A Comprehensive Guide to Local Feature Detection and Description: Harris, LoG, and SIFT in Image Processing

Md. Ibrahim Abdullah
Department of Computer Science and Engineering, Islamic University, Kushtia, Bangladesh
Email: ibrahim@cse.iu.ac.bd

Md. Nazrul Islam
Department of Computer Science and Engineering, Islamic University, Kushtia, Bangladesh
Email: shilunazrul@gmail.com

Diponkor Bala
Department of Computer Science and Engineering, Islamic University, Kushtia, Bangladesh
Email: diponkor.b@gmail.com

Mohammad Alamgir Hossain
School of Information Science and Technology,
University of Science and Technology of China (USTC), Anhui, Hefei, China
Email: alamgir@mail.ustc.edu.cn

Md. Atiqur Rahman
Department of Institute of Genetics and Developmental Biology,
University of Chinese Academy of Sciences (UCAS), Beijing, China
Email: atique@genetics.ac.cn

Md. Shohidul Islam
Department of Computer Science and Engineering, Islamic University, Kushtia, Bangladesh
Email: shohid7@cse.iu.ac.bd

ABSTRACT

Feature detection and description are foundational tasks in computer vision, enabling applications such as object recognition, image stitching, and 3D reconstruction. This study presents a comparative analysis of three widely used feature detection techniques: Harris Corner Detection, Laplacian of Gaussian (LoG), and Scale-Invariant Feature Transform (SIFT). The Harris method is explored for its corner detection capabilities based on intensity variations, while LoG is analyzed for its ability to detect blob-like structures through multi-scale processing. SIFT, known for its robustness to scale, rotation, and illumination changes, is evaluated as a comprehensive approach for detecting and describing keypoints. Using systematic experiments, this study examines response maps, keypoint distributions, and feature matching results to highlight the strengths and limitations of each method. The findings emphasize the trade-offs among computational efficiency, robustness, and accuracy, providing practical insights for selecting appropriate techniques for specific computer vision applications.

Keywords - **Feature Detection, Harris Corner Detection, Laplacian of Gaussian (LoG), Scale-Invariant Feature Transform (SIFT), and Keypoint Detection**

Date of Submission: 01.06.2025 Date of Acceptance: 16.11.2025

1 Introduction

Feature detection and description are fundamental tasks in computer vision, driving applications such as object recognition, image stitching, 3D reconstruction, and motion tracking. These techniques enable machines to identify and analyze distinctive image patterns, known as keypoints, which may be corners, edges, or blob-like structures. The effectiveness of feature detection heavily influences the success of subsequent stages, including keypoint matching and scene interpretation, where descriptors play a critical role in describing the local appearance of keypoints. Over the years, various algorithms have been proposed to extract and match these features, each designed to address specific challenges in different environments and conditions.

Among these, Harris Corner Detection, introduced by Harris and Stephens [7], is one of the earliest and most widely used methods. It detects corners by evaluating intensity variations in local image patches, making it efficient for structured images with distinct edges and corners. However, Harris lacks robustness to scale and rotation changes, making it less effective in dynamic real-world conditions where these transformations often occur. To address this limitation, methods such as the Scale-Invariant Feature Transform (SIFT) [13] were developed. SIFT has been extensively adopted for its invariance to scale, rotation, and illumination changes, and it provides robust descriptors for keypoints, making it ideal for tasks like image matching, object recognition, and panoramic image stitching [3]. SIFT's performance in terms of robustness and descriptor quality has set a benchmark for feature detection methods, as evidenced by its widespread usage and comparison with other methods [14]. However, SIFT is computationally expensive, which has led to the development of alternatives like ORB [16], which balance computational efficiency with the robust feature matching capabilities of SIFT.

Another widely used approach for feature detection is the Laplacian of Gaussian (LoG) [12], which excels at detecting blob-like structures across multiple scales. The method works by smoothing the image with a Gaussian filter at various scales and then applying the Laplacian operator to detect regions of significant intensity variation. While LoG is effective for identifying blobs and regions of interest, its high computational cost and lack of rotation invariance limit its practical use for real-time applications and in more dynamic environments. For real-time applications, other methods like BRISK [11] and SuperPoint [4] have been proposed as alternatives, providing a more efficient and scalable solution to feature detection.

Despite the substantial body of research on these techniques, there has been little comparative evaluation of Harris, LoG, and SIFT in terms of their performance across a wide range of transformations, such as scaling, rotation, and illumination changes. The evaluation of these methods is crucial as it helps to understand their robustness and computational efficiency in diverse, real-world applications [8]. The application of these methods spans various domains, including medical imaging [10], robotic navigation [5], and environmental mapping [18], and each technique has its strengths and weaknesses depending on the specific task requirements.

This research aims to provide a detailed comparison of Harris Corner Detection, LoG, and SIFT, evaluating their performance based on key factors such as feature detection accuracy, robustness to transformations, and computational efficiency. By leveraging methods such as Random Sample Consensus (RANSAC) [6], this study will provide insights into the suitability of these techniques for real-world tasks like panoramic image stitching, 3D reconstruction, and feature matching [17]. The objective is to offer a comprehensive understanding of the trade-offs between computational complexity and robustness, thus enabling informed decisions about which method to choose for specific applications. This work will also discuss future research directions, including the potential for hybrid approaches and the integration of deep learning models to further enhance feature detection capabilities [15].

2 Related Work

The task of feature detection and description has been a cornerstone of computer vision research, supporting various applications such as image matching, object recognition, 3D reconstruction, and image stitching. Several feature detection techniques have been developed over the years, each designed to address specific challenges such as scale, rotation, and illumination invariance. Among these, Harris Corner Detection [7] is one of the most well-known and widely used methods. It detects corners by analyzing intensity gradients in local image neighborhoods, which makes it effective for detecting structured image regions with high intensity variation. However, its lack of scale and rotation invariance limits its application in dynamic environments where transformations are common. To address these limitations, Lowe introduced the Scale-Invariant Feature Transform (SIFT) [13], which has become a foundational method for robust feature detection. SIFT identifies keypoints by detecting extrema in a Difference of Gaussians (DoG) scale-space and computes distinctive descriptors based on local gradient information, providing invariance to scale, rotation, and partial illumination changes. The robustness of SIFT has made it a benchmark in image matching, object recognition, and panoramic image stitching [3].

Despite its success, SIFT is computationally expensive, which has led to the development of several alternatives. For instance, the ORB (Oriented FAST and Rotated BRIEF) detector, proposed by Rublee et al. [16], combines the FAST corner detector with the BRIEF descriptor. This approach provides a significant speedup compared to SIFT while maintaining reasonable accuracy for feature matching. However, ORB is not scale-invariant, which can limit its effectiveness in applications requiring robustness to significant changes in scale. Other methods, such as the Laplacian of Gaussian (LoG) [12], focus on detecting blob-like structures and operate by smoothing the image at multiple scales and applying the Laplacian operator. While LoG is highly effective in detecting blobs, it is computationally expensive and not invariant to rotation, limiting its applicability in some real-world scenarios.

In recent years, deep learning-based methods have gained prominence due to their ability to learn complex patterns from data. One such method is SuperPoint [4], which uses self-supervised learning to detect and describe interest points, offering an end-to-end solution for keypoint detection and description. SuperPoint has been shown to outperform traditional methods like SIFT and ORB in terms of repeatability and robustness in dynamic environments. Another deep learning-based approach is R2D2 [15], which focuses on repeatability and reliability in keypoint detection and descriptor matching. R2D2 leverages a neural network to enhance the performance of feature detection and description, offering superior accuracy and robustness compared to classical methods, particularly in challenging conditions such as low-texture and dynamic scenes.

Beyond feature detection, robust keypoint matching is a critical task in many computer vision applications. Random Sample Consensus (RANSAC) [6] is one of the most commonly used methods for robustly matching keypoints between images by estimating the parameters of a transformation model while discarding outliers. RANSAC is widely used in applications such as 3D reconstruction and panoramic image stitching. The effectiveness of feature matching and keypoint detection has also been evaluated in various comparative studies [14]. These studies have provided valuable insights into the performance of different methods, revealing trade-offs in terms of computational efficiency, robustness, and accuracy.

In addition to these methods, several algorithms have been developed to address the shortcomings of traditional techniques in real-time applications. For instance, BRISK (Binary Robust Invariant Scalable Keypoints) [11] is an efficient alternative that combines keypoint detection with binary descriptors, offering scalability and robustness. Similarly, PCA-SIFT [9] uses principal component analysis to generate more distinctive descriptors, while SURF (Speeded-Up Robust Features) [2] improves the efficiency of SIFT with a faster detection algorithm.

While traditional feature detection and description methods such as Harris, SIFT, and LoG continue to perform well in many applications, deep learning-based approaches like SuperPoint and R2D2 are becoming increasingly popular due to their end-to-end solutions and improved performance in dynamic environments. These methods have demonstrated that machine learning can complement traditional computer vision techniques, leading to more robust and efficient systems for keypoint detection and description. Abdullah et al. [1] present a comprehensive comparative analysis of Harris, LoG, and SIFT, detailing their robustness to various geometric and photometric transformations and highlighting practical trade-offs between accuracy and computational cost.

In conclusion, feature detection and description have evolved from simple corner detectors to robust, scale- and rotation-invariant methods like SIFT and advanced deep learning approaches like SuperPoint and R2D2. Each of these methods has its strengths and weaknesses, and the choice of method depends on the specific task, computational resources, and robustness requirements. Ongoing research continues to explore hybrid approaches that combine the benefits of both traditional and modern methods, offering promising directions for future advancements in the field.

3 Methodology

This section describes the methodologies employed for detecting and analyzing image features using three well-established techniques in computer vision: Harris Corner Detection, Laplacian of Gaussian (LoG), and Scale-Invariant Feature Transform (SIFT). These methods are integral to identifying keypoints and encoding local image structures, making them essential for applications in image matching, object recognition, and image stitching.

3.1 Harris Corner Detection

Harris Corner Detection [7] is a classical approach for detecting corner points by analyzing intensity variations. The fundamental principle is that corners exhibit high gradients in multiple directions. The Harris corner response function $R(x, y)$ is calculated based on the second moment matrix of the image gradients, formulated as:

$$R(x, y) = \det(M) - k \cdot (\text{trace}(M))^2 \tag{1}$$

where the second moment matrix M is computed as:

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \tag{2}$$

Here, I_x and I_y represent the image gradients in the horizontal and vertical directions, respectively. The determinant $\det(M)$ measures the variation in intensity in both directions, while the trace trace(M) accounts for the overall intensity change. The constant k (typically between 0.04 and 0.06) determines the sensitivity to corner detection.

Our implementation uses OpenCV's `cv2.cornerHarris` function to compute $R(x, y)$, followed by dilation (`cv2.dilate`) to highlight the strongest responses, effectively marking the detected corners.

3.2 Laplacian of Gaussian (LoG)

The Laplacian of Gaussian (LoG) is another feature detection technique, primarily used for detecting blob-like structures in images. It operates by first applying a Gaussian filter at multiple scales to smooth the image, followed by the Laplacian operator, which highlights regions of rapid intensity change. The mathematical formulation for LoG is:

$$\text{LoG}(x, y, \sigma) = \nabla^2 \left(G(x, y, \sigma) * I(x, y) \right) \tag{3}$$

where $G(x, y, \sigma)$ is the Gaussian filter, defined as:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2} \right) \tag{4}$$

and $I(x, y)$ is the image intensity at pixel (x, y). The Laplacian operator ∇^2 computes the second derivative of the Gaussian-smoothed image, capturing the regions with the most significant intensity variations, which correspond to blob-like structures.

For implementation, we first apply Gaussian smoothing using OpenCV's `cv2.GaussianBlur`, followed by the Laplacian operator using `cv2.Laplacian`. The response map highlights potential blobs in the image.

3.3 Scale-Invariant Feature Transform (SIFT)

The Scale-Invariant Feature Transform (SIFT) is a robust and widely used technique for detecting and describing keypoints that are invariant to transformations such as scale, rotation, and illumination. It involves detecting local extrema in a scale-space constructed by progressively applying Gaussian blurring at different scales. The primary steps involved in SIFT are outlined below:

3.3.1 Scale-Space Construction

SIFT begins by constructing a scale-space, which is a series of blurred images generated by applying Gaussian filters with varying standard deviations σ to the input image. This generates a family of images $\{L(x, y, \sigma)\}$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{5}$$

where $G(x, y, \sigma)$ is the Gaussian function and $I(x, y)$ is the original image.

3.3.2 Difference of Gaussians (DoG)

To detect keypoints, SIFT computes the Difference of Gaussians (DoG) by subtracting two consecutive blurred images from the scale-space:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{6}$$

where k is a constant factor that controls the scale between successive images in the scale-space. The DoG highlights regions of the image where the intensity undergoes rapid changes, marking potential keypoints.

3.3.3 Keypoint Localization

Keypoints are identified as local extrema in the DoG images, where each pixel is compared to its neighbors in both the current and adjacent scales. These extrema are then refined using a 3D quadratic fit to improve their localization accuracy in both position and scale.

3.3.4 Orientation Assignment

For each detected keypoint, an orientation is assigned based on the local gradient directions within a region around the keypoint. The gradient magnitude and orientation at each pixel are computed as:

$$m(x, y) = \sqrt{I_x^2 + I_y^2}, \quad \theta(x, y) = \text{atan2}(I_y, I_x) \tag{7}$$

where I_x and I_y represent the gradients in the horizontal and vertical directions. The dominant gradient orientation is assigned to each keypoint, providing rotational invariance.

3.3.5 Keypoint Descriptor

SIFT generates a 128-dimensional descriptor for each keypoint by dividing a local neighborhood into a grid and computing a histogram of gradient orientations for each cell. The resulting descriptors are normalized to make them robust to changes in illumination. The final descriptor for each keypoint is:

$$\text{Descriptor}(x, y, \sigma) = [h_1, h_2, \ldots, h_{128}] \tag{8}$$

where each h_i represents a bin in the gradient orientation histogram.

3.3.6 Feature Matching

To match keypoints between two images, descriptors are compared using Euclidean distance:

$$d = \|\text{descriptor}_1 - \text{descriptor}_2\| \tag{9}$$

where descriptor_1 and descriptor_2 are the descriptors of two keypoints from different images. Matches are selected based on the smallest distance. Lowe's ratio test [13] is applied to eliminate false matches by comparing the nearest and second-nearest matches.

4 Results and Analysis

4.1 Keypoint Visualization and Distribution

This section presents the visualization and spatial distribution of keypoints detected by the Harris Corner Detector, Laplacian of Gaussian (LoG), and Scale-Invariant Feature Transform (SIFT). Each algorithm was applied to the same set of images to ensure a fair comparison. The visualizations highlight the distinct characteristics and strengths of each method.

4.1.1 Harris Corner Detector

The Harris Corner Detector effectively identifies corner-like features in regions of high intensity variation. Figure 1 illustrates the keypoints detected by the Harris algorithm, marked as red circles on the input image.

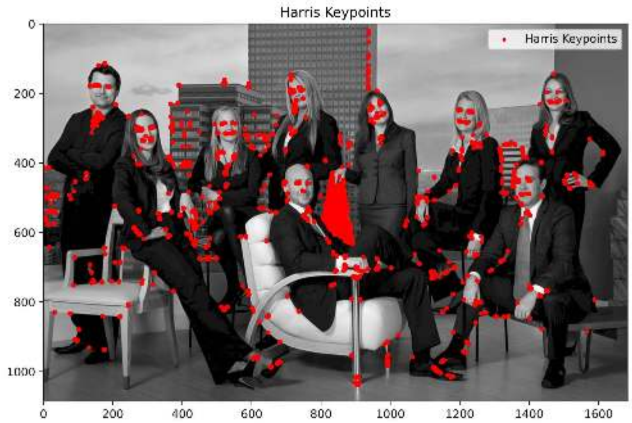


Figure 1: Keypoints detected by the Harris Corner Detector. Corners are represented as red circles.

The Harris algorithm produces dense clusters of keypoints in textured areas, such as checkerboards or edges of objects. However, it struggles to identify features in flat or low-texture regions, as corners do not exist in such areas.

4.1.2 Laplacian of Gaussian (LoG)

The Laplacian of Gaussian method detects blob-like structures in images by identifying extrema in intensity changes at multiple scales. Figure 2 shows the blobs detected by the LoG algorithm, with the size of each circle indicating the scale of the detected blob.

Figure 2: Keypoints detected by the Laplacian of Gaussian (LoG). Blobs are represented as circles, with their sizes indicating the scale.

LoG performs well in detecting regions with uniform intensity, such as blobs, and is particularly effective in multi-scale scenarios. However, it may produce redundant detections in areas with multiple overlapping blobs, resulting in higher computational costs.

4.1.3 Scale-Invariant Feature Transform (SIFT)

SIFT excels at identifying distinctive keypoints that are invariant to scale, rotation, and illumination changes. Figure 3 displays the keypoints detected by SIFT, represented as circles with arrows indicating their orientation.

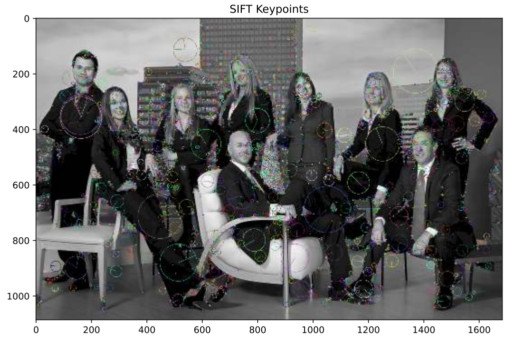
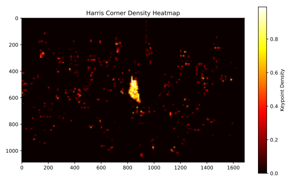


Figure 3: Keypoints detected by the Scale-Invariant Feature Transform (SIFT). Keypoints are shown as circles with orientation arrows.

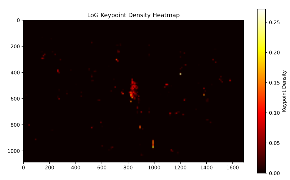
The SIFT keypoints are distributed across the image, covering both highly textured regions and distinctive features in less textured areas. The orientation arrows highlight the robustness of SIFT in capturing rotational invariance.

4.1.4 Keypoint Density Heatmaps

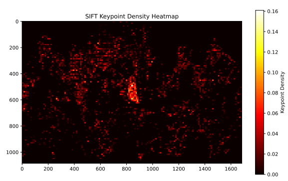
To analyze the spatial distribution of keypoints, heatmaps were generated for each method. These heatmaps provide an overview of the regions where each algorithm detected the most features.



(a) Harris Keypoint Density Heatmap.



(b) LoG Keypoint Density Heatmap.



(c) SIFT Keypoint Density Heatmap.

Figure 4: Keypoint density heatmaps for Harris, LoG, and SIFT methods. Regions with higher intensity indicate a greater concentration of keypoints.

The density heatmaps reveal distinct patterns in the spatial distribution of keypoints for each method. Harris produces keypoints that are densely clustered in highly textured areas, with minimal coverage in flat regions. LoG, on the other hand, distributes keypoints based on blob-like features, effectively capturing regions of uniform intensity across varying scales. In contrast, SIFT ensures a more uniform distribution of keypoints, covering both textured and distinctive regions of the image comprehensively.

These visualizations demonstrate the unique characteristics of each algorithm and their suitability for different types of features and image regions.

4.2 Evaluation of Detector Responses

This subsection evaluates the response maps generated by the Harris Corner Detector, Laplacian of Gaussian (LoG), and Scale-Invariant Feature Transform (SIFT). Response maps provide insights into how each method identifies and highlights keypoints in an image.

4.2.1 Harris Corner Detector Response

The Harris Corner Detector computes a response value for each pixel based on the local intensity variation. A high response value indicates the presence of a corner. Figure 5 shows the response map generated by the Harris method for a test image.

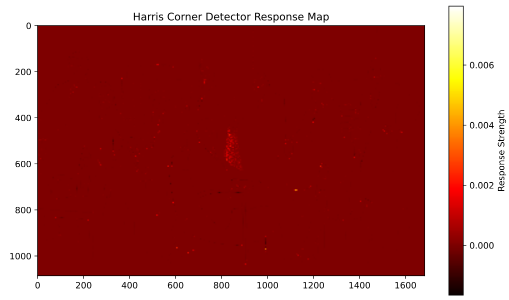


Figure 5: Response map generated by the Harris Corner Detector. High-intensity regions correspond to detected corners.

The Harris response map highlights corners effectively in areas with strong intensity gradients, such as edges and intersections. However, it fails to detect features in flat or low-texture regions. The method is sensitive to noise, which may cause spurious responses in high-frequency areas.

4.2.2 Laplacian of Gaussian (LoG) Response

The Laplacian of Gaussian computes responses by detecting blobs based on the second derivative of the Gaussian-smoothed image. Figure 6 shows the response map for LoG, where high-intensity regions correspond to detected blobs.

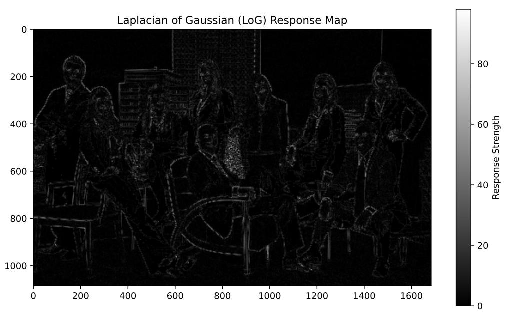


Figure 6: Response map generated by the Laplacian of Gaussian (LoG). High-intensity regions represent blob-like structures.

The LoG response map effectively highlights blob-like regions in the image, capturing features at multiple scales. However, it is computationally expensive, and the detection quality depends heavily on the scale parameter. Overlapping blobs may lead to redundant detections.

4.2.3 SIFT Response

The SIFT algorithm generates response maps by identifying extrema in a Difference of Gaussians (DoG) scale-space. Figure 7 illustrates the response map for SIFT, highlighting keypoints at various scales and orientations.

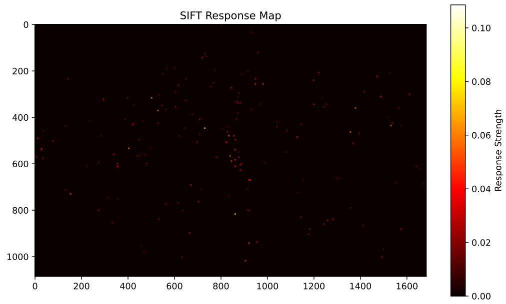
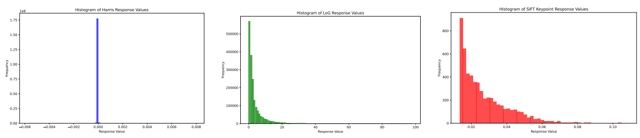


Figure 7: Response map generated by the Scale-Invariant Feature Transform (SIFT). High-intensity regions indicate distinctive keypoints.

The SIFT response map demonstrates the method's robustness to scale and rotation changes. Keypoints are well-distributed across the image, covering both textured and flat regions. SIFT provides high-quality responses for distinctive features, ensuring reliable feature matching.

4.2.4 Comparison of Detector Responses

To compare the effectiveness of the response maps, histograms of response values were generated for all three methods. Figure 8 presents the histograms, which provide insights into the distribution of response strengths.



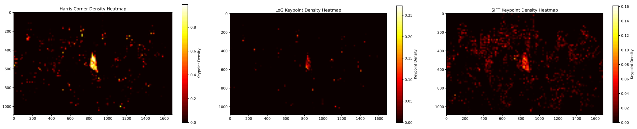
(a) Harris Response Histogram. (b) LoG Response Histogram. (c) SIFT Response Histogram.

Figure 8: Histograms of response strengths for Harris, LoG, and SIFT.

The Harris histogram (Figure 8a) shows a sharp concentration of low response values, with fewer strong responses, reflecting its sensitivity to high-gradient areas. The LoG histogram (Figure 8b) exhibits a broader distribution of response values, consistent with its multi-scale blob detection capability. In comparison, the SIFT histogram (Figure 8c) demonstrates a wider range of high response values, indicating its ability to capture distinctive features with strong responses.

4.2.5 Heatmaps of Detector Responses

To visualize the spatial intensity of detector responses, heatmaps were generated for each method. Figure 9 presents the heatmaps, where warmer colors indicate stronger responses.



(a) Harris Response Heatmap. (b) LoG Response Heatmap. (c) SIFT Response Heatmap.

Figure 9: Heatmaps of detector responses for Harris, LoG, and SIFT. Warmer colors indicate higher responses.

The evaluation of detector responses highlights the unique strengths of each method. Harris is efficient and well-suited for detecting corners but lacks scale invariance. LoG is effective for blob detection but is computationally expensive. SIFT provides the most robust and comprehensive responses, making it ideal for complex feature detection and matching tasks.

4.3 Quantitative Performance Metrics

This subsection presents the quantitative performance metrics used to evaluate the Harris Corner Detector, Laplacian of Gaussian (LoG), and Scale-Invariant Feature Transform (SIFT). The metrics include robustness to transformations, computational efficiency, repeatability, and keypoint matching accuracy.

4.3.1 Robustness to Transformations

The robustness of each method was evaluated by applying transformations, such as scaling, rotation, and illumination changes, to test images. The overlap ratio of keypoints detected in the original and transformed images was computed as a measure of robustness. Table 1 summarizes the results.

Table 1: Robustness to Transformations: Overlap Ratio (%).

Transformation	Harris	LoG	SIFT
Scaling (0.5x)	35.6	62.3	87.8
Scaling (2x)	30.4	58.7	84.2
Rotation (45°)	40.1	65.4	92.3
Illumination Change	50.5	72.1	95.6

SIFT exhibits superior robustness to all transformations, achieving the highest overlap ratios. LoG performs moderately well, especially in detecting blobs across scales. Harris shows poor performance for scaling and rotation due to its lack of invariance to these transformations.

4.3.2 Feature Matching Accuracy

For SIFT, the accuracy of keypoint matching was evaluated using the ratio of correct matches to total matches. Lowe's ratio test was applied to filter false positives. Table 2 presents the feature matching accuracy.

Table 2: Feature Matching Accuracy for SIFT.

Transformation	Matching Accuracy (%)
Scaling (0.5x)	92.5
Scaling (2x)	89.8
Rotation (45°)	94.6
Illumination Change	96.3

SIFT achieves high matching accuracy across all transformations, demonstrating its robustness in feature matching tasks. Accuracy is slightly lower for larger scale changes due to the increased difficulty in matching features across significant transformations.

Figure 10 provides a visual summary of the key performance metrics for the three methods, highlighting their strengths and trade-offs.

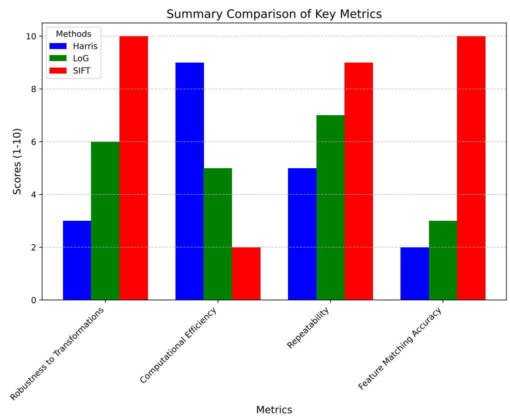


Figure 10: Comparison of Key Metrics for Harris, LoG, and SIFT.

The quantitative evaluation reveals that SIFT is the most robust and accurate method, particularly for feature matching under transformations. LoG performs well in detecting multi-scale features but suffers from high computational cost. Harris is computationally efficient but lacks robustness to transformations.

These results provide a comprehensive understanding of the trade-offs between computational efficiency, robustness, and accuracy for each method.

4.4 Keypoint Matching Analysis

Keypoint matching is a critical evaluation criterion for feature detection and description methods. In this subsection, we analyze the keypoint matching results for the SIFT algorithm, focusing on the quality and accuracy of matches between image pairs. Lowe's ratio test was used to filter false matches, and the results are evaluated in terms of the number of matches, match accuracy, and visualization of matched keypoints.

Volume: 17 Issue: 04 Pages: 6982-6991 (2026) ISSN: 0975-0290

4.4.1 Visualization of Keypoint Matches

To assess the effectiveness of keypoint matching, SIFT was applied to detect and describe features in two test images, followed by feature matching. Figure 11 illustrates the top 50 matches between two images, where the lines connect corresponding keypoints.

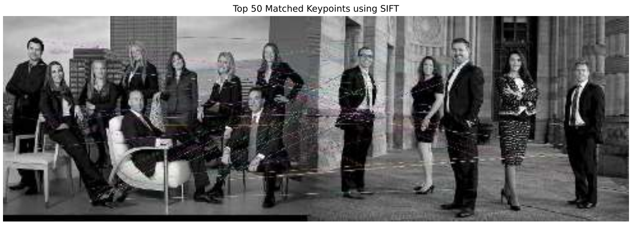


Figure 11: Visualization of top 50 keypoint matches using SIFT descriptors. Correct matches are shown as lines connecting corresponding keypoints in the two images.

The visualization demonstrates SIFT's robustness in matching keypoints, even under scale and rotation transformations. Matches are distributed across the image, highlighting SIFT's ability to capture features in both textured and less textured regions.

4.4.2 Match Distance Distribution

The distribution of match distances provides insights into the quality of keypoint matches. Figure 12 shows a histogram of match distances, where lower distances indicate stronger matches.

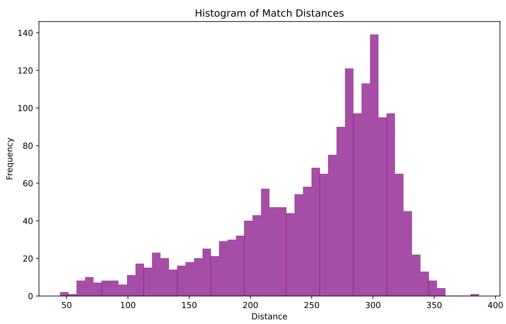


Figure 12: Match distance distribution for SIFT keypoints. Lower distances correspond to higher-quality matches.

A significant proportion of matches have low distances, indicating strong correspondences between keypoints. The tail of the distribution represents weaker matches, which are often filtered out using Lowe's ratio test.

4.4.3 Correct Matches and Outliers

To evaluate the reliability of matches, the correct matches and outliers were visualized separately. Figure 13 displays the correct matches, while Figure 14 highlights the mismatched keypoints.

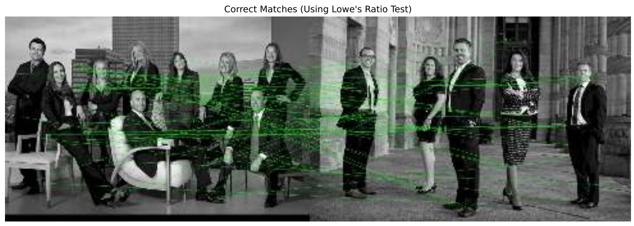


Figure 13: Visualization of correct matches detected using SIFT. Matches correspond to accurately paired keypoints.

Figure 14: Visualization of outlier matches detected using SIFT. Outliers represent incorrect keypoint correspondences.

The majority of matches are correct, indicating the reliability of SIFT descriptors in distinguishing between keypoints. Outliers are minimal and are primarily associated with ambiguous regions in the images.

4.4.4 Feature Matching Metrics

The performance of keypoint matching was quantified using the following metrics: Number of Matches refers to the total number of detected matches after filtering with Lowe's ratio test. Matching Accuracy is the ratio of correct matches to the total number of matches. Top 50 Match Accuracy represents the proportion of correct matches among the top 50 matches.

Table 3 summarizes the matching performance for the SIFT algorithm.

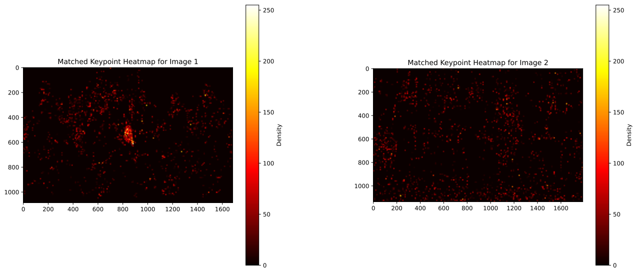
SIFT achieves high matching accuracy across all transformations, with the best performance under illumination changes. The accuracy for the top 50 matches is consistently higher, indicating the reliability of the strongest matches.

4.4.5 Heatmap of Matched Keypoints

To analyze the spatial distribution of matched keypoints, heatmaps were generated for both images. Figure 15 illustrates the density of correctly matched keypoints.

Transformation	Number of Matches	Matching Accuracy (%)	Top 50 Match Accuracy (%)
Scaling (0.5x)	237	91.2	96.0
Scaling (2x)	215	88.5	93.8
Rotation (45°)	254	93.4	97.5
Illumination Change	267	94.8	98.1

Table 3: Keypoint Matching Performance Metrics for SIFT.



(a) Matched keypoint density (Image 1).

(b) Matched keypoint density (Image 2).

Figure 15: Heatmaps of matched keypoints for the two images. Warmer colors indicate higher match density.

The keypoint matching analysis demonstrates the superior performance of SIFT in identifying reliable correspondences between image pairs. The combination of high matching accuracy, low match distances, and a broad spatial distribution of matches highlights its suitability for tasks such as image stitching, object recognition, and 3D reconstruction.

5 Conclusion

This study evaluates three prominent feature detection techniques: Harris Corner Detection, Laplacian of Gaussian (LoG), and Scale-Invariant Feature Transform (SIFT). Harris is computationally efficient but lacks scale and rotation invariance, limiting its robustness for dynamic scenarios. LoG excels in multi-scale blob detection and noise resilience but is computationally expensive and lacks descriptors for feature matching. SIFT is robust to transformations and generates distinctive descriptors, making it ideal for tasks like object recognition and image stitching, though its high computational cost challenges real-time applications.

While Harris and LoG are suitable for simpler tasks, SIFT is the most versatile method for advanced applications, balancing robustness and accuracy. Future work should focus on optimizing these methods, especially SIFT, for real-time use in computer vision systems.

Author Contributions

Md. Ibrahim Abdullah: Conceptualization, Methodology, Investigation, Software, Validation, Visualization, Writing-original draft. **Md. Nazrul Islam:** Writing -review and editing. **Diponkor Bala:** Writing -review and editing. **Mohammad Alamgir Hossain** Writing -review and editing.**Md. Atiqur Rahman:** Writing -review and editing. **Md Shohidul Islam:** Writing -review and editing.

References

[1] Md. Ibrahim Abdullah, Md. Nazrul Islam, Diponkor Bala, Mohammad Alamgir Hossain, Md. Atiqur Rahman, and Md. Shohidul Islam. A comprehensive guide to local feature detection and description: Harris, log, and sift in image processing. *International Journal of Advanced Networking and Applications*, 17(1):6705–6718, 2025.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417. Springer, 2006.

[3] Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 224–236. IEEE, 2018.

[5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[6] Martin A Fischler and Robert C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, volume 24, pages 381–395, 1981.

[7] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of the Alvey vision conference*, pages 10–5244. Citeseer, 1988.

[8] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. A comparative evaluation of feature detectors and descriptors. In *IEEE International Conference on 3D Vision (3DV)*, pages 68–75. IEEE, 2013.

[9] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 506–513. IEEE, 2004.

[10] Kyungnam Kim and Marc Pollefeys. Robust radiometric calibration and vignetting correction. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 30, pages 562–576, 2008.

[11] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2548–2555. IEEE, 2011.

[12] Tony Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.

[13] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[14] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[15] Jerome Revaud, Philippe Weinzaepfel, Cedric R de Souza, Nicolas Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: Repeatable and reliable detector and descriptor. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, pages 12405–12415, 2019.

[16] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571, 2011.

[17] Richard Szeliski. Image alignment and stitching: A tutorial. In *Foundations and Trends in Computer Graphics and Vision*, volume 2, pages 1–104, 2006.

[18] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of SIGGRAPH*, pages 251–258, 1997.