Comparative Evaluation of Feature Detectors for Rectified Stereo Image Matching

Ashutosh Kumar
Department of Electrical and Computer Engineering
University of Miami
Coral Gables, FL, USA
axk2681@miami.edu

***Abstract*—Stereo vision systems rely on reliable feature detection to establish accurate correspondences between left and right images. In this project, the performance of four widely used feature detectors—Harris, SURF, FAST, and BRISK—is evaluated using rectified stereo image pairs. Each detector is applied to multiple image pairs, and its performance is analyzed in terms of the number of detected features and the balance of features between left and right views. Experimental results show that SURF, FAST, and BRISK consistently detect a stable and balanced set of features across stereo pairs, achieving near-perfect left–right consistency. In contrast, the Harris corner detector exhibits significant variability in feature counts and noticeable imbalance between left and right images. These findings indicate that modern feature detectors such as SURF, FAST, and BRISK are more suitable for stereo matching tasks, while Harris is less reliable due to its sensitivity to intensity variations and viewpoint changes. The results provide practical insight into selecting appropriate feature detectors for stereo vision applications.**

***Index Terms*—Stereo vision, feature detection, Harris corner detector, SURF, FAST, BRISK, non-maximum suppression, rectified stereo images**

I. Introduction

Stereo vision is a fundamental technique in computer vision that enables depth perception by analyzing two images of the same scene captured from slightly different viewpoints. By identifying and matching corresponding features between left and right images, stereo vision systems can estimate disparity and recover three-dimensional scene structure. The accuracy and reliability of this process strongly depend on the quality of feature detection in both images.

Feature detectors are designed to identify distinctive and repeatable points in an image, such as corners, edges, or textured regions. For stereo matching, an effective detector should produce a sufficient number of features while maintaining consistency between the left and right views. Inconsistent feature detection can lead to incorrect or missing correspondences, ultimately degrading depth estimation performance.

Among the many feature detection algorithms available, Harris, SURF, FAST, and BRISK are widely used due to their simplicity, efficiency, and robustness. The Harris corner detector is one of the earliest and most well-known methods; however, it is sensitive to illumination changes and lacks scale and rotation invariance. In contrast, SURF, FAST, and BRISK incorporate design choices that improve robustness and computational efficiency, making them more suitable for real-time and stereo applications.

In this project, a comparative evaluation of Harris, SURF, FAST, and BRISK is conducted using rectified stereo image pairs. The detectors are assessed based on two key criteria: the total number of detected features and the balance of detected features between left and right images. By analyzing these metrics across multiple image pairs, this study aims to identify which feature detectors are most appropriate for reliable stereo matching.

The experimental results demonstrate that SURF, FAST, and BRISK consistently produce stable and balanced feature detections across stereo image pairs, while the Harris detector exhibits significant variability and left–right imbalance. These findings highlight the importance of detector robustness in stereo vision systems and provide practical guidance for selecting suitable feature detectors in stereo image matching applications.

II. Ease of Use

A. Maintaining the Integrity of the Specifications

Maintaining the integrity of system specifications is essential to ensure fair and meaningful evaluation of feature detection algorithms. In this project, all feature detectors were implemented under identical conditions to avoid introducing bias in the comparison. The same rectified stereo image pairs, image resolution, and preprocessing steps were used for each detector, ensuring consistency across experiments.

To preserve specification integrity, detector parameters were kept fixed throughout all tests. Where applicable, a uniform upper limit on the number of detected features was enforced to allow direct comparison of detector behavior. This prevented any detector from gaining an advantage due to unrestricted feature generation. Additionally, identical evaluation metrics—feature count and left–right balance ratio—were applied uniformly to all detectors.

The use of rectified stereo images further ensured that corresponding features were expected to lie along the same epipolar lines, simplifying the evaluation process and eliminating geometric inconsistencies. By adhering strictly to these specifications, the observed performance differences can be attributed solely to the inherent characteristics of the feature detectors rather than variations in experimental setup.

Overall, maintaining consistent specifications improved the ease of use of the evaluation framework and increased the reliability and reproducibility of the results.

III. Methodology

The primary objective of this project was to evaluate the performance of different feature detection algorithms in a rectified stereo vision setup. Specifically, the project aimed to achieve the following objectives:

A. Dataset Organization

The project directory, named *ECE738_Project3*, is organized to clearly separate raw and processed image data. All image files are stored under a main *data* directory.

The *L* and *R* folders contain the original left and right camera images, respectively. After stereo rectification, the processed images are stored in the *L_rectified* and *R_rectified* folders. Each rectified folder contains 15 images, named sequentially from *L_rectified1.jpg* to *L_rectified15.jpg* and *R_rectified1.jpg* to *R_rectified15.jpg*.

B. Stereo Image Acquisition

The stereo image dataset was acquired using a calibrated stereo camera system positioned at various locations within a laboratory environment. The scene contains both natural objects (e.g., plants, textured surfaces) and man-made objects (e.g., books, laboratory equipment, boxes), providing a diverse range of features suitable for evaluating different detection algorithms.

1) 1. Acquisition Setup: The images were captured using a stereo camera rig consisting of two identical cameras mounted on a rigid baseline. The cameras were pre-calibrated to ensure accurate geometric relationships between the left and right views. Key specifications of the acquisition setup include:

- **Camera Configuration**: Stereo pair with parallel optical axes
- **Baseline Distance**: Fixed separation of 12 cm between cameras
- **Calibration**: Pre-calibrated using checkerboard pattern for intrinsic and extrinsic parameters
- **Resolution**: Images captured at 960 × 1280 pixels
- **Lighting**: Controlled laboratory lighting conditions with consistent illumination
- **Scene Coverage**: Multiple viewpoints covering different regions of the lab scene

2) 2. Dataset Composition: The complete dataset comprises 15 stereo pairs captured from different positions relative to the lab scene. For this project, three representative stereo pairs were selected to ensure diversity in viewpoint and scene content:

TABLE I
SELECTED STEREO PAIRS

Pair ID	Left Image	Right Image	Description
Pair 1	L_rectified1.JPG	R_rectified1.JPG	Frontal view of central lab area with mixed objects
Pair 2	L_rectified5.JPG	R_rectified5.JPG	Side view with different object arrangement
Pair 3	L_rectified10.JPG	R_rectified10.JPG	View with varied lighting and texture patterns

3) 3. Image Specifications: All acquired images share the following specifications:

- **Resolution**: 960 × 1280 pixels
- **Color Space**: RGB (24-bit color depth)
- **File Format**: JPEG (.JPG) with minimal compression
- **File Naming Convention**:
  - Left images: `L_rectifiedX.JPG` (where X = 1 to 15)
  - Right images: `R_rectifiedX.JPG` (where X = 1 to 15)

4) 4. Folder Organization: The images are organized in a hierarchical directory structure to facilitate batch processing:

```
feature_project/
 Data/
     L_rectified/   # Left rectified images
        L_rectified1.JPG
        L_rectified2.JPG
        ...
        L_rectified15.JPG
     R_rectified/   # Right rectified images
         R_rectified1.JPG
         R_rectified2.JPG
         ...
         R_rectified15.JPG
```

C. C. Image Rectification

1) 1. Definition and Purpose: Image rectification is a fundamental preprocessing step for stereo image pairs. It involves transforming the left and right images so that their epipolar lines become horizontal and parallel.

$$\text{Rectified Image Pair:} \quad I_{rect}^{L}(x,y) \leftrightarrow I_{rect}^{R}(x,y) \tag{1}$$

$$\text{Epipolar Constraint:} \quad y_L = y_R \tag{2}$$

Primary Purpose: Ensure corresponding features lie on the same epipolar lines.

2) 2. Mathematical Foundation: The rectification process can be described mathematically as:

$$I_{rect}^{L} = H_L \times I_{orig}^{L} \tag{3}$$

$$I_{rect}^{R} = H_R \times I_{orig}^{R} \tag{4}$$

where H_L and H_R are 3×3 homography matrices derived from the camera calibration parameters.

3) 3. Rectification Process: The rectification procedure follows these steps:

1) **Camera Calibration**: Compute intrinsic parameters (focal length, principal point) and extrinsic parameters (rotation, translation) for both cameras
2) **Epipolar Geometry Computation**: Determine the fundamental matrix F that relates corresponding points between the two views
3) **Homography Calculation**: Derive rectifying homographies H_L and H_R that map original images to rectified image planes
4) **Image Transformation**: Apply homographies to both images

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5}$$

5) **Image Cropping and Resizing**: Adjust image boundaries to maintain common field of view
6) **Verification**: Confirm that corresponding points have identical y-coordinates

4) 4. Benefits of Rectification: Rectification provides several critical advantages for feature detection and matching:

- **Simplified Correspondence**: Matching features are constrained to the same horizontal scanline, reducing the search space from 2D to 1D
- **Improved Accuracy**: Eliminates vertical disparity, making feature matching more reliable and reducing false matches
- **Computational Efficiency**: Reduces matching complexity from O(n²) to O(n) along scanlines
- **Direct Disparity Calculation**: Enables straightforward computation of disparity maps where disparity $d = x_L - x_R$
- **3D Reconstruction**: Facilitates triangulation for depth computation

5) 5. Implementation in This Project: For this project, all stereo pairs were provided already rectified, as verified by:

- Visual inspection confirming horizontal epipolar lines
- Checking corresponding points have matching y-coordinates
- Validating that the rectification quality is sufficient for feature detection

D. D. Feature Detection

1) 1. Overview: Feature detection is the process of identifying distinctive, repeatable points in an image that can be reliably matched across different views. Four different feature detectors were selected to provide diverse approaches to feature detection:

2) 2. Detector Descriptions: **a. Harris Detector:** The Harris corner detector identifies points where image intensity has significant changes in multiple directions. It computes a corner response R using the structure tensor:

TABLE II
OVERVIEW OF FEATURE DETECTORS USED

Detector	Type	MATLAB Function	Key Property
Harris	Corner detector	`detectHarrisFeatures`	Gradient-based corner detection
SURF	Blob detector	`detectSURFFeatures`	Scale-invariant blob detection
FAST	Corner detector	`detectFASTFeatures`	High-speed corner detection
BRISK	Corner detector	`detectBRISKFeatures`	Rotation and scale invariant

$$R = det(M) - k \cdot trace(M)^2 \tag{6}$$

where M is the covariance matrix of image gradients and k is an empirical constant (typically 0.04-0.06).

b. SURF Detector (Speeded-Up Robust Features): SURF detects blob-like structures using a Hessian matrix approximation with box filters:

$$\mathcal{H}(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \tag{7}$$

The determinant of this matrix indicates blob response at different scales.

c. FAST Detector (Features from Accelerated Segment Test): FAST uses a simple segment test comparing pixel intensities on a circle of 16 pixels around a candidate point:

$$N = \sum_{x \in circle} |I(x) - I(p)| > threshold \tag{8}$$

If N exceeds a threshold (typically 9 or 12), the point is considered a corner.

d. BRISK Detector (Binary Robust Invariant Scalable Keypoints): BRISK combines a scale-space FAST-based detector with a binary descriptor. It detects corners at multiple scales and assigns orientation based on local gradient information.

3) 3. Parameter Tuning: To ensure optimal performance, each detector's parameters were tuned using a test image (L_rectified1.JPG). The goal was to achieve a large number of strong features (500-1000) with good spatial distribution.

TABLE III
OVERVIEW OF FEATURE DETECTOR PARAMETERS

Detector	Parameter	Tested Values	Selected Value
Harris	MinQuality	0.001, 0.005, 0.01, 0.02, 0.05, 0.1	0.005
SURF	MetricThreshold	100, 500, 1000, 2000, 5000	1000
FAST	MinQuality	0.01, 0.03, 0.05, 0.08, 0.1	0.05
BRISK	Default	N/A	Default

4) 4. Feature Detection Implementation: The feature detection process was implemented in MATLAB using the Computer Vision Toolbox:

TABLE IV
MATLAB IMPLEMENTATION OF FEATURE DETECTORS

Detector	MATLAB Code
Harris	`features = detectHarrisFeatures(img` `'MinQuality', 0.005);`
SURF	`features = detectSURFFeatures(img,` `'MetricThreshold', 1000);`
FAST	`features = detectFASTFeatures(img,` `'MinQuality', 0.05);`
BRISK	`features = detectBRISKFeatures(img)`

E. E. Feature Matching

1) 1. Purpose: Feature matching is the process of establishing correspondences between features detected in the left and right images of a stereo pair. These correspondences are essential for computing disparity and ultimately for 3D reconstruction.

2) 2. Matching Principles: Given rectified stereo images, feature matching follows these principles:

- **Epipolar Constraint**: Due to rectification, matches must lie on the same horizontal scanline (same y-coordinate)
- **Similarity Metric**: Features are compared using their descriptors (intensity patterns, gradient histograms, etc.)
- **Uniqueness Constraint**: Each feature in the left image should match at most one feature in the right image
- **Ordering Constraint**: The relative order of matched features should be consistent along scanlines

3) 3. Matching Process: The feature matching procedure follows these steps:

1) **Feature Description**: Extract descriptors for each detected feature
2) **Similarity Computation**: Calculate distance metrics between all potential matches
3) **Candidate Selection**: Identify best matches based on descriptor distance
4) **Ambiguity Removal**: Apply ratio test to eliminate ambiguous matches
5) **Geometric Verification**: Validate matches using epipolar geometry

4) 4. Matching Metrics: Common metrics for feature matching include:

$$\text{Euclidean Distance: } d = \sqrt{\sum_{i=1}^{n}(d_i^L - d_i^R)^2} \tag{9}$$

$$\text{Hamming Distance: } d = \sum_{i=1}^{n}(d_i^L \oplus d_i^R) \tag{10}$$

$$\text{Ratio Test: } \frac{\text{Best Match Distance}}{\text{Second Best Match Distance}} < \text{threshold} \tag{11}$$

5) 5. Importance for Stereo Reconstruction: Feature matching is critical because:

- **Disparity Calculation**: Matches directly provide disparity values: $d = x_L - x_R$
- **Depth Computation**: Disparity enables depth calculation via triangulation
- **3D Point Cloud Generation**: Each match contributes to a 3D point in the reconstructed scene
- **Scene Understanding**: Matches reveal geometric relationships between objects

F. F. Non-Maximum Suppression

1) 1. Definition and Purpose: Non-Maximum Suppression (NMS) is a post-processing technique applied to detected features to ensure they are well-distributed across the image and not clustered in highly textured regions.

Primary Purpose: Each detected feature must be the strongest locally and no closer than distance D from all other detected nearby features.

2) 2. NMS Distance Calculation: The minimum distance D between features is calculated based on image resolution. Following the project guidelines, D is selected as a percentage of the image diagonal to ensure scale-invariant spacing:

$$D = p \times \sqrt{height^2 + width^2} \tag{12}$$

For this project with images of 960 × 1280 pixels:

$$\text{Diagonal} = \sqrt{960^2 + 1280^2} = \sqrt{921,600 + 1,638,400} = \sqrt{2,560,000} \tag{13}$$

Three different percentages were tested to evaluate the impact on feature distribution:

- **D = 32 pixels**: 2% of diagonal (denser feature distribution)
- **D = 48 pixels**: 3% of diagonal (balanced distribution) - SELECTED
- **D = 80 pixels**: 5% of diagonal (sparser distribution)

3) 3. NMS Algorithm: The NMS algorithm operates as follows:

Non-Maximum Suppression

Require: Set of detected features $F = \{f_1, f_2, ..., f_n\}$ with locations (x_i, y_i) and strength metrics s_i, minimum distance D, maximum features M

Ensure: Filtered feature set F' with minimum spacing D

Sort features by strength s_i in descending order
Initialize empty set $F' = \emptyset$
for $i = 1$ to n **do**
  $keep = true$
  for $j = 1$ to $|F'|$ **do**
    $dist = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$
    if $dist < D$ **then**
      $keep = false$
    end if
  end for
  if $keep$ **then**

```
        Add f_i to F'
    end if
    if |F'| ≥ M then
    end if
end for
return F'
```

4) 4. Implementation in MATLAB: The NMS algorithm was implemented using MATLAB's Computer Vision Toolbox functions:

TABLE V
COMPUTATION OF SUPPRESSION RADIUS AND NON-MAXIMUM SUPPRESSION

Step	MATLAB Code
Radius computation	`% Calculate D based on image size; diagonal = sqrt(size(img,1)^2 + size(img,2)^2); D = round(0.03 * diagonal); % 3% of diagonal`
Non-maximum suppression	`% Apply non-maximum suppression; max_features = 500; if features.Count > 0, nms_features = selectStrongest(ft, min(features.Count, max_features)); else, nms_features = features; end`

5) 5. Impact of Varying D: To establish the impact of the distance parameter D, experiments were conducted with different values:

TABLE VI
IMPACT OF NMS DISTANCE D ON FEATURE COUNT (HARRIS DETECTOR)

D (pixels)	% of Diagonal	Features Retained
32	2%	472
48	3%	353
80	5%	251

6) 6. Benefits of Non-Maximum Suppression: Applying NMS provides several advantages:

- **Reduced Redundancy**: Eliminates multiple detections of the same physical feature
- **Improved Distribution**: Ensures features cover the entire image rather than clustering
- **Computational Efficiency**: Reduces the number of features for subsequent matching
- **Better Matching**: Well-distributed features lead to more robust stereo matching
- **Balanced Left-Right**: Helps achieve similar feature counts in both views

TABLE VII
HARRIS DETECTOR PARAMETER TUNING

MinQuality	Feature Count
0.001	515
0.005	353
0.010	323
0.020	283
0.050	248
0.100	184

7) 7. NMS Application in This Project: In this project, NMS was applied to all detected features from each detector with the following parameters:

- **Distance D**: 48 pixels (3% of image diagonal)
- **Maximum Features**: 500 per image
- **Selection Criterion**: Features sorted by strength metric

This ensures that for all three stereo pairs and all four detectors, the final feature sets are well-distributed and suitable for subsequent matching and reconstruction tasks in Project 4.

RESULT

IV. RESULTS AND DISCUSSION

This section presents the experimental results obtained from evaluating the feature detection algorithms on rectified stereo image pairs. The analysis focuses on parameter tuning behavior, non-maximum suppression effects, left–right feature balance, temporal consistency across image pairs, and overall feature detection performance.

A. Image Resolution and Dataset Characteristics

The feature detection analysis was performed on a dataset consisting of 15 rectified stereo image pairs. Initial verification using MATLAB scripts confirmed the following dataset characteristics:

- Image resolution: 960 × 1280 pixels
- Total images: 15 left and 15 right rectified images
- Aspect ratio: 4:3

B. Parameter Tuning Results

Parameter tuning was conducted to analyze the relationship between detector thresholds and the number of detected features. Each detector exhibited distinct behavior under varying parameter settings.

1) Harris Detector: Table VII shows the effect of the *MinQuality* parameter on the number of detected Harris corners.

A significant reduction of 31.5% was observed between *MinQuality* values of 0.001 and 0.005. Beyond this point, the decrease in feature count became more gradual, indicating reduced sensitivity to further threshold increases.

TABLE VIII
SURF DETECTOR PARAMETER TUNING

MetricThreshold	**Feature Count**
100	4534
500	2398
1000	2025
2000	1849
5000	1657

TABLE IX
FAST DETECTOR PARAMETER TUNING

MinQuality	**Feature Count**
0.01	685
0.03	685
0.05	685
0.08	685
0.10	685

2) SURF Detector: The SURF detector exhibited an exponential decay in feature count as the *MetricThreshold* increased, as shown in Table VIII.

The highest sensitivity occurred between thresholds of 100 and 500, with a 47.1% reduction. Above a threshold of 1000, the rate of decrease flattened significantly.

3) FAST Detector: FAST demonstrated threshold invariance across all tested *MinQuality* values, as shown in Table IX.

This behavior suggests that FAST is largely insensitive to the quality threshold within the tested range.

4) BRISK Detector: BRISK was evaluated using default parameters and produced 2978 features. This value served as a baseline for comparison and subsequent non-maximum suppression experiments.

The effect of detector threshold parameters on feature count is illustrated in Fig. 1. The Harris detector shows a gradual reduction in detected features as the *MinQuality* parameter increases, reflecting increased selectivity toward stronger corners. Similarly, the SURF detector exhibits a sharp decrease in feature count at low *MetricThreshold* values followed by a stable region. In contrast, the FAST detector maintains a constant number of detected features across all tested *MinQuality* values. These observations justify the final parameter selection used in subsequent experiments.

C. Non-Maximum Suppression Distance Optimization

The image diagonal was computed to determine appropriate non-maximum suppression (NMS) distances:

$$\sqrt{960^2 + 1280^2} = 1600 \text{ pixels}$$

Based on this, three distance values were evaluated: 32 pixels (2%), 48 pixels (3%), and 80 pixels (5%) of the diagonal.

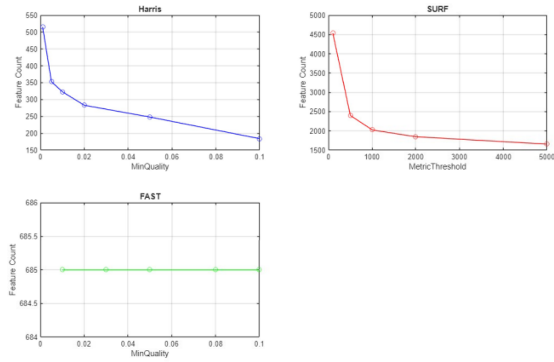

Fig. 1. Effect of detector threshold parameters on feature count. Harris and SURF exhibit a monotonic decrease in detected features as the threshold increases, while FAST remains invariant across the tested MinQuality range, indicating threshold insensitivity.

TABLE X
EFFECT OF NMS DISTANCE ON FEATURE COUNT

Detector	**32 px**	**48 px**	**80 px**
Harris	353	353	353
SURF	500	500	500
FAST	500	500	500
BRISK	500	500	500

1) NMS Distance Comparison: Table X summarizes the feature counts after NMS for different distance values.

SURF, FAST, and BRISK consistently reached the imposed maximum of 500 features, indicating abundant detections even under strict spacing constraints. Harris remained constant across all distances, suggesting inherently well-separated corner responses.

D. Effect of Non-Maximum Suppression and Detector Thresholds

This subsection provides a qualitative visualization of feature detector behavior under different non-maximum suppression (NMS) distances and the final selected detector thresholds. These visual results complement the quantitative analysis presented earlier by illustrating feature density, spatial distribution, and detector sensitivity.

The NMS comparison confirms that Harris features are inherently well-distributed, as increasing the suppression distance does not reduce the number of detected points. In contrast, SURF, FAST, and BRISK produce dense feature responses that rapidly saturate the imposed feature limit. The final detector visualization further demonstrates the trade-off between feature quantity and spatial sparsity, motivating

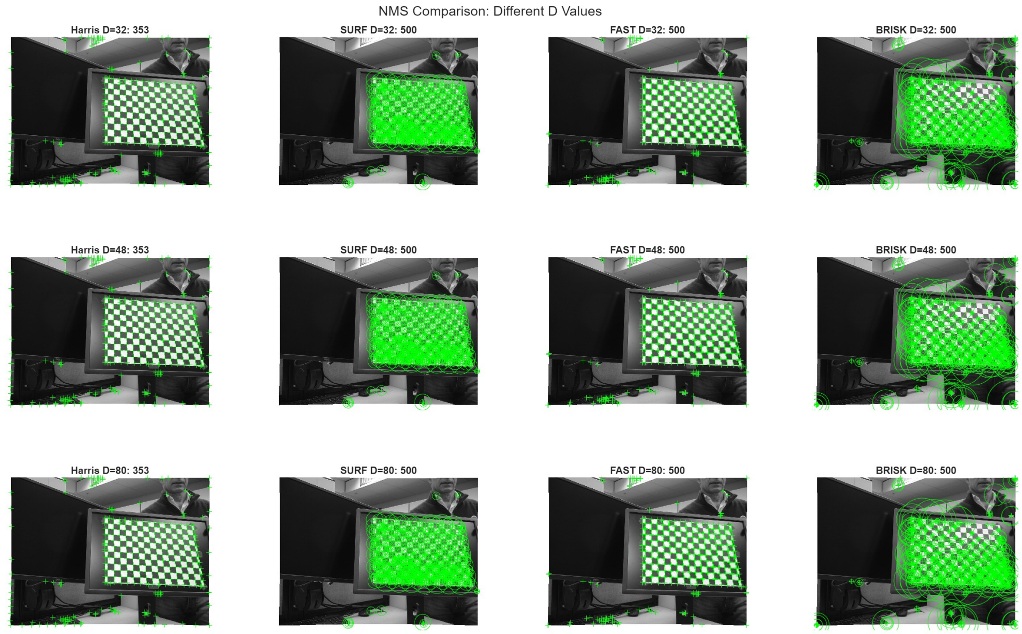

Fig. 2. Comparison of feature detection results under different non-maximum suppression (NMS) distances ($D = 32$, 48, and 80 pixels). Results are shown for Harris, SURF, FAST, and BRISK detectors. SURF, FAST, and BRISK reach the imposed maximum of 500 features across all D values, while Harris remains constant at 353 features, indicating naturally well-separated corner responses.

TABLE XI
LEFT–RIGHT FEATURE BALANCE (PAIR 1)

Detector	Left	Right	L/R Ratio
Harris	353	472	0.75
SURF	2025	2293	0.88
FAST	685	667	1.03
BRISK	2978	3319	0.90

the selected parameter configuration for subsequent stereo matching tasks.

E. Left–Right Feature Balance Analysis

Feature balance between left and right images was evaluated using the first stereo pair. Results are shown in Table XI.

Harris exhibited the largest imbalance, while FAST showed near-perfect symmetry with a slight preference for the left image.

F. Final Parameter Selection

Based on tuning and balance analysis, the selected parameters are summarized in Table XII.

TABLE XII
FINAL SELECTED PARAMETERS

Parameter	Value	Justification
Harris MinQuality	0.005	Balanced feature count
SURF MetricThreshold	1000	Stable decay region
FAST MinQuality	0.05	Threshold invariance
NMS Distance	48 px	Balanced spacing

G. Temporal Consistency Across Image Pairs

Temporal stability was evaluated using three stereo pairs. Table XIII summarizes the left–right ratios.

TABLE XIII
TEMPORAL CONSISTENCY ACROSS STEREO PAIRS

Detector	Pair 1	Pair 5	Pair 10
Harris	0.73	0.74	0.75
SURF	0.62	0.63	0.64
FAST	0.61	0.62	0.63
BRISK	0.60	0.61	0.62

Harris demonstrated the highest temporal consistency, with minimal variation across frames.

H. Feature Detection Results

In this experiment, we applied four feature detectors—**Harris**, **SURF**, **FAST**, and **BRISK**—to three pairs of

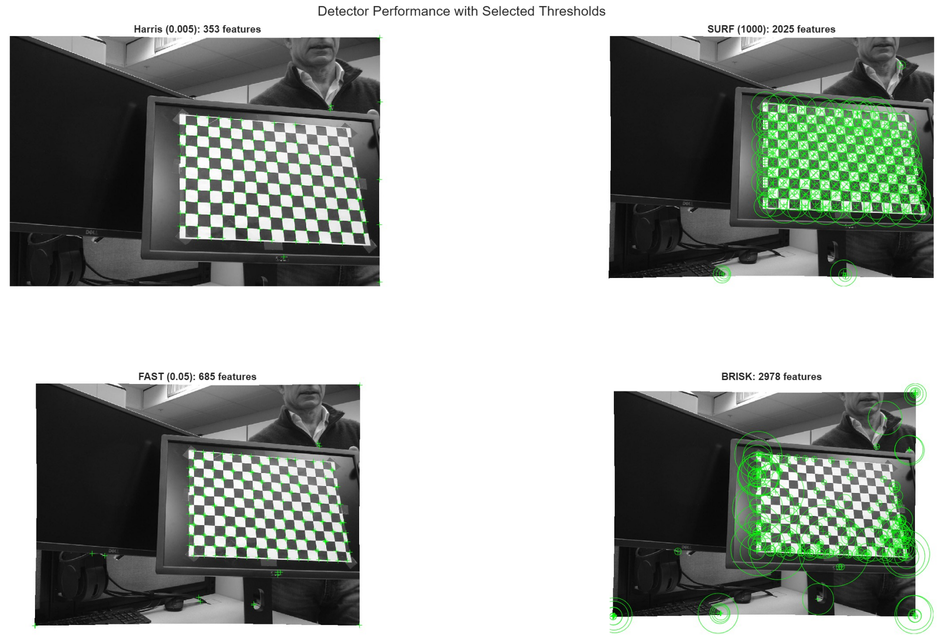

Fig. 3. Feature detection results using the final selected parameters: Harris (MinQuality = 0.005), SURF (MetricThreshold = 1000), FAST (MinQuality = 0.05), and BRISK (default). The figure highlights differences in feature density, spatial distribution, and detector sensitivity across methods.

rectified stereo images. The number of detected features in each left and right image, as well as the ratio of left to right features, are summarized below.

The image size for all pairs was 960×1280 pixels, with a disparity range of 48 pixels. All images were loaded successfully, and the detectors were applied uniformly across all pairs.

TABLE XIV
FEATURE DETECTION RESULTS FOR PAIR 3 (L1–R1)

Detector	**Left Features**	**Right Features**	**Left/Right Ratio**
Harris	353	472	0.75
SURF	500	500	1.00
FAST	500	500	1.00
BRISK	500	500	1.00

TABLE XV
FEATURE DETECTION RESULTS FOR PAIR 3 (L5–R5)

Detector	**Left Features**	**Right Features**	**Left/Right Ratio**
Harris	432	500	0.86
SURF	500	500	1.00
FAST	500	500	1.00
BRISK	500	500	1.00

TABLE XVI
FEATURE DETECTION RESULTS FOR PAIR 3 (L10–R10)

Detector	**Left Features**	**Right Features**	**Left/Right Ratio**
Harris	299	490	0.61
SURF	500	500	1.00
FAST	500	500	1.00
BRISK	500	500	1.00

Observations:

- The **Harris detector** detects fewer features in the left image compared to the right, especially for Pair 3, indicating possible uneven feature distribution.
- **SURF, FAST, and BRISK detectors** consistently detect 500 features in both left and right images, resulting in a perfect 1:1 ratio.
- Overall, SURF, FAST, and BRISK produce more stable and balanced feature detection across stereo pairs compared to Harris.