IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 5 March 2023, accepted 20 March 2023, date of publication 24 March 2023, date of current version 3 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3261665

RESEARCH ARTICLE

An Improved ORB Feature Extraction Algorithm Based on Enhanced Image and Truncated Adaptive Threshold

YONG DAI AND JIAXIN WU
School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang 110159, China
Corresponding authors: Yong Dai (578916135@qq.com) and Jiaxin Wu (848434225@qq.com)

This work was supported by the Scientific Research Fund of Liaoning Provincial Education Department under Grant LJKMZ20220616.

ABSTRACT The ORB (Oriented fast and rotated brief, ORB) algorithm is limited by its inability to extract feature points or extract only a small number of feature points when a fixed threshold is used in complex lighting conditions. To address this issue, this paper proposes a method that combines image enhancement with truncated adaptive threshold to improve the ORB feature extraction algorithm. Firstly, the original image is converted to grayscale. Secondly, the image is enhanced by applying Gaussian filtering for noise reduction, truncated adaptive gamma brightness adjustment, and unsharp masking operation. Then, the enhanced image is segmented into subregions of a specified size, and an improved truncated OTUS method is employed to calculate the adaptive threshold for each subregion. Finally, ORB feature points are extracted by utilizing the adaptive threshold. Experimental results show that the improved ORB algorithm can significantly increase the number of feature points in complex lighting conditions, with good accuracy, real-time performance, and robustness.

INDEX TERMS Feature extraction, ORB, image enhancement, adaptive threshold, OTUS.

I. INTRODUCTION

ORB (Oriented fast and rotated brief, ORB) is an efficient and pixel-based feature detection algorithm which has extensive applications in diverse fields, including target recognition [1], [2], simultaneous localization and mapping [3], [4], [5], and remote sensing images [6].

As shown in Table 1, there have been many excellent feature extraction algorithms in history. The algorithm of ORB is established upon the the algorithm of FAST (Features from accelerated segment test, FAST), which determines the orientation of feature points by the grayscale centroid method and determines the feature descriptor by BRIEF (Binary robust independent elementary features, BRIEF). In contrast to existing feature extraction algorithms, ORB algorithm gives the many advantages: Firstly, ORB algorithm has only one parameter (threshold), which makes it relatively easy to adjust. Secondly, ORB algorithm has a fast processing speed, making it suitable for real-time processing of large amounts of data. In addition, the accuracy and stability of ORB algorithm have been verified in experiments, and it can effectively detect key points. Overall, ORB algorithm is an efficient, accurate, stable, robust, and widely adaptable feature extraction method suitable for various computer vision tasks.

TABLE 1. Development of feature extraction algorithms.

Year	Algorithm	Authors
1988	Harris [7]	Harris C, Stephens M.
1994	Shi-Tmasi [8]	Shi J.
2004	SIFT [9]	Lowe G.
2006	FAST [10]	Rosten E, Drummond T.
2006	SURF [11]	Bay H, Tuytelaars T.
2011	ORB [12]	Rublee E, Rabaud V.
2011	BRISK [13]	Leutenegger S, Chli M.
2015	CNNFT [14]	Simo-Serra E.
2015	MathNet [15]	Xufeng Han.
2021	GCNv2 [16]	Tang J.

The associate editor coordinating the review of this manuscript and approving it for publication was Jeon Gwanggil.

However, the ORB feature extraction algorithm has significant limitations in the presence of lighting variations, as it uses a fixed threshold to detect whether a pixel is a key point. This means that in situations with significant lighting variations, pixel values in different regions may exceed the threshold, leading to incorrect detection of key points. Moreover, lighting variations can increase noise and interference signals in the image, further degrading the algorithm's performance. Therefore, addressing the problem of lighting variations is one of the key challenges in improving the performance of the ORB feature extraction algorithm.

In recent years, scholars have mainly made two improvements to address the illumination variation problem in the ORB algorithm: the first is to improve the calculation method of adaptive thresholds [17], [18], [19], and the second is to expand the application scope of adaptive thresholds [20], [21].

To address the problem of the unsuitability of the threshold in the ORB algorithm for different image scenes, scholars have proposed adaptive threshold calculation methods based on local pixel distribution, such as methods based on adaptive truncation threshold [22], adaptive median [23] and adaptive statistics [24]. In addition, artificial learning-based methods, such as SVM (Support vector machine, SVM) [25], CNN (Convolutional neural networks, CNN) [26], have also been proposed to predict adaptive thresholds, thereby improving the robustness and adaptability of the ORB algorithm to complex lighting conditions.

However, these methods still have some limitations in practical applications. For example, the calculation of adaptive threshold in the use of the local pixel distribution is prone to interference from noise and outliers, which may lead to false positives or false negatives. Also, the learning-based adaptive threshold methods need a large amount of training data and calculating resources, and have high requirements for data distribution and features, which is not practical in real-world usage.

The present study presents an novel feature extraction algorithm of ORB based on enhanced image and truncated adaptive threshold. Firstly, a Gaussian filter is applied for noise reduction, and then the image is preprocessed using truncated adaptive gamma brightness adjustment and unsharp masking methods to promote the initial image with good quality. Next, the image will be segmented into multiple fixed-sized sub-regions, and the corresponding thresholds are obtained using an improved truncated adaptive OTUS method in each sub-region. Finally, the improved ORB algorithm will be applied to extract feature points. This algorithm effectively solves the problem of being unable to extract feature points in environments with excessive or insufficient lighting, or a lower number of extracted feature points, while ensuring real-time performance and robustness.

This paper has the following contributions:

1) An improved ORB feature extraction algorithm is proposed, which combines image enhancement and truncated adaptive threshold methods to efficiently resolve the problem of insufficient feature points that are extracted using the ORB algorithm under the influence of factors such as lighting changes, noise, and outliers.

2) A method established on the improved truncated adaptive OTUS method is presented to calculate the threshold, which segments the image into multiple sub-regions for processing and can better adapt to various scenarios, thereby enhancing the robustness and adaptability of the ORB algorithm.

3) Experiments are made on multiple datasets to demonstrate the effectiveness and practicality of the presented algorithm in various scenarios, indicating promising application prospects.

The writing structure of the paper is given as follows: Section II reviews related works. Section III provides the methodology of the improved ORB algorithm. Experimental discussion and analysis are given in Section IV. Section V concludes the whole work.

II. RELATED WORK

The algorithm of ORB consists of the FAST feature detection and the binary descriptor. Compared with other feature extraction algorithms, it not only has higher accuracy, but also has a very fast processing speed, making it a practical and cost-effective algorithm.

A. FAST FEATURE POINTS DETECTION

The FAST algorithm depicted in Figure 1 initially selects a central point within an image, and subsequently identifies a continuous set of pixels within a fixed window area surrounding the selected center point. The grayscale difference between the central point and the continuous set of pixels is then calculated. If the grayscale difference of any pixel in the set exceeds a threshold that is pre-defined, then the central point is identified as a feature point.

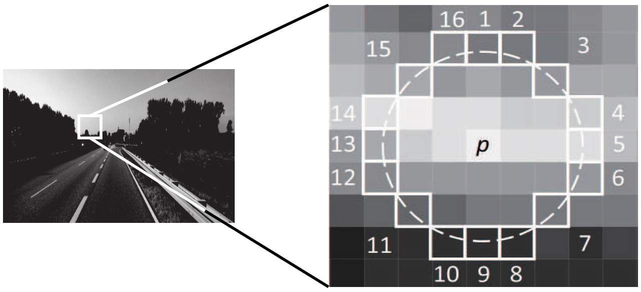


FIGURE 1. The whole algorithm structure of FAST feature extraction.

The specific steps of the original algorithm of FAST feature extraction are given as follows:

Step 1: Choose a pixel in the image as the center point f_{mid}, and record its grayscale value I_{mid}.

Step 2: Search for a region in the image by selecting N pixels within a neighborhood of radius R around the center point f_{mid}. Typically, R is set to 3 and N is set to 16.

Step 3: Set a threshold T, usually at 10% of f_{mid}.

Step 4: If K continuous pixels (empirical data shows that $K = 9$ usually gives the best results) among the N pixels have the same brightness value (either brighter or darker), the point is considered a corner point. Otherwise, move the center point and repeat steps 1-4 until all image pixels have been processed. The mathematical model for determining whether a pixel is brighter or darker is as follows:

$$S = \begin{cases} 0, & I_k > I_{\text{mid}} - T \\ 1, & I_k \le I_{\text{mid}} + T \end{cases} \tag{1}$$

where I_k represents the selected pixel's grayscale value, I_{mid} is the the center pixel's grayscale value, T denotes the threshold, and S is a binary variable indicating the brightness of the pixel, with 0 representing brighter and 1 representing darker.

To improve efficiency, one can first check whether the pixels at positions 1, 5, 9, and 13 satisfy equation (1), and if so, further check whether these pixels satisfy the continuity condition. If not, these pixels can be skipped directly.

B. BRIEF DESCRIPTION ALGORITHM

The BRIEF algorithm is used to calculate feature descriptors by comparing the grayscale values of pixel points. It generates a fixed-length binary string to describe the local features of the feature point.

In a neighborhood region with a range of $S \cdot S$ ($S = 9$), the BRIEF algorithm is able to select K pairs of pixel locations (As usual, $K = 256$) by utilizing Gaussian random sampling. For each pair, the algorithm makes comparisons with the grayscale values of the two pixels and generates a binary code based on the comparison result:

$$\tau(p; a, b) = \begin{cases} 1, & p(b) > p(a) \\ 0, & \text{other cases} \end{cases} \tag{2}$$

where a and b represent a pair of feature points, and their pixel gray values are defined as $p(a)$ and $p(b)$, respectively. Each dimension of the eigenvector is represented by τ. A binary feature vector that is K-dimensional is established by comparing pairs of points from K pairs.

The K-dimensional binary coded bit string that is used for selecting random point pairs is accomplished using the following method:

$$\mathrm{f}_k(p) = \sum_{1 \le i \le k} 2^{i-1} \tau\left(p; a_i, b_i\right) \tag{3}$$

where $\mathrm{f}_k(p)$ represents a feature vector with 256 dimensions.

C. ROTATION AND SCALE INVARIANCE

The ORB algorithm gives scale invariance by establishing an image pyramid [27]. To obtain rotation invariance, ORB utilizes the grayscale centroid algorithm to assign the FAST feature point with a direction [28], ensuring their rotational invariance. The detailed steps are given below:

1) The feature point is defined as follows:

$$m_{pq} = \sum_{x,y \in \mathrm{B}} x^p y^q I(x, y) \tag{4}$$

where m_{pq} of image is given for $p, q \in \{0, 1\}$. pq represents the coefficient, x and y denote the coordinates of the image pixel points, and $I(x, y)$ denotes the pixel point (x, y)'s gray value.

2) The image centroid can be calculated using the following formula:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{5}$$

where m_{00} and m_{01} represent the 0-th image, while m_{01} and m_{10} represent the 1-st image.

3) We can calculate the direction vector by connecting the centroid of images, which defines the direction belonging to the feature point as follows:

$$\theta = \arctan \frac{\mathrm{m}_{01}}{\mathrm{m}_{10}} \tag{6}$$

III. PROPOSED METHOD

A. IMAGE ENHANCEMENT

In complex lighting environments, the gray value belonging to a certain pixel in an image may be affected by external environmental information, leading to abrupt changes. These abrupt points have a negative impact on feature extraction. Therefore, the purpose of improving image quality is to eliminate these abrupt points and provide a good initial value for feature extraction. To obtain high-quality images, this paper proposes an image enhancement algorithm. The algorithm uses a Gaussian filter to preprocess the image to remove the influence of Gaussian noise. Then, a refined truncated adaptive gamma algorithm is proposed to further improve the visibility of the image. Finally, the image is improved by unsharp masking to achieve further improvement of image quality. Overall, the presented image enhancement algorithm is given in Figure 2.

The detailed steps of image enhancement processing are given below:

Step 1: Convert a digital image of size $M \cdot N$ to grayscale, with each pixel corresponding to a grayscale value $I_i(x, y)$. Here, M and N are positive integers and i ranges from 1 to $M \cdot N$.

Step 2: Calculate the deviation of the pixel grayscale values in the image:

$$\lambda = \sqrt{\frac{1}{N \cdot M} \sum_{y=1}^{N} \sum_{x=1}^{M} \left(I_i(x, y) - \overline{I(x, y)}\right)^2} \tag{7}$$

where $\overline{I(x, y)}$ represents the mean of the pixel grayscale values in the image.

Step 3: Judge the size of the standard deviation λ of the pixel grayscale values in the image. If it is greater than 0.25, the image quality is considered poor and proceeds to step 4; if it is less than 0.25, the image quality is considered good and proceeds to step 5.

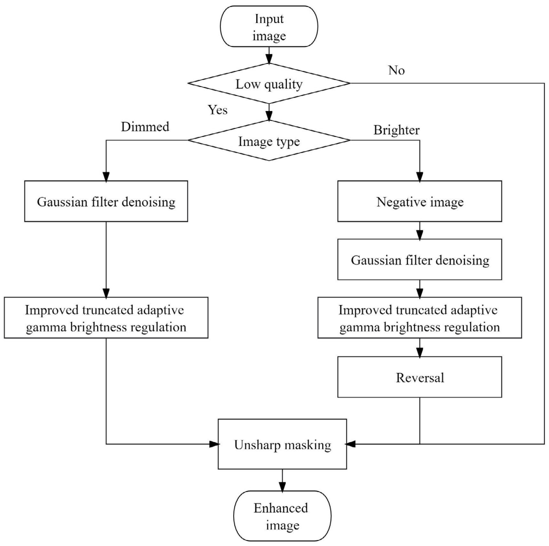

Input image
Low quality
No
Yes
Dimmed
Image type
Brighter
Gaussian filter denoising
Negative image
Gaussian filter denoising
Improved truncated adaptive gamma brightness regulation
Improved truncated adaptive gamma brightness regulation
Reversal
Unsharp masking
Enhanced image

FIGURE 2. **Structure of the proposed image enhancement method.**

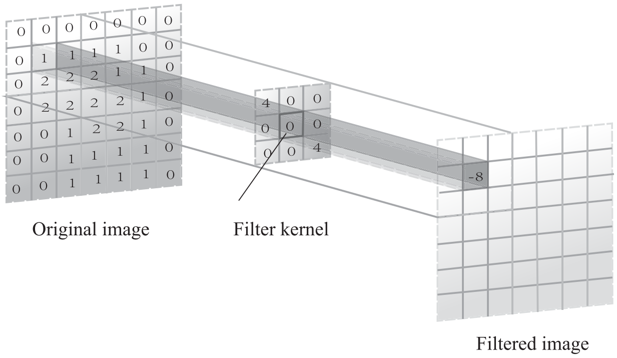


FIGURE 3. **Filter schematic diagram.**

Step 4: Judge the size of the mean pixel grayscale $\overline{I(x, y)}$ value in the image. If it is less than 128, the image is considered dark, and if it is greater than 128, the image is considered bright, then set $I_i(x, y) = 255 - I_i(x, y)$. Then perform Gaussian filtering to remove noise and adaptive gamma adjustment to adjust brightness.

Step 5: Perform unsharp masking on the image.

1) GAUSSIAN FILTERING

In order to eliminate Gaussian noise in the images, this paper adopts the Gaussian filtering algorithm, which convolves the image with a Gaussian kernel to achieve filtering operation. The shape of the kernel conforms to the form of the Gaussian distribution function. For each pixel in the image, the original grayscale value can be replaced by calculating the weighted average grayscale value of the surrounding neighborhood pixels, thereby suppressing noise. The principle structure of the Gaussian filtering algorithm is given in Figure 3.

The probability density function of the two-dimensional Gaussian function, namely the Gaussian convolution kernel, is:

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{8}$$

where σ represents the standard deviation of the normal distribution. x and y represent the horizontal and vertical coordinates of the pixel points with the top left corner of the image as the origin.

The mathematical model of Gaussian filter is:

$$f(x, y) = h(m, n) * I(x, y) \tag{9}$$

where $I(x, y)$ represents the original image's gray value, $*$ represents the result of the convolution operation, and $h(m, n)$ denotes the Gaussian convolution kernel.

2) TRUNCATED ADAPTIVE GAMMA BRIGHTNESS ADJUSTMENT

Inspired by reference [29], we improved the gamma correction algorithm to handle situations where the entire image is too dark or too bright. Assuming the total number of pixels in an image is:

$$n = N \cdot M \tag{10}$$

where N and M denote the width and length of the image, and are both positive integers. The probability density function for a grayscale level of i is provided by:

$$P_i = n_i * \frac{1}{n} \tag{11}$$

where n_i represents the pixel number that has the grayscale level i, and i can be selected as a natural number. The weighted probability density function is given by:

$$P_{w_i} = P_{\max} \cdot \left(\frac{P_i - P_{\min}}{P_{\max} - P_{\min}}\right)^{\alpha} \tag{12}$$

where $P_{\max}$ and $P_{\min}$ represent the P_i value in maximum and minimum level respectively, and α represents the smoothness factor, which is usually set to 0.2. The weighted distribution function of the image is given by:

$$C_{wi} = \sum_{h=0}^{i} \frac{P_{wi}}{\sum_{i=0}^{k} P_{wi}} \tag{13}$$

Then, the gamma parameter is designed as:

$$\gamma_i = 1 - C_{wi} \tag{14}$$

To prevent insignificant brightness enhancement, a truncated gamma parameter is designed as:

$$\gamma = \max(\tau, \gamma_i) \tag{15}$$

According to experimental analysis, a typical value for τ is 0.3. Substituting the truncated gamma parameter into the gamma function yields the gamma-corrected pixel value:

$$I'(x, y) = 255 \cdot \left(\frac{i}{255}\right)^{\prime} \tag{16}$$

Finally, the mask for contrast enhancement can be expressed as:

$$T_{\text{mask}}(x,y) = I'(x,y) - f(x,y) \tag{17}$$

3) UNSHARP MASKING

Unsharp masking is a method of image sharpening that enhances the contours and details of an image by using a low-pass filter to obtain a blurred image, creating a mask, and combining it with the original image. Compared to directly using sharpening convolution operators for image sharpening, the unsharp masking technique is more reliable and can effectively reduce the influence of noise and artifacts, making the image appear more natural and clear.

The unsharp mask can be obtained by subtracting the image blurred with a Gaussian filter from the original image:

$$G_{\text{mask}} = I(x,y) - f(x,y) \tag{18}$$

where $I(x, y)$ represents the gray level of the original image and $f(x, y)$ represents the gray level of the image after Gaussian filtering. Then the sharpened image can be expressed as:

$$I_{\text{unsharped}}(x,y) = I(x,y) + \alpha \cdot G_{\text{mask}}(x,y) \tag{19}$$

where α is the degree of sharpening, and for the unsharp mask technique, it is typically set to 1.

Finally, the image that has been enhanced can be represented by:

$$I_{\text{enhanced}} = \begin{cases} I(x,y) * h(m,n) + \alpha \cdot G_{\text{mask}}(x,y) \\ \quad +\beta \cdot T_{\text{mask}}(x,y) \\ I(x,y) * h(m,n) + \alpha \cdot G_{\text{mask}}(x,y) \end{cases} \tag{20}$$

where β represents mask level, and from experimental analysis, it is generally set to 0.3.

B. TRUNCATED ADAPTIVE THRESHOLD

Based on the previous analysis, it can be seen that the selection of the threshold is crucial for the ORB feature extraction process. However, manually setting the threshold in complex environments may result in degradation. Therefore, a more adaptive threshold calculation method is needed. To address this issue, this paper proposes an enhanced adaptive threshold calculation method based on the OTUS algorithm to improve the accuracy and stability of ORB feature extraction. Specifically, the working flow of this algorithm are as follows:

When dividing a digital image of size $M \cdot N$ into $m \cdot m$ sub-regions, it is generally assumed that $m = 5$. Within each sub-region, assuming there are a total of K grayscale levels with integer values ranging from 0 to 255, the probability density function of a grayscale level i is given by:

$$P_i = \frac{n_i}{n} \tag{21}$$

where n_i denotes the number of pixels corresponding to the grayscale level i, with a value range of 0 to K. n represents the total number of pixels in a sub-region. It should be noted that the above formula assumes that the grayscale level distribution of the pixels within each sub-region is the same, that is, the probability density function of each sub-region is identical. If a separate grayscale level analysis is required for each sub-region, the probability density function must be calculated separately for each sub-region.

For two types of pixel elements, C_0 and C_1 the probability of occurrence for each class is:

$$\begin{aligned} w_0 &= P(C_0) \\ &= \sum_{i=0}^{t} P_i \end{aligned} \tag{22}$$

$$\begin{aligned} w_1 &= P(C_1) \\ &= \sum_{i=t+1}^{K-1} P_i \\ &= 1 - P(C_0) \end{aligned} \tag{23}$$

The corresponding mean grayscale value is:

$$\begin{aligned} u_0 &= \sum_{i=0}^{t} i \cdot P(i \mid C_0) \\ &= \frac{\sum_{i=0}^{t} i \cdot P_i}{\sum_{i=0}^{t} P_i} \end{aligned} \tag{24}$$

$$\begin{aligned} u_1 &= \sum_{i=t+1}^{K-1} i \cdot P(i \mid C_1) \\ &= \frac{\sum_{i=t+1}^{K-1} i \cdot P_i}{\sum_{i=t+1}^{K-1} P_i} \\ &= \frac{\sum_{i=t+1}^{K-1} i \cdot P_i}{P(C_1)} \end{aligned} \tag{25}$$

The mean grayscale value from 0 to t is:

$$u = \sum_{i=0}^{t} i \cdot P_i \tag{26}$$

From the equation:

$$w_0 \cdot u_0 + w_1 \cdot u_1 = u \tag{27}$$

$$w_0 + w_1 = 1 \tag{28}$$

The inter-class variance can be obtained as:

$$\sigma_1^2 = w_0 \cdot (u_1 - u)^2 + w_0 \cdot (u_0 - u)^2 \tag{29}$$

The global variance is:

$$\sigma_2^2 = \sum_{i=0}^{K-1} (i-u)^2 \cdot P_i \tag{30}$$

The normalized separability measure is defined as:

$$\eta = \frac{\sigma_1^2}{\sigma_2^2} \tag{31}$$

As analyzed, it can be transformed into a least squares problem:

$$\max f(t) = \sigma_1^2 = \frac{\left(\sum_{i=0}^{t} P_i \cdot \sum_{i=0}^{K-1} i \cdot P_i - \sum_{i=0}^{t} i \cdot P_i\right)^2}{\sum_{i=0}^{t} P_i \cdot \left(1 - \sum_{i=0}^{t} P_i\right)}. \quad (32)$$

By solving the least squares problem, the optimal classification threshold t for each sub-region can be obtained. Therefore, the optimal threshold is:

$$C = a \cdot |I_{\mathrm{mid}} - t| \quad (33)$$

where the fixed parameter a is usually set to 0.3 according to experimental results. I_{mid} denotes the center pixel value of a sub-region, and t denotes the optimal classification threshold.

To avoid extracting too many low-quality feature points due to excessively small adaptive thresholds, this algorithm adopts a strategy of adaptive truncation threshold:

$$C' = \max\left(C, C_{\min}\right) \quad (34)$$

where the fixed parameter $C_{\min}$ is usually set to 7 according to experimental results.

IV. DISCUSSION AND ANALYSIS

A. EXPERIMENTAL ENVIRONMENT

This experiment used a laptop computer running Ubuntu 20.08 operating system, with an Intel (R) Core (TM) i7-7700HQ processor with a frequency of 2.60GHZ and 8GB of RAM.

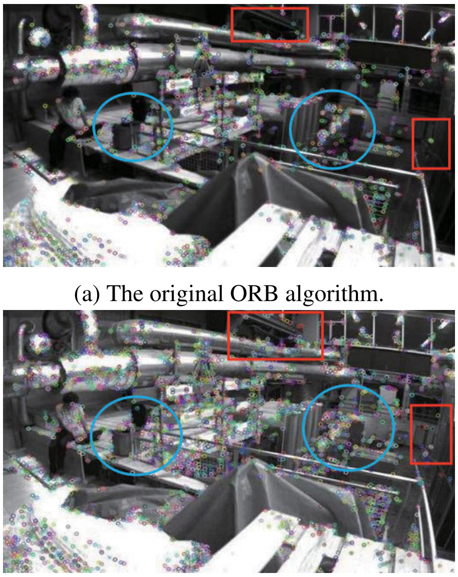
The original image is from the MH_01_easy open source dataset in Euroc, which is a classic set of robot vision datasets composed of image sequences captured during robot movement. Since the images in this dataset contain significant changes in brightness, it is particularly suitable for verifying the correctness and robustness of our algorithm.

B. PERFORMANCE COMPARISON

After multiple experiments, the original algorithm used in this study adopts a fixed threshold of 20 for feature point extraction, as it can extract more and more accurate feature points. The improved algorithm uses a parameter selection scheme with a sharpening degree of 1 and a mask degree of 0.3.

To make sure of the universality of the experimental results, a randomly selected image from the dataset was used for the experiment. The results show that the promoted ORB algorithm has a significant improvement in the number of feature points extracted compared to the original algorithm.

As shown in Figure 4, the original ORB algorithm extracts fewer feature points (marked by blue circles) in areas with poor lighting, and some important feature points are not even detected (marked by red boxes). The improved ORB algorithm, which combines image enhancement and truncated adaptive threshold, can extract more accurate feature points in the same region. Moreover, according to Table 2, the improved ORB algorithm extracts far more feature points than the original algorithm, with an increase in extraction time of only about 55ms.



(a) The original ORB algorithm.

(b) The improved ORB algorithm.

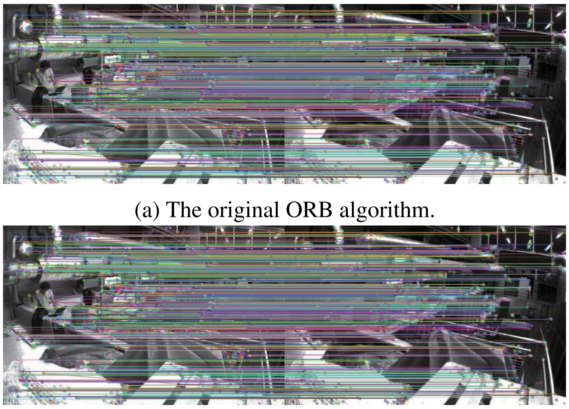
FIGURE 4. Illustration of feature point extraction comparison among different methods in frame 585.

TABLE 2. Comparison of feature point extraction performance between original ORB algorithm and improved ORB algorithm in the 585th frame.

Method	Original ORB	Improved ORB
Extraction number / (pcs)	8357	11625
Extraction time / (ms)	301.55	356.90

Based on the above analysis, the improved ORB algorithm shows a significant increase in the number of extracted feature points compared to the original algorithm in the frame 585. However, the increase in feature points alone does not necessarily indicate improved matching robustness under complex lighting conditions. Therefore, a set of images with different contrast levels (lower image quality) were used to compare the feature point matching performance of the original and improved algorithms. Figure 5 shows the feature matching results using the original and improved algorithms in the frame 585. According to Table 3, the experiment demonstrated that the improved ORB algorithm extracted significantly more feature points with a higher rate of correct matches. However, since no filtering of false matches was performed, there were also more false matches before and after the improvement.

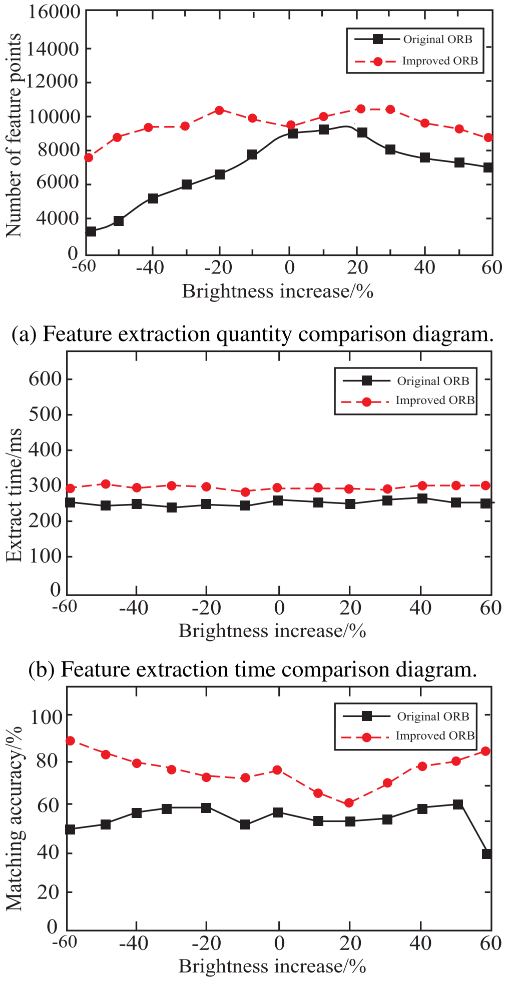
To further evaluate the adaptability of the improved ORB algorithm to complex lighting changes, tests were conducted on images with gradually changing brightness, including brightness changes ranging from ±60% of the original image.



(a) The original ORB algorithm.

(b) The improved ORB algorithm.

FIGURE 5. Comparison of matching result between original ORB algorithm and improved ORB algorithm in the 585th frame.



(a) Feature extraction quantity comparison diagram.

(b) Feature extraction time comparison diagram.

(c) Feature matching accuracy comparison diagram.

FIGURE 6. Comparison of feature extraction performance among different methods under varying lighting conditions in the 585th frame.

TABLE 3. Comparison of feature point matching accuracy between original ORB algorithm and improved ORB algorithm in the 585th frame.

Method	Original ORB	Improved ORB
Matching number / (pcs)	1244	1937
Matching accuracy / (%)	63.29	81.18
Matching time / (ms)	888.32	1428.34

Performance comparison was made with the original algorithm, as shown in Figure 6. The results showed that the improved ORB algorithm outperformed the original one, especially under extreme brightness changes, with a significant increase in the number of extracted feature points.

As shown in Figure 6a, the improved ORB algorithm extracts feature points more steadily and smoothly as the brightness changes. The greater the magnitude of brightness change, the more obvious the improvement in the number of feature points extracted. As shown in Figure 6b, the average feature extraction time of the improved ORB algorithm is only about 50ms higher than that of the original algorithm. As shown in Figure 6c, the accuracy of feature matching increases as the image becomes darker or brighter by the improved ORB algorithm. Therefore, although the improved algorithm increases some running time, it ensures the real-time performance of the system within an acceptable range, and improves its accuracy.

From the analysis of Figure 7, it can be seen that compared to the original algorithm, the proposed improved ORB feature extraction algorithm in this paper has increased the number and matching accuracy of feature points extracted in almost each frame, especially in environments with excessively dark or bright lighting conditions, where the effect is particularly significant.

In order to further demonstrate the superiority of the proposed improved ORB algorithm, performance comparisons were made with several state-of-the-art feature extraction algorithms in the same environment, namely SIFT, SURF, and BRISK, as shown in Table 4. The comparison results show that the ORB algorithm runs significantly faster than other feature extraction algorithms, and the improved algorithm has the highest number and precision of extracted features.

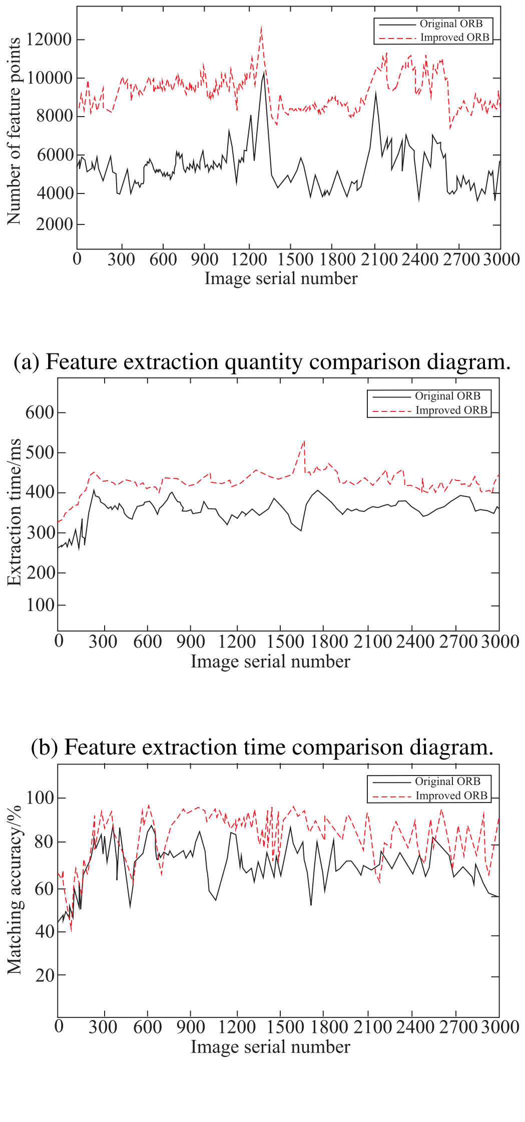
Based on the results of the experiments, it can be concluded that the proposed improved ORB feature extraction algorithm in this paper has a great improvement in the number and accuracy of extracted features in complex lighting environments and can satisfy the real-time requirements of the system.

C. FUTURE WORK

Due to the lack of outlier rejection in the experiment, there were many incorrect feature correspondences before and after the improvement. Additionally, no uniformization was applied, leading to clustering of some feature points. In future research, we will focus on improving the ORB feature

TABLE 4. Comparison of more feature extraction performance from different methods in MH_01_easy data set.

Method	Extraction number/(pcs)	Matching number/(pcs)	Matching accuracy/(%)	Total time/(ms)
Original ORB	54863254	435852	43.85	652301
SIFT	64158523	624582	53.42	1231456
SURF	68248534	451236	58.53	1524135
BRISK	38524630	548246	61.52	1435896
Improved ORB	76848245	843564	76.32	843259



(a) Feature extraction quantity comparison diagram.

(b) Feature extraction time comparison diagram.

(c) Feature matching accuracy comparison diagram.

FIGURE 7. Comparison of feature extraction performance from the improve before and after methods in MH_01_easy data set.

extraction algorithm by performing outlier rejection and uniformization processing. We will also apply the improved algorithm to ORB-SLAM to improve localization accuracy.

V. CONCLUSION

In this paper, the proposed improved ORB feature extraction algorithm is given to achieve reliable feature point extraction in complex lighting environments for real-time systems. The algorithm first uses Gaussian filtering for denoising, truncation adaptive gamma brightness adjustment, and unsharp masking to enhance the image, and proposes a method for obtaining the truncation adaptive threshold based on the OTUS algorithm. Compared with the original ORB feature extraction algorithm, this algorithm has stronger robustness, accuracy, and real-time performance, and can capture corresponding feature points under adverse lighting conditions. Experimental results show that the presented algorithm is able to increase the number of feature point extractions in complex lighting environments while maintaining a relatively stable extraction speed, and has a wide application in the future.

ACKNOWLEDGMENT

The authors are grateful to the data sets which are open access in Github.

REFERENCES

[1] Y. Xie, Q. Wang, Y. Chang, and X. Zhang, "Fast target recognition based on improved ORB feature," *Appl. Sci.*, vol. 12, no. 2, p. 786, Jan. 2022.

[2] L. Sun, X. Zhang, and W. Qin, "Research on target recognition and tracking in mobile augmented reality assisted maintenance," *Comput. Animation Virtual Worlds*, vol. 33, no. 6, Nov. 2022.

[3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[4] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, "A comprehensive survey of visual SLAM algorithms," *Robotics*, vol. 11, no. 1, p. 24, Feb. 2022.

[5] J. Cheng, L. Zhang, Q. Chen, X. Hu, and J. Cai, "A review of visual SLAM methods for autonomous driving vehicles," *Eng. Appl. Artif. Intell.*, vol. 114, Sep. 2022, Art. no. 104992.

[6] D. Ma and H.-C. Lai, "Remote sensing image matching based improved ORB in NSCT domain," *J. Indian Soc. Remote Sens.*, vol. 47, no. 5, pp. 801–807, May 2019.

[7] C. G. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, vol. 15, no. 50, pp. 5210–5244, 1988.

[8] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1994, pp. 593–600.

[9] G. Lowe, "Sift-the scale invariant feature transform," *Int. J.*, vol. 2, nos. 91–110, p. 2, 2004.

[10] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. 9th Eur. Conf. Comput. Vis.*, Graz, Austria: Springer, May 2006, pp. 430–443.

[11] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. 9th Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 3951, 2006, pp. 404–417.

[12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[13] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2548–2555.

[14] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 118–126.

[15] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3279–3286.

[16] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, "Learning guided convolutional network for depth completion," *IEEE Trans. Image Process.*, vol. 30, pp. 1116–1129, 2021.

[17] C. Geng, J. Yang, J. Lin, T. Yu, and K. Shi, "An improved ORB feature extraction algorithm," *J. Phys., Conf.*, vol. 1616, no. 1, Aug. 2020, Art. no. 012026.

[18] W. Guangyun and Z. Zhiping, "An improved ORB feature extraction and matching algorithm," in *Proc. 33rd Chin. Control Decis. Conf. (CCDC)*, May 2021, pp. 7289–7292.

[19] S. Wang, A. Zhang, and H. Wang, "A feature extraction algorithm for enhancing graphical local adaptive threshold," in *Intelligent Computing Theories and Application*. Xi'an, China: Springer, Aug. 2022, pp. 277–291.

[20] Q. Zang, K. Zhang, L. Wang, and L. Wu, "An adaptive ORB-SLAM3 system for outdoor dynamic environments," *Sensors*, vol. 23, no. 3, p. 1359, Jan. 2023.

[21] R. Wu, M. Pike, and B.-G. Lee, "DT-SLAM: Dynamic thresholding based corner point extraction in SLAM system," *IEEE Access*, vol. 9, pp. 91723–91729, 2021.

[22] L. Yu, E. Yang, and B. Yang, "AFE-ORB-SLAM: Robust monocular VSLAM based on adaptive FAST threshold and image enhancement for complex lighting environments," *J. Intell. Robotic Syst.*, vol. 105, no. 2, p. 26, Jun. 2022.

[23] S. Li, Q. Wang, and J. Li, "Improved ORB matching algorithm based on adaptive threshold," *J. Phys., Conf.*, vol. 1871, no. 1, Apr. 2021, Art. no. 012151.

[24] B. Nan, Y. Wang, Y. Liang, M. Wu, P. Qian, and G. Lin, "Ad-RMS: Adaptive regional motion statistics for feature matching filtering," in *Advances in Computer Graphics*. Cham, Switzerland: Springer, Sep. 2023, pp. 129–141.

[25] F. D. Adhinata and A. Harjoko, "Object searching on video using ORB descriptor and support vector machine," in *Advances in Computational Collective Intelligence*, D. Nang, Ed. Vietnam: Springer, Dec. 2020, pp. 239–251.

[26] G. Babu and P. A. Khayum, "Elephant herding with whale optimization enabled ORB features and CNN for iris recognition," *Multimedia Tools Appl.*, vol. 81, no. 4, pp. 5761–5794, Feb. 2022.

[27] Z. Zhang, L. Wang, W. Zheng, L. Yin, R. Hu, and B. Yang, "Endoscope image mosaic based on pyramid ORB," *Biomed. Signal Process. Control*, vol. 71, Jan. 2022, Art. no. 103261.

[28] Z. Dong, X. Sun, F. Xu, and W. Liu, "A low-rank and sparse decomposition-based method of improving the accuracy of sub-pixel grayscale centroid extraction for spot images," *IEEE Sensors J.*, vol. 20, no. 11, pp. 5845–5854, Jun. 2020.

[29] G. Cao, L. Huang, H. Tian, X. Huang, Y. Wang, and R. Zhi, "Contrast enhancement of brightness-distorted images by improved adaptive gamma correction," *Comput. Electr. Eng.*, vol. 66, pp. 569–582, Feb. 2018.

YONG DAI received the M.S. degree in control science and engineering from the University of Science and Technology Liaoning, Anshan, in 2016, and the Ph.D. degree in control theory and control engineering from the College of Marine Electrical Engineering, Dalian Maritime University, Dalian, in 2020. Since April 2021, he has been an Associate Professor with the School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang. His research interests include robotic nonlinear control theory and applications, navigation, simultaneous localization and mapping (SLAM), and self-driving intelligent vehicle systems.

JIAXIN WU received the B.E. degree in the Internet of Things engineering from the Henan University of Technology, Zhengzhou, China, in 2021. He is currently pursuing the master's degree with the School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang, China. His current research interests include robotic simultaneous localization and mapping (SLAM) and navigation and applications.