S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

Practical No. 7

Aim: Implement programs to Analyse following feature detection techniques using OpenCv library.

a) Hough Transform b) Scale Invariant Scale detection (SIFT).

Name of Student: Shrutika Pradeep Bagdi

Roll No.: CS22130

Semester/Year: VII/IV

Academic Session: 2025-2026

Date of Performance:

Date of Submission:

AIM: Implement programs to Analyse following feature detection techniques using OpenCv library.
a) Hough Transform b) Scale Invariant Scale detection (SIFT)

OBJECTIVE/EXPECTED LEARNING OUTCOME:

- To be able to identify and locate lines, circles, and other geometric shapes in images using Hough transform.
- To be able to match keypoints between images and perform object recognition tasks using SIFT.
- To be able to compare and contrast the performance of Hough transform and SIFT feature detection techniques.

THEORY:

The Hough transform is a technique which can be used to isolate features of a particular shape within an image. Because it requires that the desired features be specified in some parametric form, the classical Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, etc. A generalized Hough transform can be employed in applications where a simple analytic description of a feature(s) is not possible. Due to the computational complexity of the generalized Hough algorithm, we restrict the main focus of this discussion to the classical Hough transform. Despite its domain restrictions, the classical Hough transform (hereafter referred to without the classical prefix) retains many applications, as most manufactured parts (and many anatomical parts investigated in medical imagery) contain feature boundaries which can be described by regular curves. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.

Working:
The Hough technique is particularly useful for computing a global description of a feature(s) (where the number of solution classes need not be known a priori), given (possibly noisy) local measurements. The motivating idea behind the Hough technique for line detection is that each input measurement (e.g. coordinate point) indicates its contribution to a globally consistent solution (e.g. the physical line which gave rise to that image point).
As a simple example, consider the common problem of fitting a set of line segments to a set of discrete image points (e.g., pixel locations output from an edge detector). Figure 1 shows some possible solutions to this problem. Here the lack of a priori knowledge about the number of desired line segments (and the ambiguity about what constitutes a line segment) render this problem under-constrained.

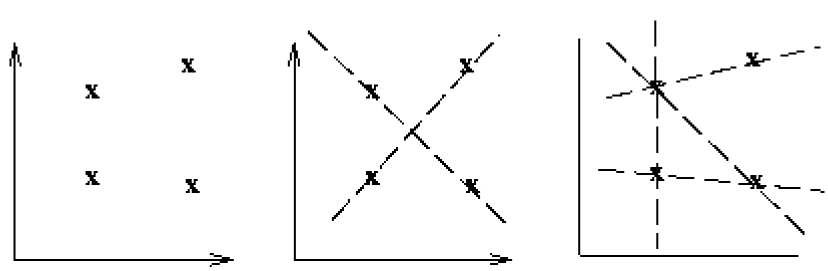


Figure 1 a) Coordinate points. b) and c) Possible straight-line fittings.

we can analytically describe a line segment in a number of forms. However, a convenient equation for describing a set of lines uses *parametric* or *normal* notion:

$$x \cos\theta + y \sin\theta = r$$

where r is the length of a normal from the origin to this line and Θ is the orientation of r with respect to the X-axis. (See Figure 2.) For any point on this line, r and Θ are constant.

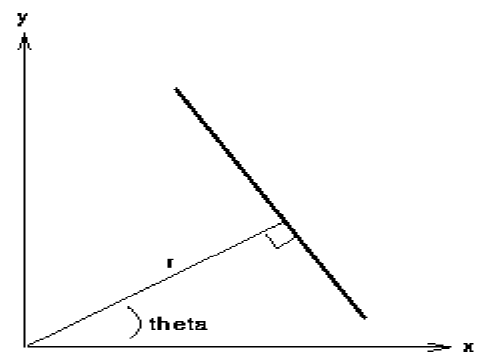


Figure 2 Parametric description of a straight line.

Implementation in OpenCV:

ALGORITHM:

PROGRAM CODE:

a. Scale Invariant Scale detection.

The scale-invariant feature transform (SIFT) is a computer vision algorithm to detect, describe, and match local features in images, invented by David Lowe in 1999. Applications include object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, individual identification of wildlife and match moving.

SIFT key points of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of key points that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches. The determination of consistent clusters is performed rapidly by using an efficient hash table implementation of the generalized Hough transform.

Each cluster of 3 or more features that agree on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally, the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence.

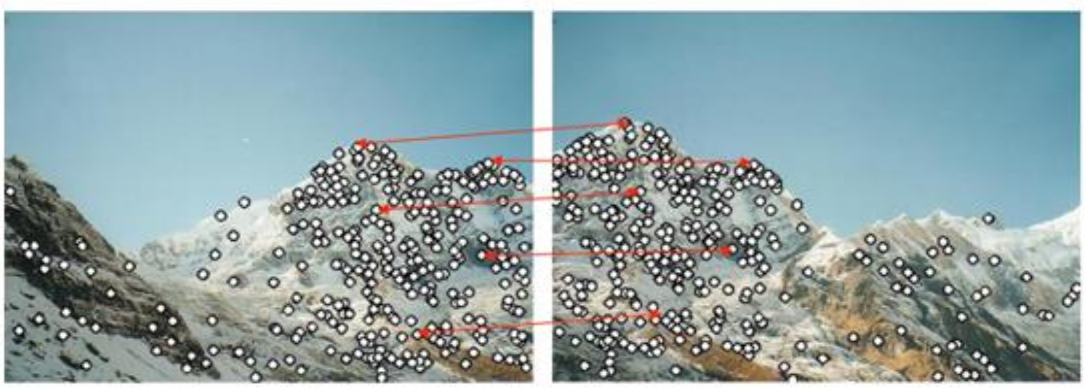


Figure 3-Image Stitching

Working:

Step 1: Scale-space extrema Detection - Detect interesting points (invariant to scale and orientation) using DOG.
Step 2: Keypoint Localization - Determine location and scale at each candidate location, and select them based on stability.
Step 3: Orientation Estimation - Use local image gradients to assign orientation to each localized keypoint. Preserve orientation, scale and location for each feature.

Step 4: Keypoint Descriptor - Extract local image gradients at selected scale around keypoint and form a representation invariant to local shape and illumination distortion them.

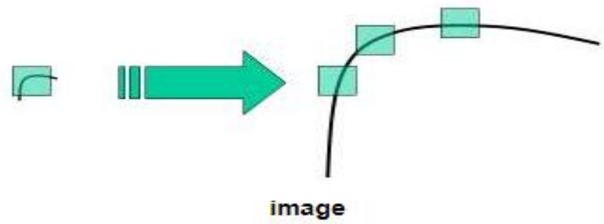


Figure 4 image.jpg

There are mainly four steps involved in SIFT algorithm.

Extrema Detection

From the image above, it is obvious that we can't use the same window to detect keypoints with different scale. It is OK with small corner. But to detect larger corners we need larger windows. For this, scale-space filtering is used. In it, Laplacian of Gaussian is found for the image with various σ values. LoG acts as a blob detector which detects blobs in various sizes due to change in σ. In short, σ acts as a scaling parameter. For eg, in the above image, gaussian kernel with low σ gives high value for small corner while gaussian kernel with high σ fits well for larger corner. So, we can find the local maxima across the scale and space which gives us a list of (x,y,σ) values which means there is a potential keypoint at (x,y) at σ scale.

But this LoG is a little costly, so SIFT algorithm 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 σ, let it be σ and kσ. This process is done for different octaves of the image in Gaussian Pyramid. It is represented in below image:

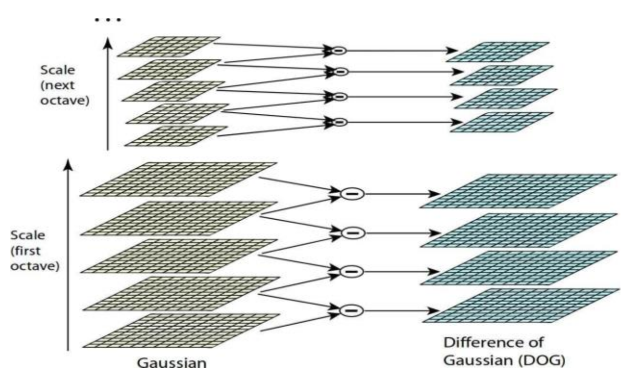


Figure 5. sift_dog.jpg image

Once this DoG are found, images are searched for local extrema over scale and space. For eg, one pixel in an image is compared with its 8 neighbours as well as 9 pixels in next scale and 9 pixels in previous scales. If it is a local extrema, it is a potential keypoint. It basically means that keypoint is best represented in that scale. It is shown in below image:

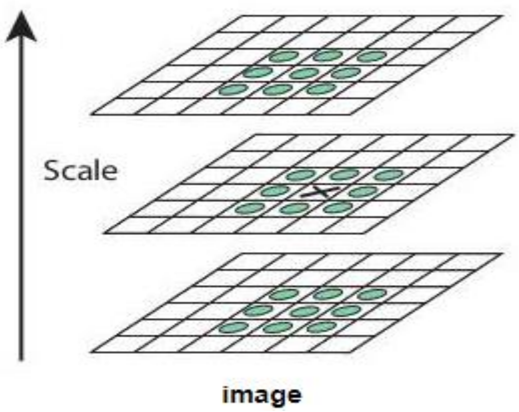


Figure 6 sift_local_extrema.jpg image

Regarding different parameters, the paper gives some empirical data which can be summarized as, number of octaves = 4, number of scale levels = 5, initial σ=1.6, k=√2etc as optimal values.

Step 2: Keypoint Localization

Once potential keypoints locations are found, they have to be refined to get more accurate results. They used Taylor series expansion of scale space to get more accurate location of extrema, and if the intensity at this extremum is less than a threshold value (0.03 as per the paper), it is rejected. This threshold is called contrastThreshold in OpenCV

DoG has higher response for edges, so edges also need to be removed. For this, a concept similar to Harris corner detector is used. They used a 2x2 Hessian matrix (H) to compute the principal curvature. We know from Harris corner detector that for edges, one eigen value is larger than the other. So here they used a simple function,

If this ratio is greater than a threshold, called edgeThreshold in OpenCV, that keypoint is discarded. It is given as 10 in paper.

So, it eliminates any low-contrast keypoints and edge keypoints and what remains is strong interest points.

Step 3: Orientation Assignment

Now an orientation is assigned to each keypoint to achieve invariance to image rotation. A neighbourhood 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 (It is weighted by gradient magnitude and gaussian-weighted circular window with σ equal to 1.5 times the scale of keypoint). The highest peak in the histogram is taken and any peak above 80% of it is also considered to calculate the orientation. It creates keypoints with same location and scale, but different directions. It contribute to stability of matching.

Step 4: Keypoint Descriptor

Now keypoint descriptor is created. A 16x16 neighbourhood around the keypoint is taken. It is divided 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.

Step 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. If it is greater than 0.8, they are rejected. It eliminates around 90% of false matches while discards only 5% correct matches, as per the paper.

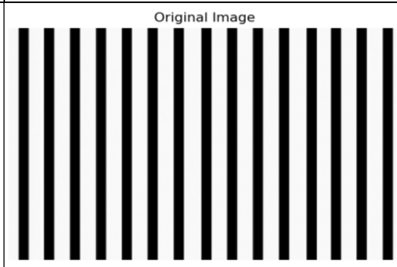
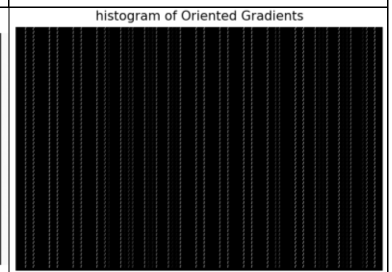
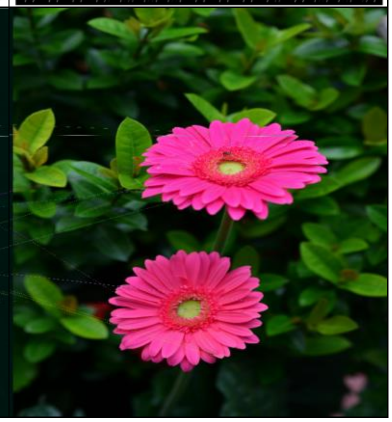
Implementation in OpenCV:

ALGORITHM:

- Read two images for feature matching
- Intialize SIFT DETECTOR
- Fast keypoint detector and BRIEF
- Find the keypoints and descriptors with SIFT
- Create BFMatcher object
- Match Descriptor
- Sort them in the order of thier distance
- #Draw the First 10 matches

PROGRAM CODE:

INPUT & OUTPUT:

Sr. No.	INPUT	OUTPUT
1.		
2.		

CONCLUSION: We learn how to use Hough Transform and SIFT used for a feature detection method.

DISCUSSION QUESTIONS:

1. Can you explain the concept of scale-invariance in feature detection?
2. What are the difference between SIFT and SURF.
3. What is Feature detectors'
4. What is Feature Descriptor in image processing.
5. Why Is It Necessary to Use Feature Descriptors?

REFERENCES:

- Computer Vision: Algorithms and Applications, Richard Szeliski, 2010, Springer.
- Computer Vision - A modern Approach, D. Forsyth, J. Ponce, 2nd Edition 2011, Pearson India.
- OpenCV Computer Vision with Python, Joseph Howse, 2013, Packt Publishing.
- Dictionary of Computer Vision and Image Processing, R. B. Fisher, T. P. Breckon, K. Dawson-Howe, A. Fitzgibbon, C. Robertson, E. Trucco, C. K. I. Williams, 2nd Edition, 2014, Wiley.
- https://www.geeksforgeeks.org/digital-image-processing-basics/
- https://www.javatpoint.com/digital-image-processing-tutorial
- https://www.simplilearn.com/image-processing-article
- https://www.mygreatlearning.com/blog/digital-image-processing-explained/