of corners to be found and the application returns the set with the highest discriminant values.The corners are also sorted in descending order. It is possible that the number specified by the user is bigger than the corners detected.•Finally, we may also select a set of corners equally distributed on the image, which is interesting for several applications such as camera calibration, panorama stitching [3] or video stabilization . In that case, the user specifies a number of cells and the total number of points to be detected. The algorithm tries to find the same amount of points in each cell. It is possible that no distinctive points are detected in some cells, so, in general, the number of features will be smaller than the target number of points specified by the user.

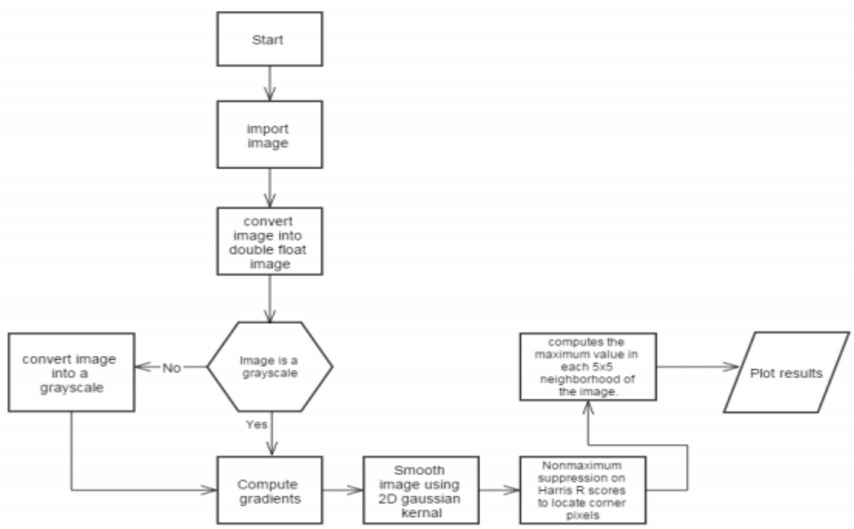
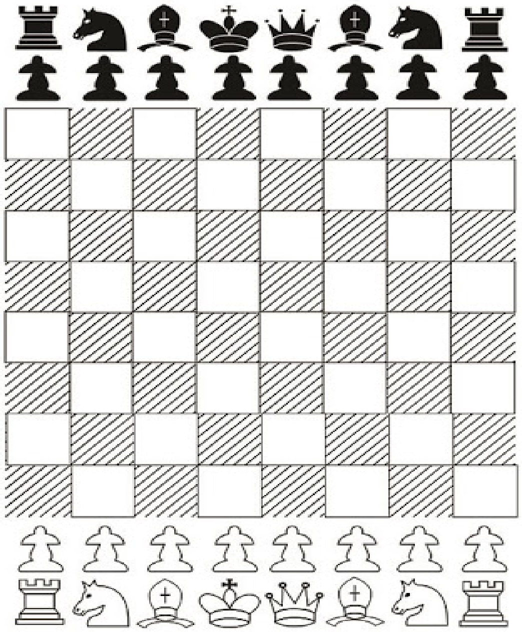
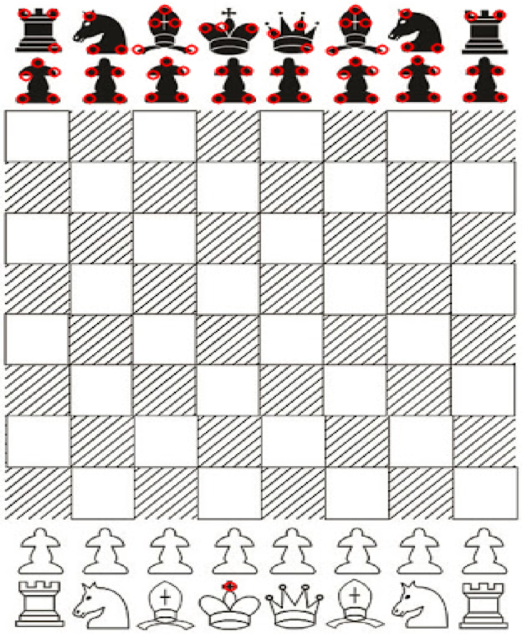


Figure 2: Harris Corner Flowchart

4. Examples of Images with Corners detected by Harris Method:-

4.1 Example 1:Image of a House


4.2 Example 2:Chessboard

4.3 Example 3:Eifel Tower

5. SIFT-Scale Invariant Feature Transform:-

The Scale-Invariant Feature Transform (SIFT) is an algorithm used to detect and describe local features in digital images. It locates certain key points and then furnishes them with quantitative information (so-called descriptors) which can for example be used for object recognition. The descriptors are supposed to be invariant against various transformations which might make images look different although they represent the same object(s).

6. SIFT Algorithm:-

6.1 Extreme Detecting in Scale Space:-

Scale-space theory intends to simulate the multi-scale features of image data, and get different spatial scales by Gaussian blur. Lindeberg etc. has proved that the Gaussian convolution kernel is the only transform nuclear to achieve scale transformation, and is the only linear kernel. The scale-space of two-dimensional image is defined as follows:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \tag{7}$$

In which, $G(x,y,\sigma)$ is a scale variable Gaussian function:

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{8}$$

The σ decides image's smoothness. Large scale corresponds to image's profile feature, and small scale corresponds to minutiae. I(x,y) represents an image.

Using Gaussian pyramid to represent scale space. In order to reflect its scale spatial continuity, Gaussian pyramid adds Gaussian filter on the basis of simple downsampling. Maximum and minimum values of the scale normalized Laplacian of Gaussian function can produce a stable image characteristics, and the Difference of Gaussian function is similar with the scale normalized Laplacian of Gaussian function. Lowe uses Gaussian Differential operator to detect extreme value instead of Gaussian Laplacian operator.

$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma) \end{aligned} \tag{9}$$

6.2 Keypoints Locating:-

The extreme points detected in the Difference of Gaussian pyramid is the extreme points in discrete space, not the real extreme points. In order to accurate locating keypoints' locations and scales, it necessary to curve fitting the Difference of Gaussian scale-space function (DOG). Utilizing the DOG function Taylor expansion in scalespace:

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X \tag{10}$$

In which, $X = (x,y,\sigma)^T$ Take the derivative of D(X), and set it equal to zero. Then extreme points' offset X^T is available. X^T represents an offset relative to the center of the interpolation. When its value is greater than 0.5 on any dimension, indicating that the interpolation center is shifted to adjacent point. In this condition, the location of keypoint must be changed and it needs to interpolate repeatedly in the new location until convergence.

6.3 Keypoints Orientation Detection:-

Every selected feature point needs to assign a reference orientation, so that it can invariant to rotation. In the scalespace where keypoint belongs, computing pixels' gradient and orientation distribution within 3 neighborhood region. The values of gradient and orientation are calculated as follows:

$$A = L(x+1,y) - L(x-1,y) \tag{11}$$

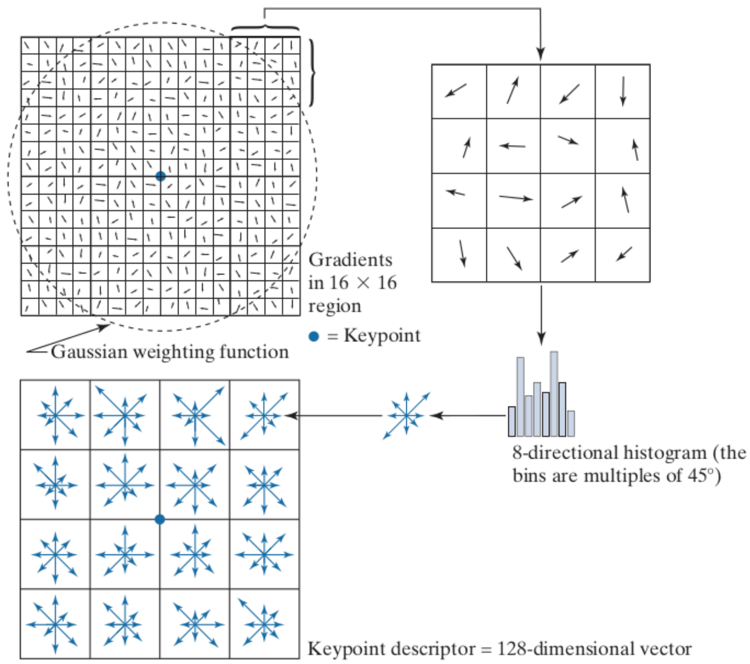
$$B = L(x,y+1) - L(x,y-1) \tag{12}$$

$$m(x,y) = \sqrt{A^2 + B^2} \tag{13}$$

$$\theta(x,y) = tan^{-1}(\frac{B}{A}) \tag{14}$$

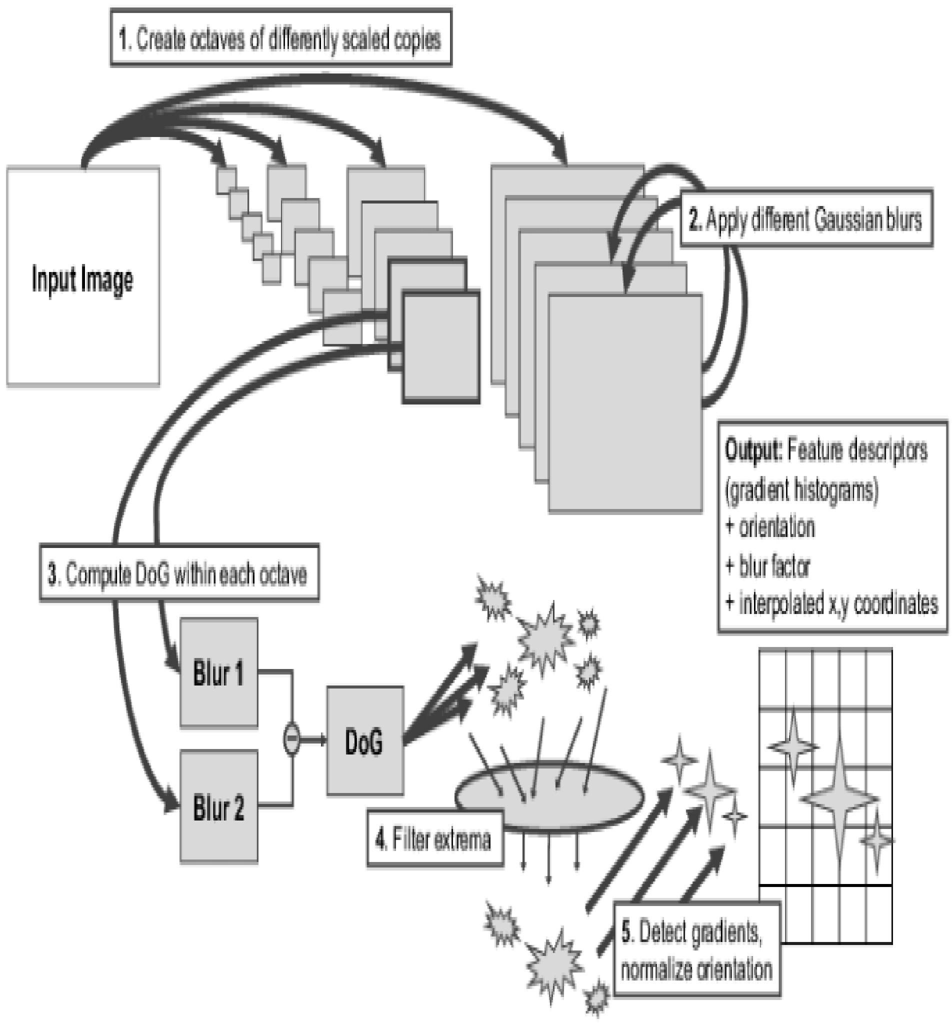
In which, L is the scale-space where keypoint belongs.

After finishing calculating the gradient of keypoint, using histogram to count pixels' gradient and orientation in neighborhood region. Histogram's peak represents the main orientation of keypoint, and take other bins which greater than 80 % of the peak as the auxiliary orientation of keypoint.



6.4 Key-points's Descriptor Generation :-

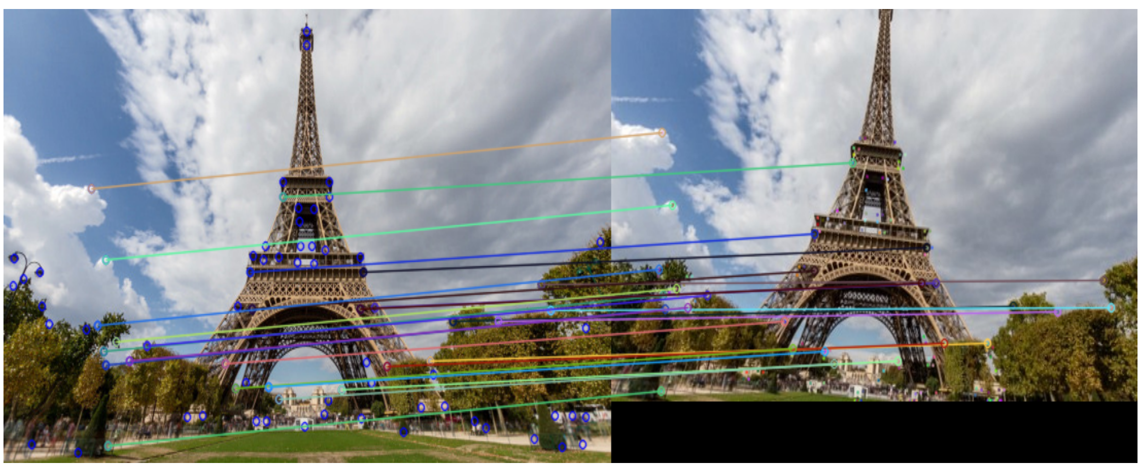
Taking the key-point as center, dividing the square neighborhood region into 4×4 sub-regions, and rotating the axis to the orientation of key-point. Calculating every pixels' gradient and orientation in every sub-region, and assigning the gradient values to eight orientation by weight, to form seed points with eight data. In order to reduce the influence of light, the 128-dimensions gradient data needs to be normalized. After these four steps, it finally can get 128-dimensions feature descriptor of SIFT.



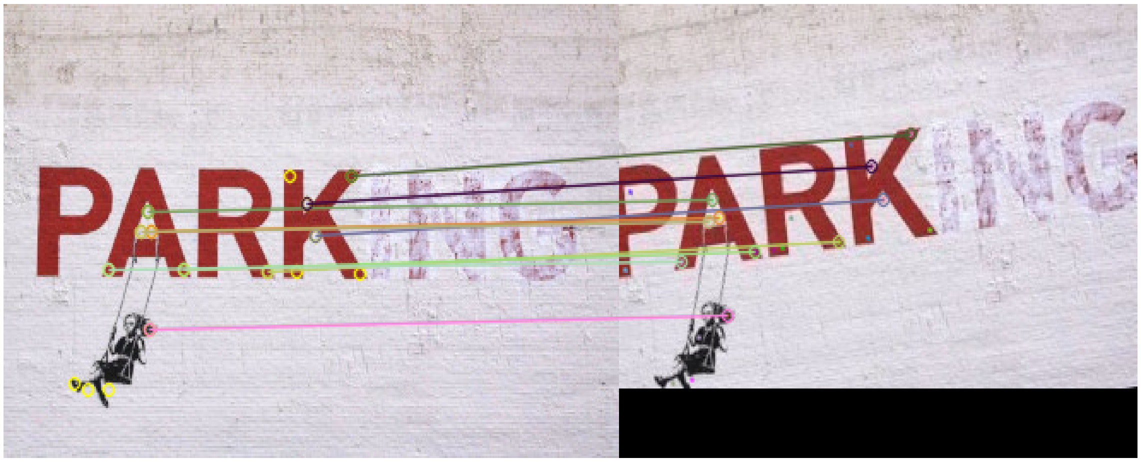
7. Example of Images on which SIFT is applied and the results:-

7.1 Image of a House

7.2 Image of Eifel Tower

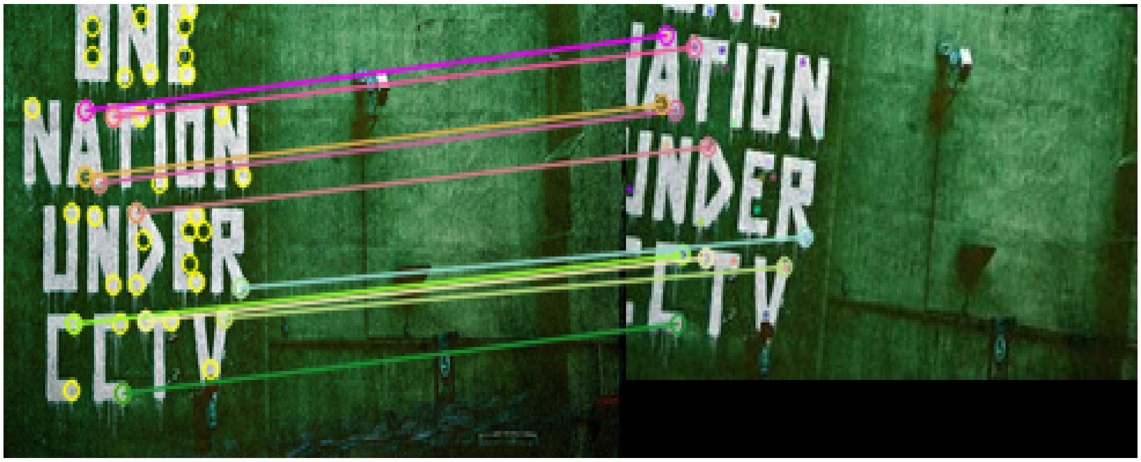


7.3 Image of Banksy Drawing 1

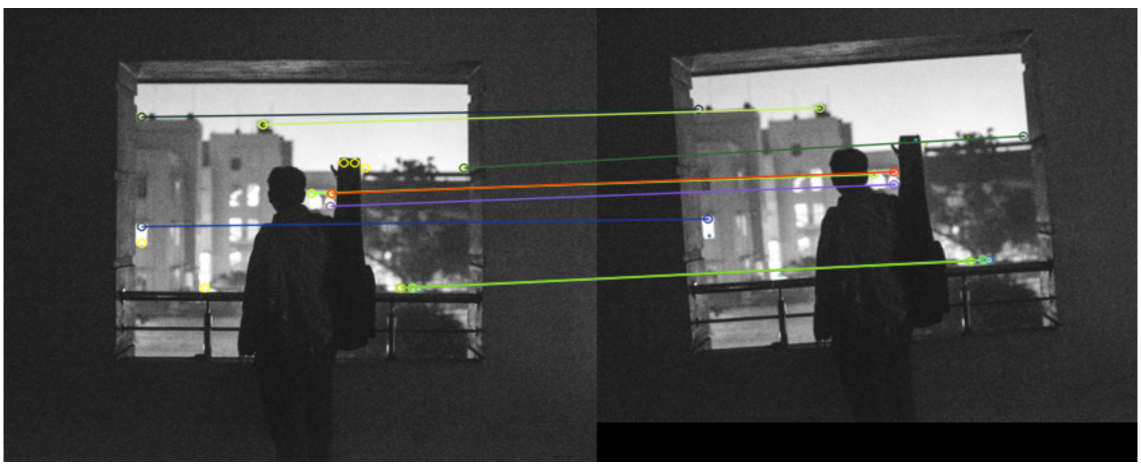


7.4 Image of Banksy Drawing 2

7.5 Banksy Drawing 3



7.6 Image in Atma Vikas



7.7 Image of the book ”The Burning Forest by Nandini Sundar”

8. Discussion:

As corners serve as very good feature points in an image, it is widely used for pattern recognition. Moreover, detection of corners allows us to identify places, figures etc. Harris corner detection method works as a great solution in this.

Now, SIFT comes in as a much finer application of feature detection. This stands out as an important method as it is scale invariant and rotation invariant as the available data at hand is not always the way we want to be. So, to detect features, we have used Harris Corner detection method here as an elementary step.

9. References:

[1] David G. Lowe . Distinctive Image Features from Scale-Invariant Keypoints, January 2005.

[2] Richard Szeliski . Computer Vision: Algorithms and Applications

[3] Rafael C. Gonzalez, Richard E. Woods . Digital Image Processing