

Improved ORB Algorithm Through Feature Point Optimization and Gaussian Pyramid

Rohmat Indra Borman¹, Agus Harjoko², Wahyono^{3*}

Faculty of Engineering and Computer Science, Universitas Teknokrat Indonesia, Lampung, Indonesia¹
Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia^{1,2,3}

Abstract—Feature points obtained using traditional ORB methods often exhibit redundancy, uneven distribution, and lack scale invariance. This study enhances the traditional ORB algorithm by presenting an optimal technique for extracting feature points, thereby overcoming these challenges. Initially, the image is partitioned into several areas. The determination of the quantity of feature points to be extracted from each region takes into account both the overall number of feature points and the number of divisions that the image undergoes. This method tackles concerns related to the overlap and redundancy of feature points in the extraction process. To counteract the non-scale invariance issue in feature points obtained via the ORB method, a Gaussian pyramid is employed, and feature points are extracted at each level. Experimental findings demonstrate that our method successfully extracts feature points with greater uniformity and rationality, while preserving image matching accuracy. Specifically, our technique outperforms the traditional ORB algorithm by approximately 4% and the SURF algorithm by 2% in terms of matching performance. Additionally, the processing time of our proposed algorithm is three times faster than that of the SURF algorithm and twelve times faster than the SIFT algorithm.

Keywords—Feature point; Gaussian pyramid; image matching; ORB algorithm; scale invariance

I. INTRODUCTION

In the realms of image processing and pattern recognition, algorithms focused on local feature-based image matching are utilized for identifying specific objects or patterns in images. Local features are useful for identifying characteristics or patterns that exist in small parts of the image [1]. These algorithms target local features in an image, such as edges, corners, or textures, instead of analyzing the entire image [2]. Prominent methods for local descriptor-based feature extraction include the Scale-Invariant Feature Transform (SIFT) [3], Speeded-Up Robust Features (SURF) [4], and Oriented FAST and Rotated BRIEF (ORB) [5]. Compared with other algorithms, ORB has the advantage of faster computing speed than SURF and SIFT and can meet real-time needs [6].

The ORB algorithm, particularly effective in various applications like object positioning, facial recognition, and robot navigation, combines "FAST" (Features from Accelerated Segment Test) for key feature detection and "BRIEF" (Binary Robust Independent Elementary Features) for feature description [7]. It leverages image intensity-based detection for rapid key feature identification [8] and produces compact and affine-resistant binary vector feature descriptions [9]. The ORB algorithm uses a binary representation (BRIEF)

for its feature descriptors, so this approach is much more memory-efficient [10]. However, despite its advantages over traditional methods like Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) in terms of computational efficiency, ORB encounters limitations in scale and rotation invariance, which can result in mismatches during the feature matching process when there are significant changes in object size or orientation.

These challenges are inherent in the design of ORB, which, despite its speed and efficiency, still faces issues with feature matching incompatibility and the robustness needed to effectively handle scale and rotation variations. Previous research has improved the ORB algorithm to address these issues. One study suggests combining ORB with the Lucas–Kanade (LK) Optical Flow algorithm to mitigate mismatches and improve feature matching accuracy [11]. Further research proposes enhancing ORB with an improved quadtree-based uniform distribution to address uneven feature distribution and increase feature extraction calculation efficiency simultaneously [12]. Another improvement utilizes the grayscale centroid method for rotation invariance [13]. Additionally, a combination of ORB's scale invariance advantages with SURF and reducing high-frequency noise impact through NSCT (Nonsubsampled Contourlet Transform) aims to overcome ORB's scale invariance limitations, enhancing matching accuracy and speed by accounting for scale and rotation changes [14]. Furthermore, there are studies that improve the robustness of the ORB algorithm by building pyramid scales and using improved FREAK descriptors to improve scale invariance [15].

Despite various enhancements to the ORB algorithm for handling mismatches, scale, and rotation invariance, the limitations of these advanced algorithms persist. This paper introduces an innovative method that optimizes feature point extraction and employs Gaussian pyramids to bolster scale invariance and minimize feature point redundancy. Gaussian pyramiding, which generates progressively lower-resolution images from the original, aims to enhance ORB's scale variation adaptability by facilitating multi-scale feature analysis. Optimization of feature point extraction refines key point selection, ensuring only the most pertinent and distinct features are utilized for matching, thereby enhancing accuracy and reducing computational demands. The proposed approach not only seeks to address ORB's specific shortcomings but also advances feature matching algorithms by offering a robust, scalable solution. A comprehensive comparison with current techniques underscores the proposed method's unique benefits,

*Corresponding author.

laying the foundation for its application in diverse real-world contexts.

II. RELATED WORKS

The prominence of ORB is attributed to its proficiency in achieving high-speed performance and its applicability in real-time processing contexts [16]. Nevertheless, the robustness of ORB and its adaptability under diverse imaging conditions remain areas of concern [17]. Efforts by scholars to refine the ORB algorithm have aimed at mitigating its inherent constraints and deficiencies. Subsequent investigations have enhanced ORB, including an integration with the Lucas–Kanade Optical Flow (LK) technique to diminish mismatches and augment the precision of feature matching [11]. Additional studies have introduced improvements such as a quadtree-based uniform distribution enhancement for ORB, aimed at rectifying the issue of uneven distribution of features while simultaneously boosting the efficiency of feature extraction [12]. Moreover, the adaptation involving the grayscale scale-invariant centroid technique seeks to address rotation invariance [13]. The incorporation of ORB's scale invariance features with those of SURF, coupled with the mitigation of high-frequency noise via NSCT (Nonsubsampled Contourlet Transform), targets the amelioration of ORB's scale invariance challenges, thereby improving both accuracy and the speed of matching by accounting for variations in scale and rotation [14]. Additionally, enhancing the robustness of the ORB algorithm through the construction of a pyramid scale and the application of an advanced FREAK descriptor has been proposed to bolster scale invariance [15].

However, the advancements in the ORB algorithm still encounter challenges. Specifically, the application of the Hamming distance for matching feature points continues to result in mismatches and a decrease in matching precision, especially when the source and target images exhibit numerous analogous regions [18]. Moreover, the primary ORB feature detection algorithm struggles with issues such as uneven distribution of feature points, a high rate of feature mismatches, and limited robustness [19]. Attempts to refine the ORB matching algorithm through adaptive thresholding have been directed at solving problems related to the extraction of background pixels as feature points and the incorrect matching of feature points in environments with complex backgrounds, underscoring the original ORB algorithm's reduced robustness in intricate scenarios [20].

Despite these enhancements aimed at improving the ORB algorithm's performance in overcoming mismatches and the traditional robustness issues of ORB feature matching, the necessity of evaluating the potential limitations and compromises of these proposed solutions remains critical. The efficacy of these algorithms in complex settings and their computational efficiency warrant comprehensive assessment, particularly regarding improvements in matching capabilities and the ability to navigate scale and rotation variations.

III. FEATURE POINTS EXTRACTION IN THE ORB

A. Detection of Feature Points

Image feature points refer to the more crucial points within an image, such as contour points, bright spots within darker

regions, and dark points within lighter areas [21]. In ORB, FAST (Features from Accelerated Segment Test) is employed for the identification of these feature points. Fig. 1 shows the feature point extraction process in the FAST approach.

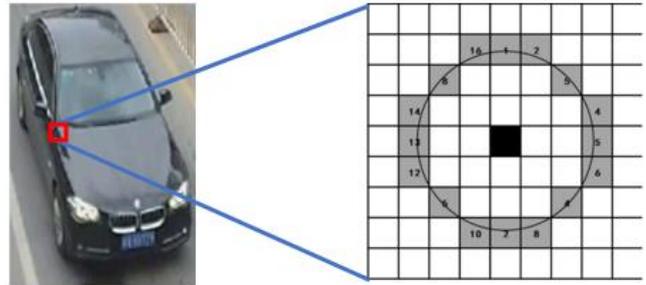


Fig. 1. Illustration of the schematic diagram of FAST feature point extraction.

The fundamental principle underlying FAST entails identifying salient points by comparing a given point with its neighboring points. If the point substantially deviates from the majority of its surrounding points, it is designated as a feature point [15]. The procedure for identifying feature points in an image involves several steps, as outlined below:

- 1) A pixel point P is selected from Fig. 1 to assess its potential as a feature point. Assuming its initial grayscale value.
- 2) Set a suitable threshold t (e.g., 20%). Points are deemed distinct if the absolute disparity in gray scale values between them above the threshold, t .
- 3) Choose 16 points in a circle with a radius of 3, centered at point P .
- 4) Point P is identified as a corner point if among the 16 surrounding points, there exist n consecutive points with grayscale values significantly higher or lower than that of P . Typically, the value of n is set to 12.
- 5) To enhance the efficiency of feature point detection, a predictive operation can be implemented. This method efficiently eliminates the majority of points that are not corner points. This is achieved by analyzing the gray degree values of the points located at locations 1, 5, 9, and 13 on the circumference of the circle P . First, check if points 1 and 9 are similar to P , and if so, examine points 5 and 13. P is considered a corner point only if at least three of these four points are all greater or less than P . If these requirements are not satisfied, P is not considered a corner point and is immediately eliminated.

Through this process, the points obtained in the environment around P become three categories, as in Eq. (1).

$$S_P \rightarrow x = \begin{cases} a, I_P \rightarrow i \leq I_P - t \\ b, I_P - t < I_P \rightarrow i < I_P + t \\ c, I_P \rightarrow i \geq I_P + t \end{cases} \quad (1)$$

where, P refers to the gray scale value of point I at 16 points on the circle, a refers to the point whose darkness value is more than P , b refers to the point that is similar to P , and c indicates the point that is brighter than P .

B. Calculating the Feature Point Descriptor

The ORB algorithm employs an enhanced version of BRIEF to compute descriptors for feature points. In this approach, N pairs are strategically selected around P feature points. The results of comparing these N pairs of points are then combined to create a descriptor. This process of descriptor compilation is depicted in Fig. 2.

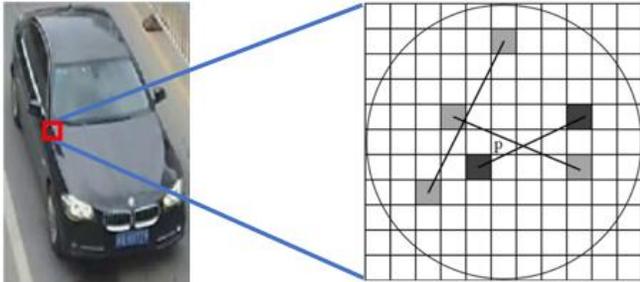


Fig. 2. Illustration of the descriptor calculation schematic diagram.

The steps involved in calculating the descriptor are as follows:

1) Construct a circle, designated as O , centered around the point P , with d representing the radius of the circle.

2) On circle O , select N pairs of points, where for explanatory purposes, N is set to 4. Refer to Fig. 2 for a visual representation. Label these four pairs of selected points as $P_1(A, B)$, $P_2(C, D)$, $P_3(E, F)$, and $P_4(G, H)$. It's important to note that each pair of points (A, B) , (C, D) , (E, F) , and (G, H) is strategically chosen on the circumference of the circle to facilitate the calculation of the feature point descriptors as per the ORB algorithm's methodology.

3) Specify the function T in Eq. (2).

$$T(P(A, B)) = \begin{cases} 1 & I_A > I_B \\ 1 & I_A \leq I_B \end{cases} \quad (2)$$

where, A and B are the respective gray scale values.

4) For each of the four selected pairs of points, apply operation T . The results of this operation on each pair are then combined. As an example,

$$\begin{aligned} T(P_1(A, B)) &= 1, T(P_1(A, B)) = 0 \\ T(P_3(A, B)) &= 1, T(P_4(A, B)) = 1 \end{aligned}$$

The resulting descriptor obtained is 1011.

As FAST lacks the ability to determine the orientation information of feature points, ORB addresses this limitation by employing image moments to ascertain the direction of these points. This is accomplished by computing the centroid of the grayscale image in the vicinity of the feature point. The image moment within a small image block B is characterized according to Eq. (3).

$$m_{pq} = \sum x^p y^q I(x, y) \quad (3)$$

where, m_{pq} refers to the gray scale value of the point (x, y) . The center of mass, denoted as C , can be calculated using Eq. (4).

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (4)$$

where, m_{00} represents the zero-order moment, while m_{10} and m_{01} denote the first-order moments.

Next, connect the geometric center O of the image with the center of mass C to calculate the orientation of the vector OC . Simultaneously, if the x and y coordinates fall within the range of $[-r, r]$, where r represents the neighborhood radius of the feature point, and considering the feature point as the coordinate origin, the directional angle of the feature point can be determined using Eq. (5).

$$\theta = \arctan \left(\frac{m_{01}}{m_{10}} \right) \quad (5)$$

IV. THE PROPOSED METHOD

The ORB utilizes the FAST algorithm for detecting image feature points, yet this often results in a dense and redundant distribution of these points. Generally, a higher count of feature points correlates with more precise image matching. Nevertheless, a dense accumulation of feature points can be detrimental to subsequent feature description, potentially compromising the accuracy of image matching [22], [23]. This paper introduces an enhanced ORB algorithm, featuring an optimized method for extracting feature points. This enhanced methodology includes segmenting the image into distinct regions and selectively extracting feature points from each region, thereby addressing the aforementioned issue. Additionally, it employs an adaptive threshold, halting feature extraction once a predetermined number of feature points is achieved, thereby reducing extraction time.

The ORB algorithm for feature extraction begins with a pyramid-scale transformation of the image. The scale pyramid used in FAST will be optimized by adopting a Gaussian SIFT pyramid to overcome the scale invariance problem. An image pyramid is a multiscale representation of a single image, comprising a sequence of images that are various versions of the original image at different resolutions. To be strong against food scale invariants, the difference-of-Gaussian function is used. In this function, convolution operations will be obtained on the input image with a difference-of-Gaussian filter. The Difference-of-Gaussian (DoG) is the dissimilarity between images that have been blurred with Gaussian filters at different scale values represented by the parameter 'k.' The convolution-resulting images are grouped by octave; the k value is set at the beginning so that the same number of blurred images is obtained in each octave and the same DoG image is obtained for each octave. After obtaining the DoG image for each octave, the next step is to look for key point candidates. The procedural flow of this enhanced ORB algorithm is illustrated in Fig. 3.

Subsequent to the initial steps, the FAST detector is employed to identify corner points within the image. FAST operates by comparing a pixel, say p , with the 16 surrounding pixels that constitute a circle. These circumjacent pixels are classified into three categories based on their brightness relative to p : brighter, darker, or equal in intensity. p is designated as a keypoint if there are more than 8 pixels in the circle that are either significantly darker or lighter than p . While the use of FAST for keypoint detection yields feature

points that are densely packed and redundant, it is generally understood that an increased quantity of feature points enhances the accuracy of image matching. However, the dense clustering of these points poses challenges for subsequent feature description and may adversely impact the precision of image matching [24].

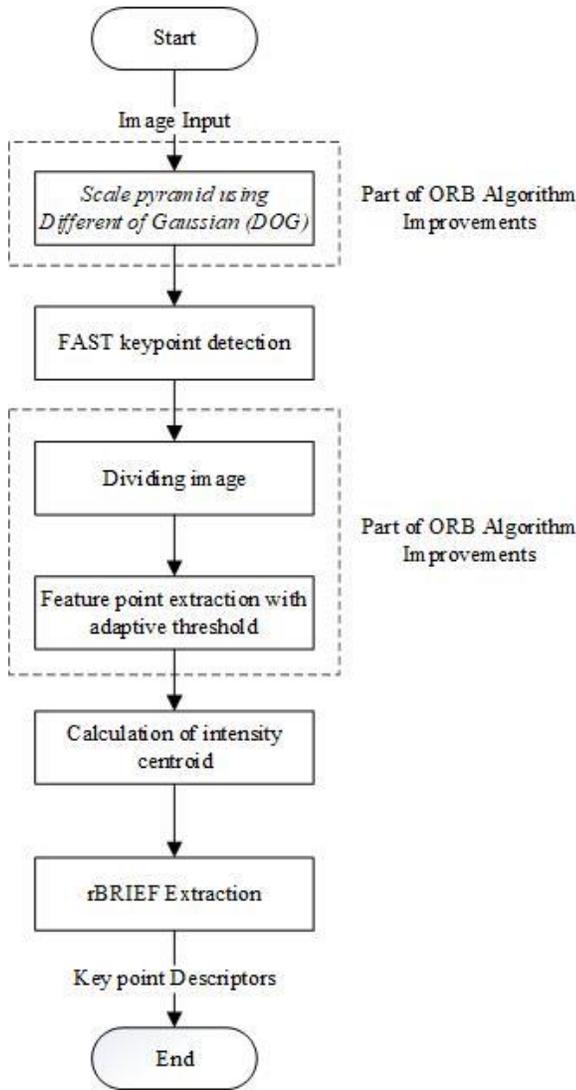


Fig. 3. Proposed algorithm workflow.

To mitigate this issue, the image is initially segmented into several regions. This segmentation is guided by the total number of feature points being searched for and the number of regions to be divided, then the number of feature points is extracted for each region. The image is initially evenly partitioned into $M \times N$ regions of identical size, with the feature points being randomly distributed within these regions. These regions are then organized based on the first and last columns, as depicted in Eq. (6).

$$j = \frac{FP_{required}}{M \times N} \quad (6)$$

where, $FP_{required}$ represents the number of feature points used, M indicates the number of rows separated, and N indicates the number of columns separated.

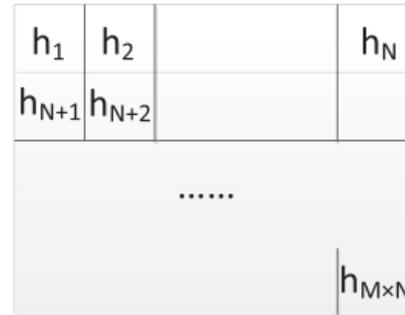


Fig. 4. Displays the sorting of divided regions based on the first and last columns.

Regions are arranged based on the first and last columns, as illustrated in Fig. 4. Within each segmented region, the FAST algorithm is employed for feature point detection. During this process, a threshold value, denoted as t , is utilized. The number of feature points obtained in each section is then evaluated based on the specified number j . If the count of detected feature points is less than j , the threshold t is decreased, followed by a re-detection. A pair of points is deemed distinct only if the absolute difference in their grayscale values exceeds the threshold t , allowing for the continuation of the detection process. Consequently, by lowering the value of t , a larger number of corner points can be identified, thereby enhancing the scope for subsequent filtering. This adjustment in threshold levels facilitates a more comprehensive and effective detection of feature points within each region.

In cases where the number of detected feature points is not fewer than j , it becomes necessary to select j optimal feature points from the pool. For this selection process, the non-maximum suppression method is utilized. Consider two adjacent points, P and Q , in this context. The method involves sequentially calculating the sum of differences between each of these points and their respective 16 surrounding points. Subsequently, the point exhibiting the fewest disparities is eliminated. This elimination process continues until the number of remaining points matches the desired number, j . The points that remain after this procedure are considered the optimal points. The formula for calculating the sum of differences is detailed in equation (7).

$$V = \max(\sum_{x \in S_{bright}} |I_P \rightarrow x - I_P| - t, \sum_{x \in S_{dark}} |I_P - I_P \rightarrow x| - t) \quad (7)$$

At this juncture, the BRIEF algorithm is employed to process the results obtained from the previous stage. Given that BRIEF lacks the capacity to accommodate rotational variations, the rBRIEF variant is utilized, wherein BRIEF is oriented in alignment with the keypoint. The ensuing phase involves an analysis of all the sampling pairs, comparing the first pixel with the second pixel in each pair. In this comparison, if the first pixel is brighter than the second, it is assigned a value of 1; otherwise, it receives a value of 0. This

binary valuation process is iterated until 256 pairs are evaluated. The ORB algorithm, through this procedure, generates 32-dimensional descriptors. These descriptors are derived from the 256-bit pairs, which are further segmented into bytes for computational efficiency and clarity.

V. EXPERIMENTAL RESULTS

To assess the efficacy of the enhanced ORB algorithm, this study conducts a comparative analysis among the conventional SIFT, SURF, and ORB algorithms and the improved ORB algorithm. The experimental setup utilizes Jupiter Notebook and OpenCV as the primary tools. The dataset employed for the experiments is the Cities Transportation dataset, which offers a diverse range of imagery. The evaluation is quantitative, covering several key aspects: the performance of feature point extraction, the efficacy of image matching, and the algorithms' resilience to rotational and scale variations. This comparison aims to evaluate the performance in terms of accuracy and the time required for matching. The results, which include both the accuracy rates and the matching durations, are meticulously recorded. These collected data points offer a comprehensive picture of how the improved ORB algorithm compares with other algorithms.

The experiment began by analyzing the extraction of feature points using the SIFT, SURF, ORB, and enhanced ORB algorithms, during which the number of feature points and the time needed for their detection were calculated. The outcomes of this evaluation, including both the quantity of feature points and the time taken for their detection across each algorithm, are methodically outlined in Table I.

TABLE I. COMPARISONS IN FEATURE POINT EXTRACTION PERFORMANCE

| Algorithm | Feature Points | Detection Time (ms) |
|--------------|----------------|---------------------|
| SIFT | 3024 | 2334.4 |
| SURF | 1567 | 1804.7 |
| ORB | 924 | 79.465 |
| Our Proposed | 1089 | 80.734 |

As indicated in Table I, the enhanced ORB algorithm in the paper demonstrates a superior performance in terms of the number of detected feature points compared to the traditional ORB algorithm. However, it is essential to note that an excessive number of feature points can lead to effective information redundancy and increased computational complexity. Regarding detection time, the improved ORB algorithm exhibits significantly shorter processing times compared to the SIFT and SURF algorithms. In general, the improved ORB algorithm proves effective in rapidly detecting image information and emphasizing image details, showcasing its efficiency in feature point detection and the validity of feature point selection. To validate the matching performance of the proposed algorithm, image matching was conducted using the SIFT, SURF, ORB, and improved ORB algorithms, as illustrated in Fig. 5.

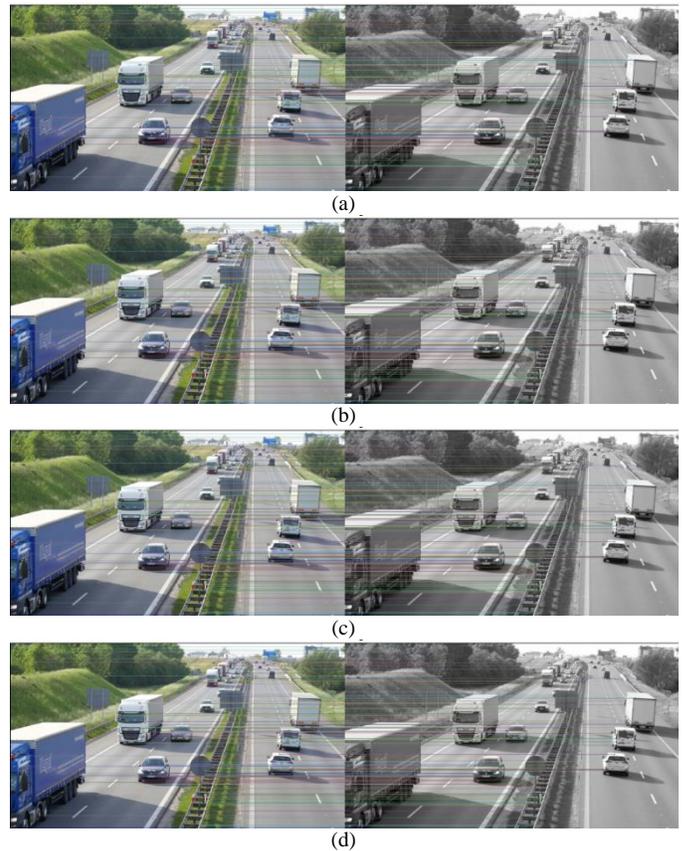


Fig. 5. Matching results (a) SIFT, (b) SURF, (c) ORB, (d) Proposed model.

The accuracy and matching time are analyzed as shown in Table II.

TABLE II. COMPARISONS IN MATCHING PERFORMANCE

| Algorithm | Matching Accuracy (%) | Match Time (ms) |
|--------------|-----------------------|-----------------|
| SIFT | 97.67 | 12,539.74 |
| SURF | 90.90 | 3332.43 |
| ORB | 88.48 | 987.00 |
| Our Proposed | 92.90 | 975.02 |

The algorithm presented in this paper does not attain a faster matching speed compared to the conventional ORB algorithm. Nonetheless, it exhibits superior matching accuracy over the traditional SURF and ORB algorithms. As indicated in Table II, the enhanced ORB algorithm notably decreases the matching time relative to the other three algorithms. Specifically, with regard to matching time, the enhanced ORB algorithm demonstrates better performance than both the SIFT and ORB algorithms.

The algorithm is then tested with scale invariance by randomly enlarging or reducing the image and then matching. The scale invariance test sample can be seen in Fig. 6.

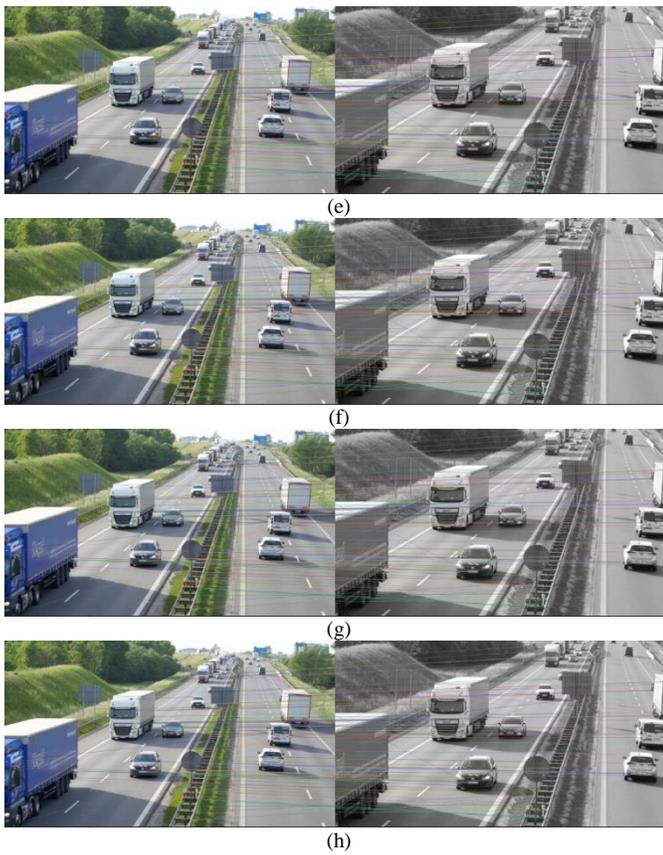


Fig. 6. Matching results for scaled images (e) SIFT, (f) SURF, (g) ORB, (h) Proposed model.

Image matching across various scales tests the algorithm's ability to overcome scale invariance. The results of matching accuracy and matching time are presented in Table III.

TABLE III. COMPARISON OF MATCHING PERFORMANCE FOR SCALE VARIATIONS

| Algorithm | Matching Accuracy (%) | Match Time (ms) |
|--------------|-----------------------|-----------------|
| SIFT | 97.97 | 11,892.61 |
| SURF | 84.84 | 3520.58 |
| ORB | 76.96 | 1774.61 |
| Our Proposed | 87.54 | 1664.79 |

To validate the algorithm's capability in terms of rotation invariance, a series of image rotation procedures and corresponding matching experiments were conducted. Specifically, to examine the algorithm's resilience to rotation, each image in the experimental sample was subjected to a range of rotations, spanning from 0° to 180° at intervals of 30° . This methodical approach ensures a thorough evaluation of the algorithm's performance under various rotational transformations. An illustrative example of this rotation invariance test experiment is presented in Fig. 7, providing a visual demonstration of the algorithm's effectiveness in maintaining accurate matching despite the rotational alterations of the images.

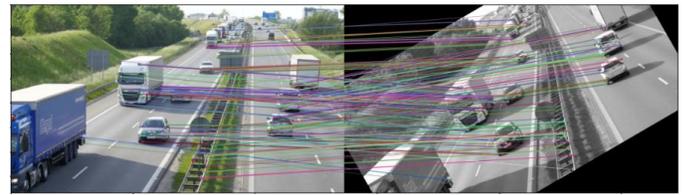


Fig. 7. Matching results for rotated images.

An algorithm that exhibits good resilience is one that can handle rotational variations with high matching accuracy and short matching times. To view the comparative results of the algorithms tested against rotational variations (see Table IV).

TABLE IV. COMPARISON OF MATCHING PERFORMANCE FOR VARIATIONS OF ROTATION

| Algorithm | Matching Accuracy (%) | Match Time (ms) |
|--------------|-----------------------|-----------------|
| SIFT | 97.67 | 12,539.74 |
| SURF | 90.90 | 3332.43 |
| ORB | 88.48 | 987.00 |
| Our Proposed | 92.90 | 975.02 |

As shown in Table III and Table IV, SIFT, SURF, ORB, and the algorithms proposed in this paper exhibit commendable performance in handling image rotation and scale changes. However, these algorithms demonstrate clear advantages in terms of running time. Notably, the ORB algorithm lacks scale invariance, resulting in faster execution but inferior matching performance. Taking into account both matching effectiveness and time efficiency, the proposed algorithm represents an improvement over the ORB algorithm. It preserves the advantages and accuracy of the ORB algorithm while addressing its deficiency in scale invariance, leading to better overall results.

Based on the experiments conducted, the enhanced ORB algorithm demonstrates strong matching capabilities, as the feature points extracted by this algorithm exhibit uniformity without compromising image matching accuracy. The results of feature point extraction highlight the representativeness of the points extracted by our proposed algorithm, thereby contributing to more accurate and stable image matching. Furthermore, considering runtime is an essential criterion for evaluating algorithm superiority. When comparing the outcomes of image matching in the same test, a shorter processing time indicates a more efficient method. Conversely, if the running times are similar, superior matching results signify a better algorithm. The overall matching accuracy results across various tests are presented in the graph in Fig. 8.

According to the graph in Fig. 8, the SIFT algorithm has the highest matching accuracy. These results are consistent with previous research, which states that SIFT can produce high matching accuracy against image features invariant to scale and rotation, enabling it to find consistent matches, though the required matching time is very significant [17]. This becomes a problem when applied to cases requiring real-time matching capabilities. Nonetheless, the proposed algorithm outperforms the traditional ORB algorithm by approximately 4% better and exceeds the SURF algorithm by 2% better in

terms of matching performance. The use of feature point optimization by dividing randomly distributed feature points improves matching capabilities. Additionally, the use of a Gaussian pyramid enhances the proposed algorithm's ability to handle scale invariance. This is in line with previous research that employs pyramid scale construction on images to improve scale invariance [15]. For an overall comparison of matching times across various tests, the experimental results are presented in the graph in Fig. 9.

is adjusted adaptively to enhance matching accuracy results in an increase in computational processes, thereby slightly reducing speed. This is consistent with research that uses a truncated adaptive threshold in the ORB algorithm to address uneven feature distribution, which can improve accuracy but reduce computational speed [13]. However, overall, from several test parameters, the proposed algorithm has a matching time that is three times faster than the SURF algorithm and twelve times faster than the SIFT algorithm.

Fig. 9 shows that the ORB algorithm has the best average matching time. The use of an adaptive threshold that varies or

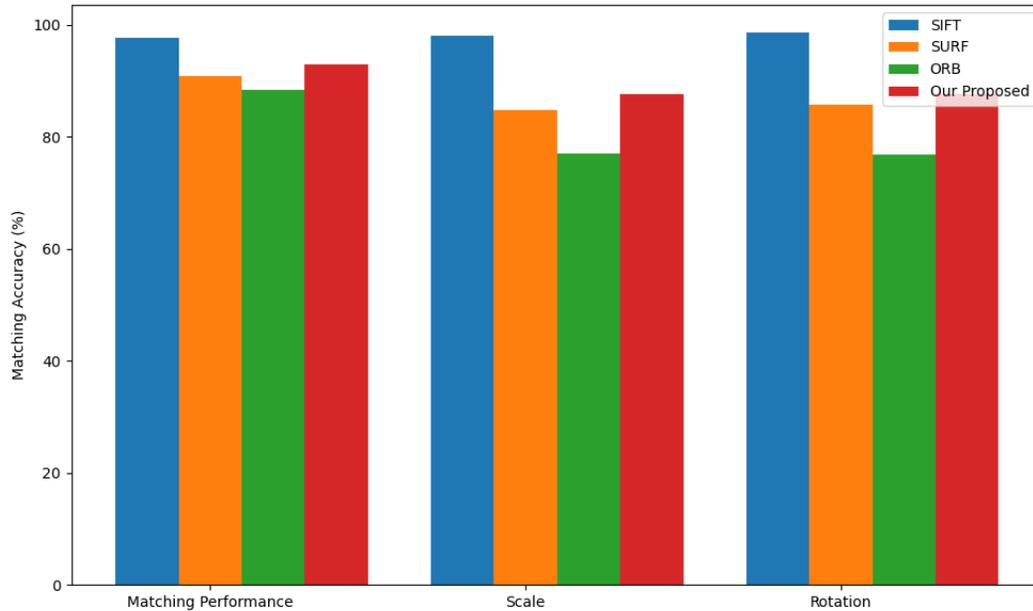


Fig. 8. Matching accuracy graph for all test parameters.

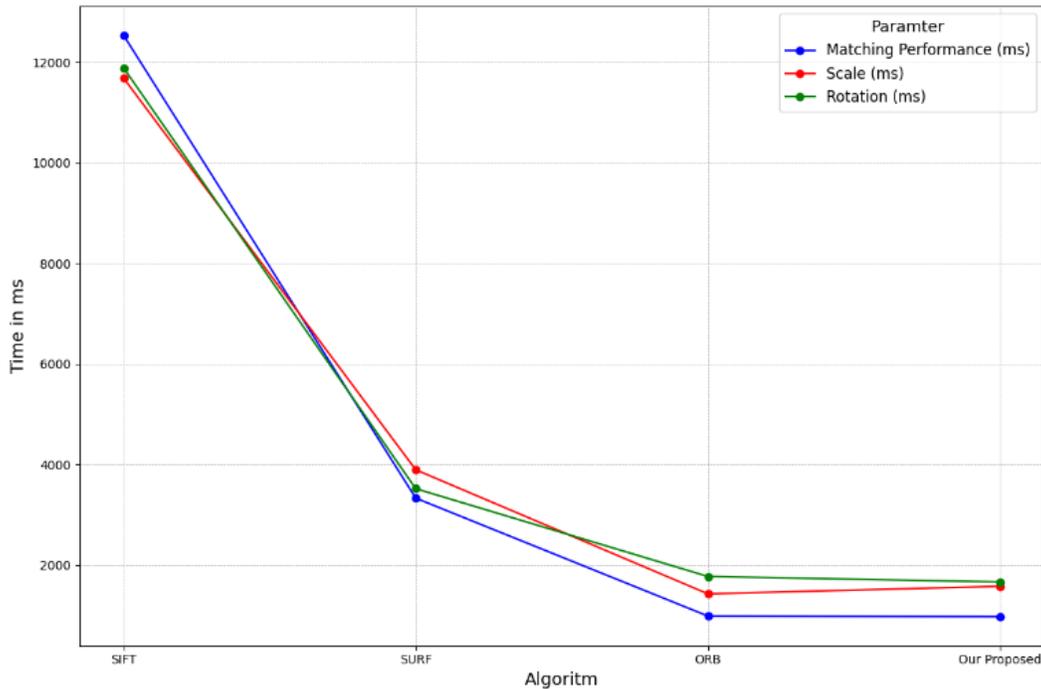


Fig. 9. Matching time graph for all test parameters.

VI. CONCLUSION

This paper presents an enhanced ORB algorithm that focuses on optimized extraction of feature points, aiming to resolve the issues of overlapping feature points and the lack of scale invariance found in traditional ORB methods. The algorithm strategically segments images into distinct regions, thereby ensuring that the feature points extracted are optimally effective within each specific region. By setting a threshold tailored to the extraction of feature points from individual regions, the algorithm not only accelerates the extraction process but also achieves a more even distribution of feature points, which consequently enhances the speed of matching. Future research might include an in-depth evaluation of this improved ORB algorithm's performance across diverse photographic conditions, including varying lighting environments. Moreover, there is scope for further refinement of this algorithm to extend its capabilities to object detection and tracking within video content.

ACKNOWLEDGMENT

This work was partially funded by the Department of Computer Science and Electronics, Universitas Gadjah Mada under the Publication Funding Year 2024.

REFERENCES

- [1] M. E. Wibowo, A. Ashari, A. Subianto, and W. Wahyono, "Human Face Detection and Tracking Using RetinaFace Network for Surveillance Systems," in IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society, 2021, pp. 1–5. doi: 10.1109/IECON48115.2021.9589577.
- [2] G. Ding, P. Zhao, T. Li, H. Zhao, and T. Lou, "An Image Feature Matching Algorithm with Clustering Constraints," in International Conference on Machine Vision, Automatic Identification and Detection (MVAID), 2023. doi: 10.1088/1742-6596/2577/1/012006.
- [3] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004, doi: 10.1023/B:VISI.0000029664.99615.94.
- [4] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *Computer Vision – ECCV*, 2006, pp. 404–417.
- [5] E. Rublee, W. Garage, and M. Park, "ORB: an efficient alternative to SIFT or SURF," in *International Conference on Computer Vision*, 2011, pp. 2564–2571.
- [6] S. A. K. Tareen and Z. Saleem, "A Comparative Analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," in *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2018, pp. 1–10. doi: 10.1109/ICOMET.2018.8346440.
- [7] Y. Zhao, Z. Xiong, S. Duan, S. Zhou, and Y. Cui, "Improved ORB Based Image Registration Acceleration Algorithm in Visual-Inertial Navigation System," in *Proceedings - 2020 Chinese Automation Congress (CAC)*, 2020, pp. 5714–5718. doi: 10.1109/CAC51589.2020.9326928.
- [8] C. Li, Y. Jia, H. Wang, C. Rong, and Y. Zhu, "Improved ORB Algorithm Based on Binocular Vision," in *International Conference on Computer and Communications (ICCC)*, 2019, pp. 1739–1743.
- [9] X. Tian, G. Zhou, and M. Xu, "Image copy-move forgery detection algorithm based on ORB and novel similarity metric," *IET Image Process. Res.*, vol. 14, no. 10, pp. 2092–2100, 2020, doi: 10.1049/iet-ivr.2019.1145.
- [10] K. Wu, "Creating Panoramic Images Using ORB Feature Detection and RANSAC-based Image Alignment," *Adv. Comput. Commun.*, vol. 4, no. 4, pp. 220–224, 2023, doi: 10.26855/acc.2023.08.002.
- [11] Q. Chen et al., "Horticultural Image Feature Matching Algorithm Based on Improved ORB and LK Optical Flow," *Remote Sens.*, vol. 14, no. 4465, pp. 1–18, 2022, doi: <https://doi.org/10.3390/rs14184465>.
- [12] J. Yao, P. Zhang, Y. Wang, Z. Luo, and X. Ren, "An Adaptive Uniform Distribution ORB Based on Improved Quadtree," *IEEE Access*, vol. 7, pp. 143471–143478, 2019, doi: 10.1109/ACCESS.2019.2940995.
- [13] Y. Dai and J. Wu, "An Improved ORB Feature Extraction Algorithm Based on Enhanced Image and Truncated Adaptive Threshold," *IEEE Access*, vol. 11, pp. 32073–32081, 2023, doi: 10.1109/ACCESS.2023.3261665.
- [14] D. M. H. Lai and D. Ma, "Remote Sensing Image Matching Based Improved ORB in NSCT Domain," *J. Indian Soc. Remote Sens.*, vol. 47, no. 5, pp. 801–807, 2019, doi: 10.1007/s12524-019-00958-y.
- [15] L. Zhao, J. Yang, Y. Zhang, and J. Huang, "Research on Feature Matching of an Improved ORB Algorithm," in *IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2022, pp. 765–769. doi: 10.1109/ITOEC53115.2022.9734583.
- [16] P. Bansal, J. B. Dinesh, V. R. Shravan Kumar, B. Sujay Krishna, and T. S. Chandar, "Video Stabilization Using ORB Detector," in *International Conference on Computer and Automation Engineering (ICCAE)*, 2022, pp. 50–55. doi: 10.1109/ICCAE55086.2022.9762438.
- [17] A. Kaur, M. Kumar, and M. K. Jindal, "Cattle identification system: a comparative analysis of SIFT, SURF and ORB feature descriptors," *Multimed. Tools Appl.*, vol. 82, no. 18, pp. 27391–27413, 2023, doi: 10.1007/s11042-023-14478-y.
- [18] X. Ji, H. Yang, and C. Han, "Research on image stitching method based on improved ORB and stitching line calculation," *J. Electron. Imaging*, vol. 31, no. 5, p. 51404, Jan. 2022, doi: 10.1117/1.JEL.31.5.051404.
- [19] X. Wang, F. Liu, and Y. Xue, "Visual odometer method based on improved ORB feature," in *International Conference on Electrical, Electronic Information and Communication Engineering (EEICE)*, 2021. doi: 10.1088/1742-6596/1920/1/012110.
- [20] S. Li, Q. Wang, and J. Li, "Improved ORB matching algorithm based on adaptive threshold," in *International Symposium on Advances in Electrical, Electronics and Computer Engineering (ISAEECE)*, 2021. doi: 10.1088/1742-6596/1871/1/012151.
- [21] Y. Xie, Q. Wang, Y. Chang, and X. Zhang, "Fast Target Recognition Based on Improved ORB Feature," *Appl. Sci.*, vol. 12, no. 786, pp. 1–14, 2022.
- [22] W. Chen, Y. Zhang, J. Wen, K. Li, and G. Yang, "An Application of Improved RANSAC Algorithm in Visual Positioning," in *International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2019, no. Itaic, pp. 1358–1362.
- [23] C. Yao, H. Zhang, J. Zhu, D. Fan, Y. Fang, and L. Tang, "ORB Feature Matching Algorithm Based on Multi-Scale Feature Description Fusion and Feature Point Mapping Error Correction," *IEEE Access*, vol. 11, pp. 63808–63820, 2023, doi: 10.1109/ACCESS.2023.3288594.
- [24] H. Sun, P. Wang, D. Zhang, C. Ni, and H. Zhang, "An Improved ORB Algorithm Based on Optimized Feature Point Extraction," in *2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering, AUTEEE 2020*, 2020, pp. 389–394. doi: 10.1109/AUTEEE50969.2020.9315683.