IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 May 2023, accepted 18 June 2023, date of publication 22 June 2023, date of current version 28 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3288594

RESEARCH ARTICLE

ORB Feature Matching Algorithm Based on Multi-Scale Feature Description Fusion and Feature Point Mapping Error Correction

CHENGXIAN YAO[1], HAIFENG ZHANG[1], JIA ZHU[1], DIQING FAN[1], YU FANG[1], AND LIN TANG[2]

[1]School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China
[2]School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China

Corresponding author: Haifeng Zhang (15901866629@163.com)

This work was supported by the Class III Peak Discipline of Shanghai—Materials Science and Engineering (High-Energy Beam Intelligent Processing and Green Manufacturing).

ABSTRACT To improve the accuracy of feature extraction and description of various scales in traditional Oriented FAST and Rotated BRIEF (ORB) feature matching algorithm, this paper proposes an ORB feature matching algorithm based on multi-scale feature description fusion and feature point mapping error correction. Firstly, when establishing the feature scale pyramid, the method of using the same patch-size for description on each feature layer is adopted instead of using different patch-sizes on a unified feature layer in the original algorithm, which enhances the robustness of the descriptor and improves the matching accuracy. Secondly, FAST-SCORE maps are established on different scales, and the coordinates of high-level feature points mapped to the bottom layer are corrected to further improve the positioning accuracy of feature points. The algorithm is verified in remote sensing images, autonomous driving, and industrial automation fields. Experimental results show that when resisting theoretical interference, the average matching accuracy of the proposed algorithm is 67.9%, which is about 2.0 times that of the ORB algorithm, and the average stability is 14.0, which is about 1.5 times that of the ORB algorithm. After correcting the feature point mapping, the matching accuracy can be further improved by 19.2%, indicating that the improved algorithm has excellent robustness when resisting interference. In the experiments on the KITTI and custom datasets, the matching accuracy of the proposed algorithm reached 88.70% and 96.88%, respectively, which is an improvement of 10.15% and 1.2% compared to the ORB algorithm. At the same time, the matching time was reduced by 17.34% and 24.30%, ensuring the accuracy and real-time performance of the algorithm in practical scenarios.

INDEX TERMS Feature matching, machine vision, image processing, ORB.

I. INTRODUCTION

Feature matching algorithm is one of the important research technologies in the field of computer vision. Compared with traditional template matching algorithms, feature matching algorithms can take into account the local sub-features of objects, thereby improving matching accuracy and speed. Therefore, it has been widely used in many fields such as 3D reconstruction, remote sensing image stitching, and robot SLAM [1], [2], [3], [4], [5]. However, there are still some problems in the feature extraction part of the algorithm. In practical environments, it is easy to introduce a lot of interference information, which affects the subsequent feature matching and tracking work. Therefore, how to improve the matching accuracy and robustness of the algorithm in practical application scenarios has always been the goal of many researchers.

Currently, the main idea of feature matching algorithms is to extract feature points from images and describe them, comparing the similarity between feature points in two images to determine whether they match. The SIFT algorithm [6] is a classic feature matching algorithm that has rotation and

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

scale invariance and good robustness to changes in lighting, but it is computationally expensive. To address this issue, the SURF algorithm [7] was proposed, which uses integral images for image convolution, and a metric based on the Hessian matrix and a distribution-based descriptor for the detector to speed up the extraction process. The algorithm's efficiency is significantly improved compared to SIFT, but it still cannot meet the real-time requirements of most use cases. The ORB algorithm [8] is a feature matching algorithm that achieves high efficiency while implementing rotation invariance, but its matching accuracy is not as good as SIFT. Currently, improvements to feature matching algorithms are mainly carried out in two aspects. The first aspect is to improve the ability to extract and describe image features. Gao and Sun [9] used the position and orientation system (POS) to simulate image distortion to improve the quality of feature extraction. Mur-Artal and Tardós [10] proposed the Qtree_ORB (quadtree-based ORB algorithm) algorithm based on ORB, which uses a quadtree to segment the image and combines adaptive thresholds to extract feature points. Yao et al. [11] further limited the segmentation depth by using the quadtree segmentation technique. Wu [12] analyzed the influence of the number of layers in the image pyramid on the matching accuracy of ORB feature points and proposed the optimal parameter distribution of the algorithm to improve the matching accuracy. Lv et al. [13] proposed a method based on hue, saturation, and value (HSV) and histogram equalization, which extracts the HSV information of the image to construct a feature vector as the descriptor information and uses the PatchMatch algorithm with local consistency to search for the nearest neighbor for matching. Finally, the dense linear fitting method is used to improve the detection accuracy. Ma et al/ [14] used the difference in image grayscale information as a new feature descriptor, and Yan et al. [15] constructed a fusion of ORB feature descriptors and point cloud feature descriptors to further improve the matching effect. The second aspect is to improve the screening ability of feature point matching. Muja and Lowe accelerated the matching speed of high-dimensional descriptors by establishing kd-tree [16] and k-means tree [17], but could not effectively eliminate mis-matched feature points. Bian et al. [18] proposed the Grid-based Motion Statistics (GMS) algorithm, which uses the support in the matching point neighborhood to identify mis-matched points, but the matching accuracy decreases when the image is disturbed by blur and other factors.

This paper aims to improve the effectiveness of image feature extraction, thereby reducing the mismatch rate and improving the matching accuracy. To achieve this, we propose a multi-scale feature fusion-based ORB feature matching algorithm. The algorithm first establishes a scale feature pyramid, performs feature extraction and description on each layer, and then fuses the extracted feature points from each layer. When mapping feature position information downward, we use the FAST-SCORE map of the bottom layer to correct the coordinates of the high-level feature points, thereby improving the positioning accuracy of the feature points. Finally, we use the RANSAC algorithm to eliminate mismatched points. This method not only ensures the real-time performance of the algorithm but also improves the matching accuracy and robustness.

The innovations of this paper mainly include the following aspects:

(1) Adaptively describing the corresponding feature layer based on the scale information of the feature points, enhancing the robustness of the descriptor;

(2) Establishing FAST-SCORE maps at different scales and correcting the coordinates of high-level feature points mapped to the bottom layer, further improving the positioning accuracy of the feature points.

Specifically, in the second section of this paper, we introduce the improved multi-scale feature description method and coordinate mapping correction method. In the third section, we compare the previous algorithm with the proposed algorithm through experiments and summarize the experimental results. Finally, in the fourth section, we conclude the paper.

II. IMPROVED ALGORITHM

The proposed improved algorithm in this paper consists of three main parts: image preprocessing, multi-scale feature description, and feature point coordinate mapping correction, as shown in Figure 1. The image preprocessing part mainly aims to reduce the influence of noise and illumination changes on matching accuracy by performing denoising and histogram equalization on the input image. The multi-scale feature description part utilizes the FAST algorithm to quickly screen feature points at each feature layer and perform feature description based on their scale information in the corresponding layer. The feature point coordinate mapping correction part first establishes FAST-SCORE maps at various scales of the image and calculates the optimal target position of each high-level feature point in the corresponding position of the FAST-SCORE map, and finally performs coordinate correction.

A. IMAGE PREPROCESSING

Preprocessing is an important step in image matching, which mainly includes two parts: image denoising and histogram equalization. In the process of image matching, the original image may be affected by imaging hardware and environmental factors, resulting in noise that affects the subsequent matching effect. In order to preserve the fine details of the image as much as possible while removing these noise points, we use the P-M equation (anisotropic diffusion partial differential equation) for denoising. This method uses the gradient of the image as an edge detection operator, which can protect the boundary information of the image while denoising [19]. Secondly, we use histogram equalization technology to enhance the contrast of the image and improve the perception ability of machine vision for key features of the image. This method is less affected by external lighting changes and

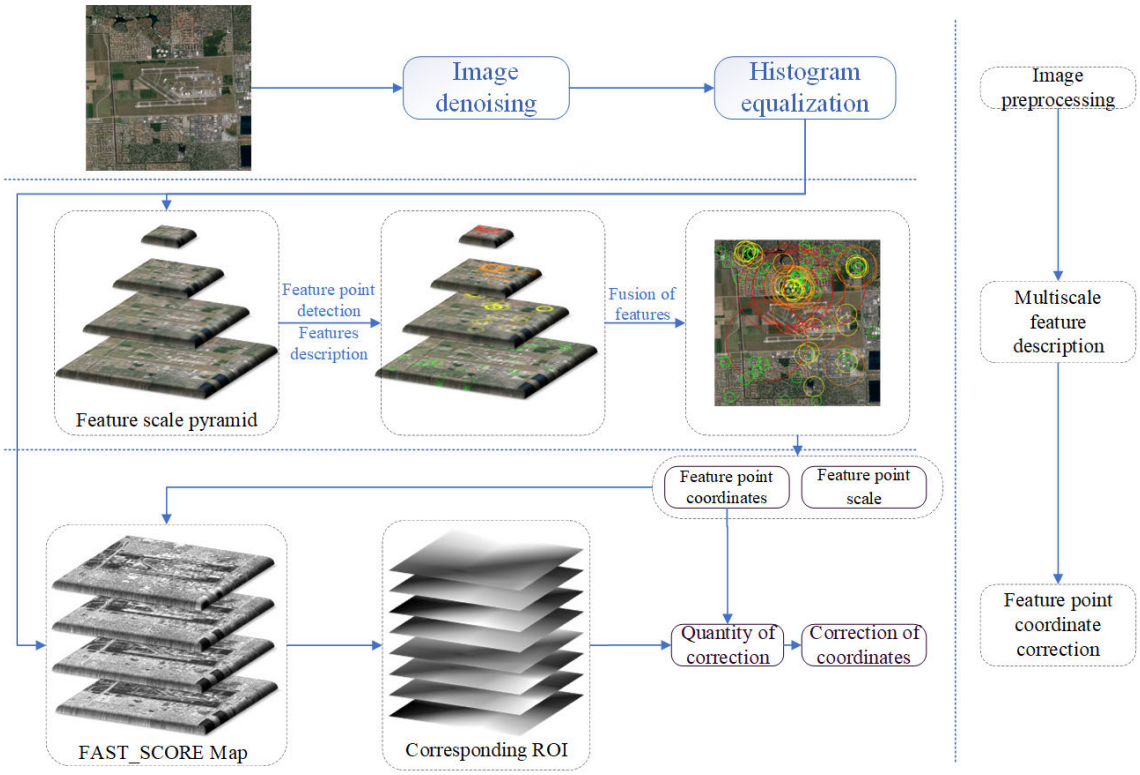

FIGURE 1. Improved feature extraction algorithm framework.

has good stability. After the above two steps of processing, the image will be sent to the multi-scale feature description part for the next step of operation, in order to achieve more accurate matching results.

B. MULTISCALE FEATURE DECRIPTION

ORB features are one of the commonly used image features in the field of computer vision, and are widely applied due to their rotation invariance and real-time performance. The ORB algorithm uses the FAST algorithm to determine feature points by comparing the difference between the pixel values of the test point and its surrounding pixels. However, since the FAST algorithm compares points on a radius of 3 around the feature point, the extracted corner points themselves do not have scale invariance.

To address this issue, the ORB algorithm establishes a feature scale pyramid and extracts FAST corner points on each layer to achieve scale invariance of the corner points. Meanwhile, the ORB algorithm uses BRIEF descriptors to describe feature points, which have the advantages of rotation invariance and fast matching. However, the descriptors of the ORB algorithm still rely on the mapping position of feature points on the bottom layer image for description [20], which leads to the introduction of invalid information into the majority of feature point descriptors when the image loses pixel-level features due to external factors such as changes in lighting and motion blur, thereby reducing the matching performance of the algorithm.

To enhance the effectiveness of keypoint description in our algorithm, we have improved the original feature scale pyramid. The specific steps of the algorithm are as follows:

1) Establish a feature scale pyramid, where each layer of the image is obtained by Gaussian blur and downsampling of the next layer. The bottom layer of the pyramid has a larger image size and contains low-dimensional texture features, while the upper layers have smaller image sizes but retain large-scale global features. Describing at different layers is beneficial for the algorithm to extract features at various scales. The number of layers L in the pyramid is mainly determined by the size of the image and the patch-size of the descriptor, which is calculated as follows:

$$L = [\log_k(\min(w, h)) - \log_k(patch\_size \cdot 4)] \quad (1)$$

where, k is the downsampling coefficient (here k is set to 2), w and h are the width and height of the image, and $patch_size$is the scale of the feature point description.

2) Utilize the FAST algorithm to extract feature points in each layer of the pyramid image. To ensure a balanced distribution of feature points in each layer, the number of feature points N_f^i extracted in each layer should be adjusted according to the downsampling coefficient k, and satisfy the following equation:

$$\sum_{i=1}^{L} \frac{1}{k^{i-1}} N_f^i = N_f \quad (2)$$

where, N_f represents the total number of extracted feature points, which is usually set to 500 in the ORB algorithm.

For each layer of the image, the number of feature points extracted based on the FAST algorithm, n_f^i, is compared with the predetermined number of feature points for that layer, N_f^i. If $n_f^i \geq N_f^i$, we select the top N_f^i best points based on the FAST score. If $n_f^i < N_f^i$, it indicates that the texture features at the corresponding scale of that layer have been affected and it is not possible to extract enough feature points effectively. In this case, blindly lowering the feature point extraction threshold to increase the number of feature points may introduce a large number of invalid feature points, making subsequent matching difficult [21]. Therefore, this paper keeps the feature point extraction threshold unchanged and distributes the missing feature points proportionally based on the remaining layers' N_f^i. This method can extract enough feature points while ensuring the quality and uniform distribution of feature points extracted from each layer.

Due to the characteristics of feature scale pyramid, as the number of layers increases, the size of the image decreases while retaining higher-dimensional scale features [22]. In the original algorithm, different scales of features were described by changing the *patch_size* of the descriptor directly on the bottom layer image. However, the receptive field of the descriptor does not increase with the increase of *patch_size*, which leads to insufficient description of large-scale features of the target. Therefore, a method is needed to adjust the receptive field of the descriptor according to the demand. In this paper, descriptors with the same *patch_size* are used to describe different scales of features on each feature layer, as shown in Figure 2. As the feature pyramid increases, the receptive field of the image expands with the increase of layers, so descriptors with the same *patch_size* can also fully describe various scales of features on each feature layer. The equivalent feature scale $patch\_size_r$ extracted from each feature layer satisfies:

$$patch\_size_r = patch\_size \cdot k^{i-1} (1 \leq i \leq L) \tag{3}$$

3) In order to achieve scale invariance in feature matching algorithms, it is necessary to fuse feature information from different scales. In the pyramid structure, the coordinates of features in the upper layers need to be mapped to the lower layers in a top-down manner for matching, while keeping the descriptors unchanged. In order to ensure that the relative positions of upper-level feature points remain unchanged when mapped to lower-level images, the mapping relationship needs to satisfy Equation (4):

$$\{(x_i, y_i) = k^{j-i}(x_j, y_j) + 0.5 \cdot k^{j-i} | 1 < i < j < L\} \tag{4}$$

In the equation, (xi, yi) $(x_i, y_i)$and (x_j, y_j) represent the coordinates of feature points in the i-th and j-th layers, respectively, while k denotes the downsampling factor.

The following is the pseudocode of Algorithm 1, which demonstrates the implementation of multi-scale feature description fusion.

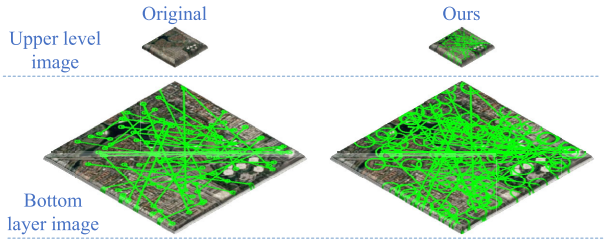

FIGURE 2. Description method in this article.

Algorithm 1 Multi-scale Feature Description Fusion Method

Input :*initial_image*,
Output :*Key_points*,*descriptors*

```
for i =1; i < NL; i + +do
    image_i = blur&down_sample(initial_image)
    ORB_detect(image_i)
    If nf > Nf then
        Key_points ← thefirstNf Key_points based on
    FAST SCORE
    else
        Upgrade otherNf
    ORB_describe(Key_points)
    Coordinate mapping(Key_points)
```

C. FEATURE POINT COORDINATE MAPPING CORRECTION

Compared to low-level images, high-level images can extract more global features, thus possessing stronger adaptability to interference. Figure 3 shows the distribution of matching errors of each feature layer under the influence of interference factors. It can be observed from the figure that in practical application scenarios, the matching loss of feature extraction in high-level images is significantly higher than that in low-level images, demonstrating stronger anti-interference ability.

Figure 4 illustrates the distribution of matching errors for each feature layer under the condition of macroscopic matching correctness. It can be observed from the figure that the micro-matching loss continuously expands with the increase of layers. This is because the high-level images undergo multiple downsampling processes, resulting in a significant reduction in the precision of feature point coordinates. As a result, when using a fixed upsampling method to map coordinates back to the original image, the coordinate error will increase exponentially with the increase of layer difference, thereby affecting the performance of the algorithm.

In order to alleviate the phenomenon of magnified matching errors when mapping high-level feature point coordinates to the bottom level, it is necessary to adopt a method that integrates high-level image feature information with bottom-level image feature point location information. Specifically, while extracting features from the high-level image, the texture information of the bottom-level image is analyzed to correct the coordinates of the mapped feature points, thereby achieving more accurate matching results.

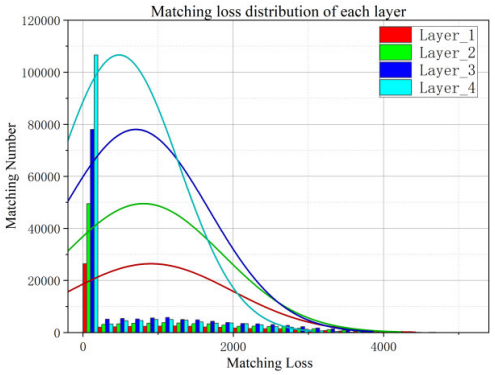

FIGURE 3. Macroscopic matching error distribution of each feature layer.

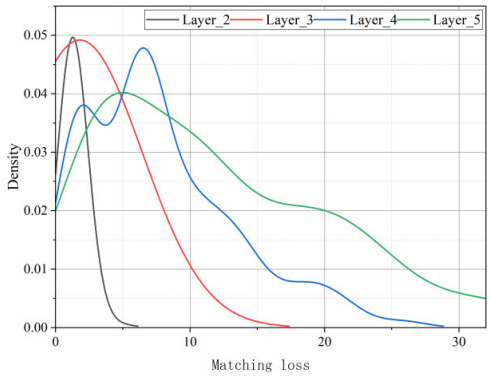

FIGURE 4. Microscopic matching error distribution of each feature layer.

In the pyramid algorithm, the image feature points in each layer are extracted using the FAST9-16 algorithm. To obtain the correction amount of each layer's feature points in the bottom layer image, it is necessary to establish the FAST-score map of the bottom layer image in each scale range. To improve algorithm efficiency and reduce computational complexity, different scales of box filter $f(x, y, s)$ are used instead of Gaussian filters. Based on the number of layers in the previous feature scale pyramid, the original image $I(x, y)$ is convolved at different scales to obtain each layer image $I_i(x, y)$, which satisfies:

$$I_i(x, y) = f(x, y, s_i) * I(x, y), 1 \le i \le N_L + 1 \quad (5)$$

$$f(x, y, s_i) = \frac{1}{s_i^2}, 1 \le x, y \le s_i \quad (6)$$

where, * denotes convolution calculation,and s_i represents the scale factor of the filter.

In order to efficiently obtain FAST-SCORE maps at different scales, we perform a weighted differential calculation on the L+1 images, followed by a difference calculation with the original image $I(x, y)$, resulting in L approximate FAST-SCORE maps. Taking the construction of a 4-layer feature pyramid as an example, the specific process is illustrated in Figure 5.

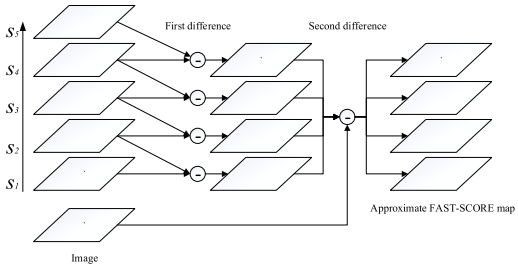

FIGURE 5. Approximate fast_score flowchart.

In the figure 5, each layer of the approximate FAST-SCORE graph is obtained from equations (7) to (9)

$$FAST - SCORE_i = \delta_{i+1\to i} \cdot I_{i+1}(x, y) - \delta_{i\to i} \cdot I_i(x, y) - I(x, y) \quad (7)$$

$$\delta_{i+1\to i} = \frac{s_{i+1}^2}{s_{i+1}^2 - s_i^2} \quad (8)$$

$$\delta_{i\to i} = \frac{s_i^2}{s_{i+1}^2 - s_i^2} \quad (9)$$

where, $\delta_{i+1\to i}$ represents the weighting coefficient between the $i+1$th layer image of the filtering layer and the ith layer image of the differential layer.

When mapping the coordinates of feature points in the upper layer image to the lower layer image, it is necessary to consider the score distribution of the corresponding ROI region in the approximate FAST-SCORE map, in order to calculate a suitable target position. The specific calculation process of this position is shown below:

According to the score values within the ROI region, the coordinates within the region are weighted and the geometric moments are calculated under different weighting coefficients χ. Specifically, the following formula (10) can be used:

$$M_{pq}^{\chi} = \int\int x^p y^q I^{\chi}(x, y) dxdy, (x, y) \in ROI \quad (10)$$

For a two-dimensional discrete function representing an image, the above equation can be rewritten as:

$$M_{pq}^{\chi} = \sum_{(x,y)\in ROI} x^p y^q I^{\chi}(x, y) \quad (11)$$

Therefore, the centroid C^{χ} of the scores within the ROI region can be obtained by the following formula:

$$C^{\chi} = \left(\frac{M_{10}^{\chi}}{M_{00}}, \frac{M_{01}^{\chi}}{M_{00}}\right) \quad (12)$$

The weighting coefficient χ represents the correlation between the score distribution within a region and the mapping result. As χ changes, C represents the target position under different selection methods, thereby affecting the mapping effect. When $\chi = 0$, C is the geometric center of the region; when $\chi = 1$, C is the centroid of the region; when $\chi \to \infty$, C is the coordinate of the highest score point within the region. If χ is too large, the mapping process is prone to local optimal points, which affects the matching accuracy.

If χ is too small, the mapping amount is too small, and the mapping effect is not significant. Therefore, a momentum coefficient η is introduced to enhance the mapping effect while avoiding local optimal solutions. Since the mapping adjustment is limited to the ROI region where the feature point is located, the upper bound of the mapping amount needs to be restricted. Thus, the mapping correction relationship based on the approximate FAST-SCORE map is as follows:

$$(x_i, y_i)_{1<i<j<L} = \begin{cases} k^{j-i}(x_j, y_j) + C^0 + \eta(C^\chi - C^0) 0 < \eta \leq 1 \\ k^{j-i}(x_j, y_j) + C^\chi + (\eta - 1)(C^{-1} - C^\chi)\ 1 < \eta \leq 2 \end{cases} \tag{13}$$

In this experiment, a comparative study was conducted by varying the values of χ and η, and applying the mapping correction method described above. For each mapping correction method, the mapping accuracy was calculated under the maximum layer difference, and the results were presented in the form of a heatmap, as shown in Figure 6.

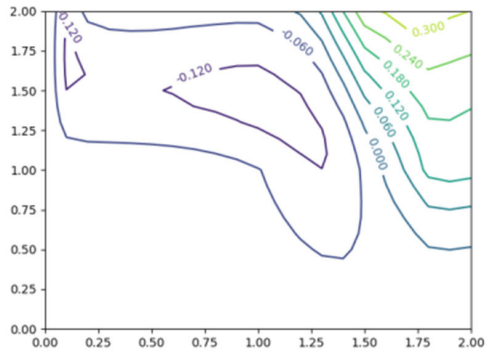

FIGURE 6. Heatmap of mapping accuracy.

From the above figure, it can be observed that the mapping error is significantly reduced when χ and η are respectively set to 1 and 1.5. This mapping method can determine the mapping coordinates by comprehensively considering the overall score level within the ROI region. However, the upper right region of the figure, although showing better results, is susceptible to interference from individual scores within the ROI region, which may affect the mapping accuracy. Therefore, further improvements are needed to enhance the accuracy and stability of the mapping method.

The following is the pseudocode of Algorithm 2, which demonstrates the process of coordinate mapping correction for the Lth layer.

Algorithm 2 Coordinate Mapping Correction

Input :*initial_key_points*
Output :*improved_key_points*
1 $FS\_map^{[L]} = FAST\_SCORE\_map(image)$
2 **for** $i = 1; i < Nf; i++$ **do**
3 $(x\_offset, y\_offset) = C^1(FS\_map^{[L]}.ROI)$
4 $key\_point[i].x += x\_offset$
5 $key\_point[i].y += y\_offset$
6 **End**

III. EXPERIMENT AND RESULTS

A. EXPERIMENTAL ENVIROMENT

The computer configuration used in this study includes an i7 processor and 16GB of memory. The experimental images were obtained from the DOTA [23] dataset, KITTI [24] dataset, and industrial application scenarios. To validate the effectiveness of our algorithm, a certain number of images were selected for testing. The experiment will evaluate the algorithm's performance from both theoretical and practical perspectives. The theoretical performance experiment includes tests of anti-interference ability, feature point matching quality, and coordinate mapping correction effects, using the DOTA dataset for validation. The practical performance experiment includes tests of autonomous driving and industrial application scenarios, using the KITTI dataset and actual industrial case images for validation. The experiment will comprehensively evaluate the algorithm's performance based on metrics such as matching accuracy, stability (standard deviation of matching accuracy), AP, and matching error. The calculation methods for each evaluation metric are as follows:

Based on the algorithm mentioned above, the matching point information of the two images is obtained, and the Hamming distance between each feature point is calculated. A coarse screening is performed based on the Hamming distance between the nearest neighbor point pairs and the second nearest neighbor point pairs. The selected point pairs m satisfy the following condition:

$$m.dis\tan ce < cof \cdot n.dis\tan ce \tag{14}$$

where, m represents the closest neighbor pair, n represents the second closest neighbor pair, and cof represents the predetermined screening threshold.

The percentage of the number of feature points N_C after coarse screening to the total number of matching points N_S can be expressed as the coarse screening rate P:

$$P = \frac{N_c}{N_s} \times 100\% \tag{15}$$

Based on the preliminary screening of matching point information, the RANSAC algorithm is used to filter the inliers, and the percentage of inliers N_i to the number of initially screened feature points N_c is calculated as the matching accuracy. Specifically, the matching accuracy A can be expressed as:

$$A = \frac{N_i}{N_c} \times 100\% \tag{16}$$

Due to the direct impact of coarse screening rate on the accuracy of matching, it is generally believed that the lower the coarse screening rate, the higher the matching accuracy. However, in a single situation, the coarse screening rate

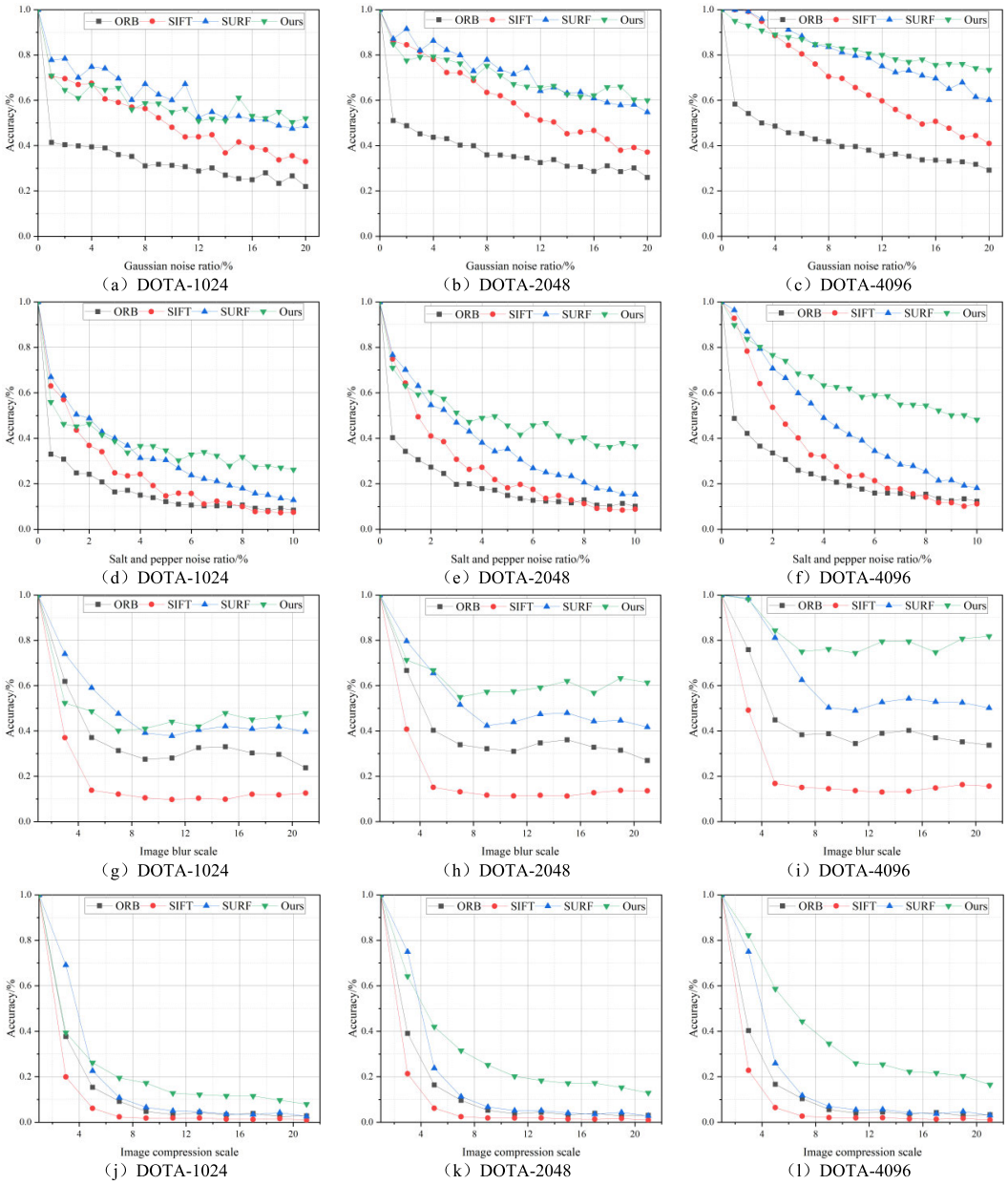

FIGURE 7. (a-l) Variation of matching accuracy for each algorithm.

and matching accuracy cannot comprehensively evaluate the overall performance of the algorithm. Therefore, this paper introduces a new evaluation index AP to take into account the algorithm performance in various situations. The specific calculation method is as follows:

$$AP = \int_0^1 A \mathrm{d}P \tag{17}$$

B. COMPARATIVE EXPERIMENT ON ANTI-INTERFERENCE ABILITY

The aim of this experiment is to compare the feature extraction performance of ORB, SIFT, SURF, and the proposed algorithm on the DOTA dataset. The DOTA dataset is a large-scale dataset for aerial images, containing 2806 images with pixel sizes ranging from 800*800 to 4000*4000, covering various scales of feature information, which can be used to evaluate the matching performance of algorithms at different scales. In this experiment, the images in the dataset are divided into three groups according to their sizes, namely 1024*1024, 2048*2048, and 4096*4096, to detect the matching performance of each algorithm at different scales. To ensure parameter consistency, consistent interference information is introduced into the images, and the coarse screening rate of each algorithm is unified to 80% for the experiment.

The experimental results in Figure 7 show that after introducing interference factors such as salt and pepper noise, Gaussian noise, image compression, and image blur, the

TABLE 1. Summary of matching accuracy for each algorithm.

Dataset	Algorithms	Salt and pepper noise		Gaussian noise		Image blur		Image compression	
		Average accuracy/%	Stability/%	Average accuracy/%	Stability/%	Average accuracy/%	Stability/%	Average accuracy/%	Stability/%
DOTA -1024	ORB	19.38	19.40	34.87	15.68	39.56	21.38	17.04	28.03
	SIFT	26.09	22.82	52.30	16.11	21.80	25.82	12.65	28.12
	SURF	34.56	20.91	62.96	12.98	51.14	18.64	21.13	31.08
	Ours	38.71	15.65	59.76	10.74	50.47	16.04	24.38	25.44
DOTA -2048	ORB	22.12	19.32	39.32	15.16	42.38	20.82	17.43	28.05
	SIFT	29.40	23.98	60.86	17.72	23.19	25.61	12.82	28.14
	SURF	39.58	22.33	72.67	12.21	55.36	17.94	21.98	31.84
	Ours	50.26	14.16	71.18	9.54	64.60	12.09	33.08	25.59
DOTA -4096	ORB	26.24	19.33	43.12	14.81	47.03	20.71	17.79	28.04
	SIFT	35.52	26.70	68.41	19.59	25.68	25.50	13.05	28.15
	SURF	48.48	25.43	80.59	12.51	63.99	18.75	22.41	31.75
	Ours	65.15	13.81	83.12	7.23	82.19	8.51	41.07	26.58

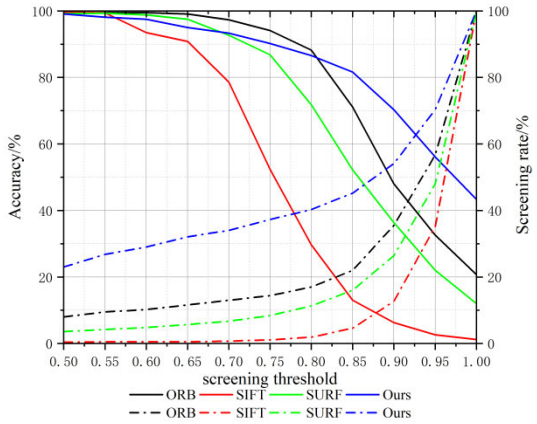

FIGURE 8. P-A Curve.

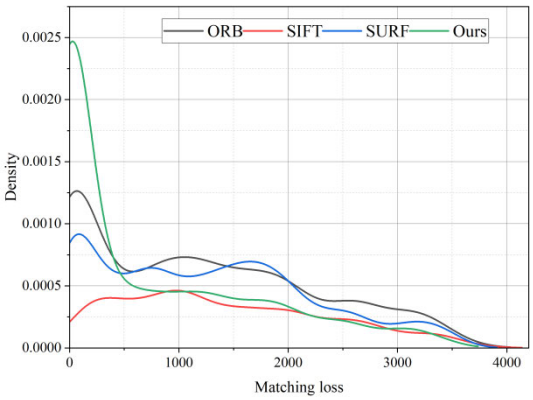

FIGURE 9. Distribution of matching errors.

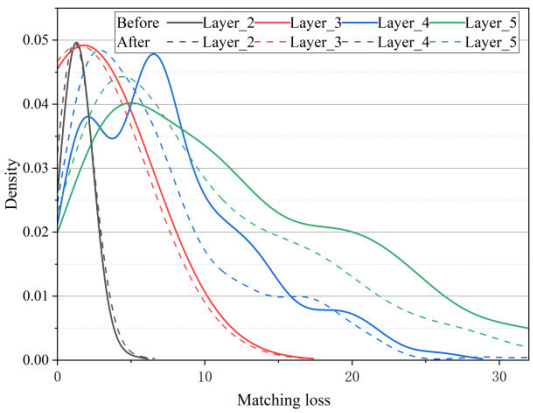

FIGURE 10. Comparison of matching loss distribution before and after mapping correction.

matching accuracy of the compared algorithms is significantly affected, and the decrease is obvious. The proposed algorithm has improved the accuracy decrease compared to other algorithms.

From Table 1, it can be seen that the proposed algorithm achieves significant improvements in matching performance compared to other algorithms. Particularly, in the DOTA-1024 dataset with lower image resolution, the proposed algorithm performs best in salt-and-pepper noise and image compression, with a respective increase of 19.33% and 7.34% in matching accuracy compared to the original algorithm. Although the proposed algorithm is slightly inferior to the SURF algorithm in Gaussian noise and image blur, it shows significant advantages in all aspects in the DOTA-4096 dataset with larger image resolution.

Specifically, when introducing salt-and-pepper noise and Gaussian noise, the proposed algorithm improves the matching accuracy by 38.91% and 40%, respectively, compared to the original algorithm, while the stability is also improved by 5.52% and 7.58%, indicating an enhanced ability to resist noise. In the case of image blur, the proposed algorithm improves the matching accuracy by 35.16% and the stability by 12.2%. In the case of image compression, the proposed algorithm improves the matching accuracy by 23.28% and the stability by 1.46%.

C. COMPARATIVE EXPERIMENT ON FEATURE POINT MATCHING QUALITY

Generally speaking, the matching accuracy of an algorithm is greatly influenced by the coarse screening rate. The lower the coarse screening rate, the higher the matching accuracy, but too low a coarse screening rate will result in an insufficient number of matched point pairs, thereby affecting the matching effect. Therefore, an excellent matching algorithm needs to ensure a certain matching accuracy while a sufficient number of point pairs are screened out. In this

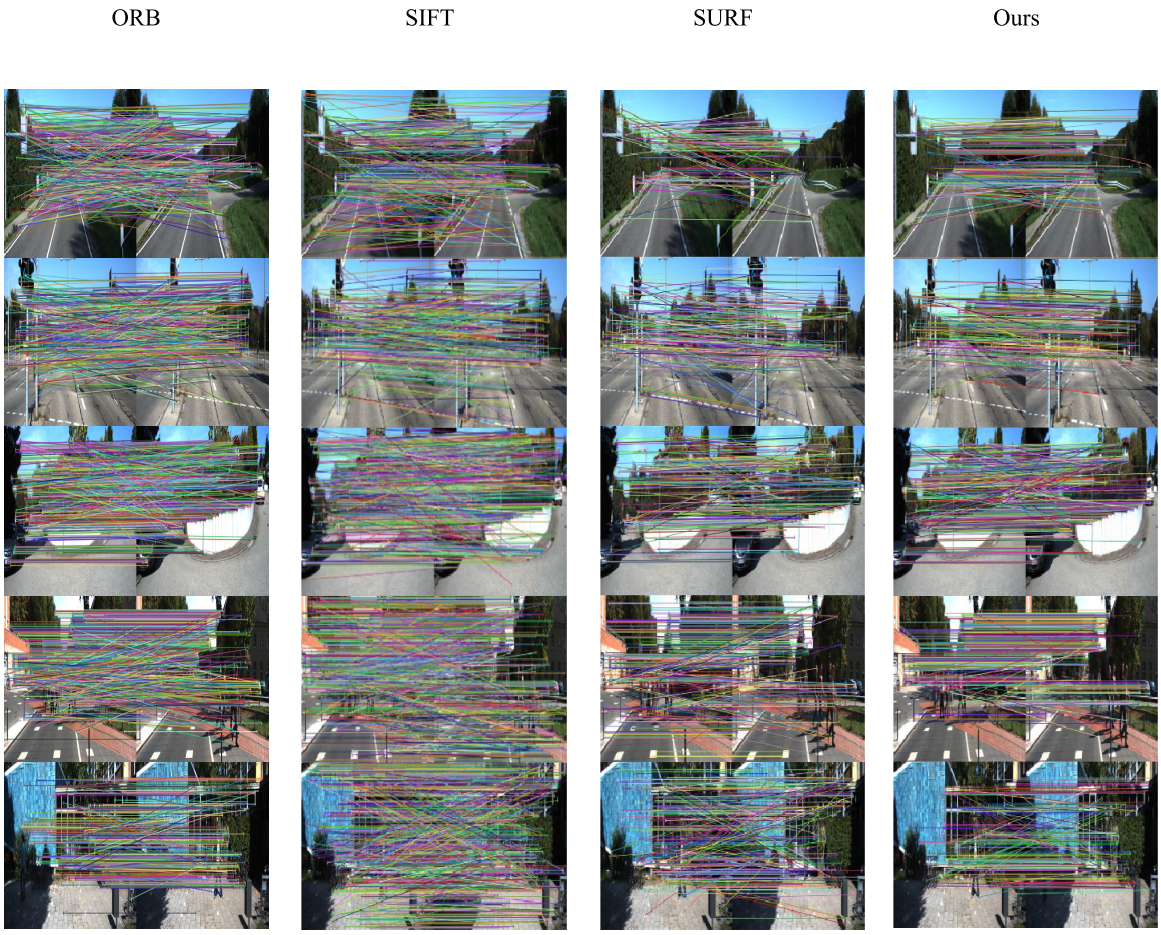

FIGURE 11. Matching results of each algorithm.

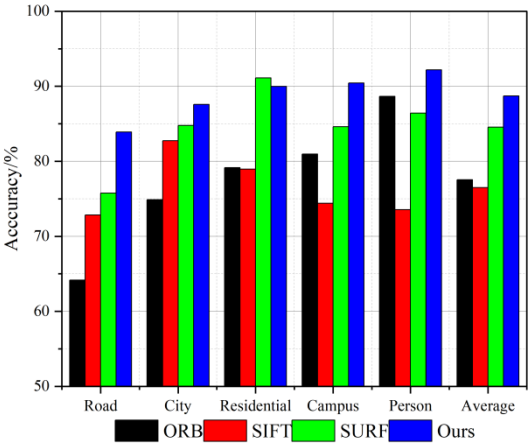

FIGURE 12. Histogram of matching accuracy for each algorithm.

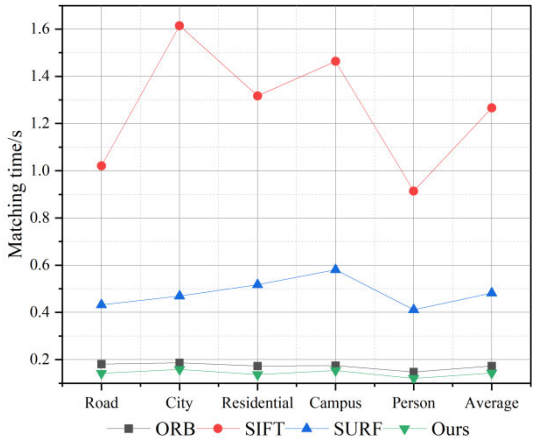

FIGURE 13. Summary of matching time for each algorithm.

experiment, based on Experiment 3.1, the matching accuracy and corresponding screening rate were calculated under different screening thresholds, and the P-A (coarse screening rate-matching accuracy) curve was plotted. The experimental results are shown in Figure 8, and the specific data is shown in Table 2. The matching error distribution of each algorithm was calculated for unmatched point pairs, as shown in Figure 9.

As can be seen from the figure, when the screening threshold is set low, the matching accuracy of several algorithms is relatively good. Among them, the ORB algorithm has the highest accuracy, but as the screening threshold increases, the

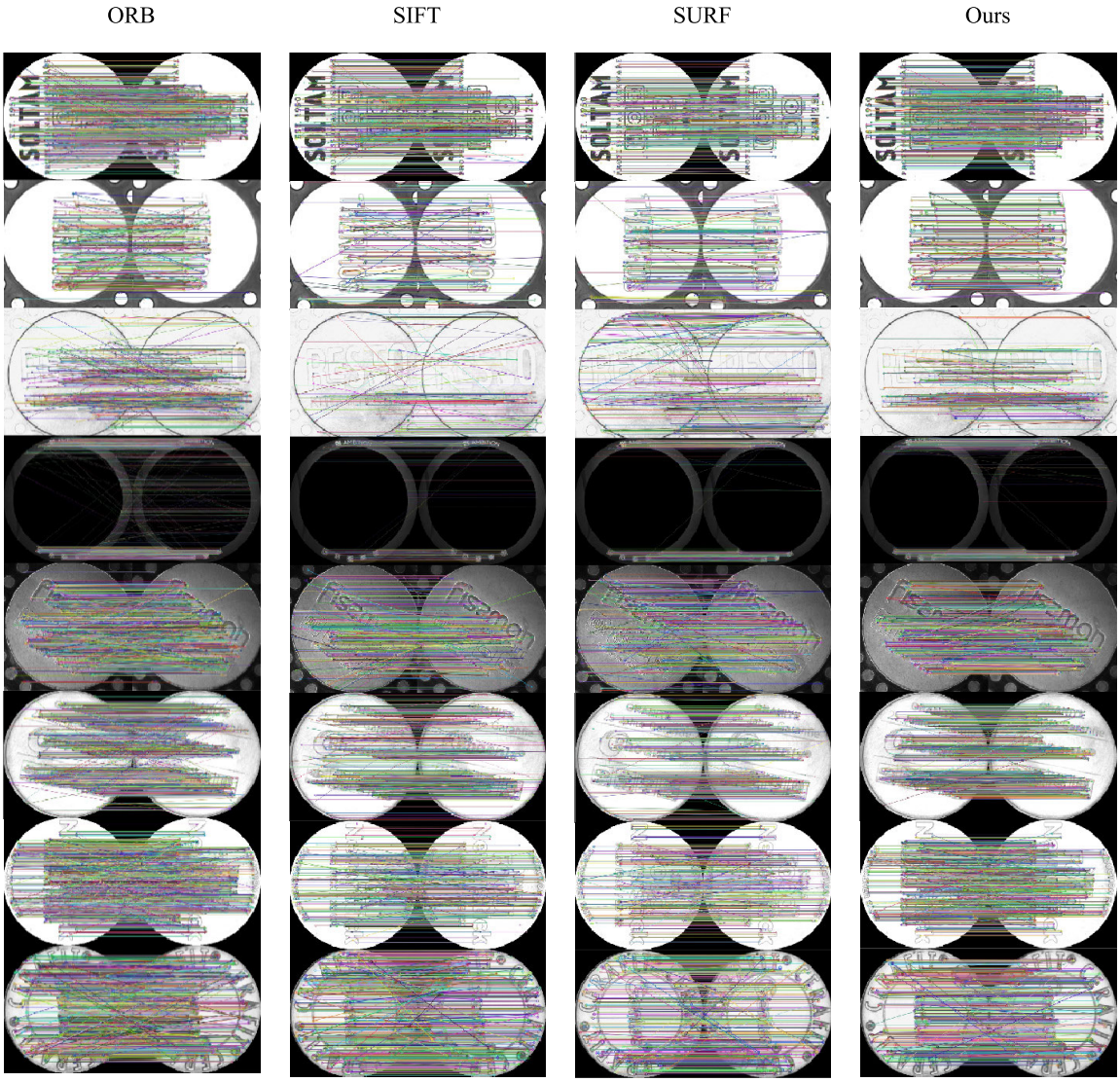

FIGURE 14. Matching results of each algorithm.

matching accuracy of the ORB, SIFT, and SURF algorithms all decrease significantly, and the decrease of the proposed algorithm is the smallest. Further analysis of the data in the figure shows that although the matching accuracy of the proposed algorithm is slightly lower than that of the other algorithms when the matching threshold is set to a lower value, it significantly exceeds the other algorithms in terms of screening rate. Combined with the data in the table, it is not difficult to find that when the matching accuracy of the ORB algorithm reaches 99.1%, the screening rate is 9.5%; the screening rates of the SURF and SIFT algorithms at the same accuracy level are only 0.5% and 4.2%, respectively. This indicates that the matching accuracy of these algorithms is achieved at the expense of sacrificing screening rate. In contrast, the proposed algorithm achieves a screening rate of 23% at a matching accuracy of 99.1%, which is significantly better than other algorithms. When the screening threshold is set to 1, that is, the screening rate is 100%, the matching accuracy can reflect the quality of feature extraction of the

TABLE 2. Experimental results.

Cof	（Matching Accuracy/%, Filtering Rate/%）			
	ORB	SIFT	SURF	Ours
0.50	(99.4,8.0)	(99.6,0.4)	(99.2,3.6)	(99.1,23.0)
0.55	(99.5,9.5)	(99.4,0.5)	(99.3,4.2)	(98.1,26.8)
0.60	(99.5,10.2)	(93.5,0.5)	(98.8,4.8)	(97.5,29.0)
0.65	(99.1,11.6)	(90.8,0.5)	(97.5,5.7)	(95.0,32.1)
0.70	(97.3,13.0)	(78.6,0.7)	(92.7,6.7)	(93.3,34.0)
0.75	(94.1,14.4)	(52.4,1.1)	(86.8,8.4)	(90.2,37.3)
0.80	(88.2,17.0)	(29.7,1.9)	(71.8,11.3)	(86.6,40.3)
0.85	(71.1,22.0)	(13.0,4.6)	(52.2,16.8)	(81.6,45.2)
0.90	(48.0,35.5)	(6.3,12.7)	(36.4,26.4)	(70.3,54.1)
0.95	(32.6,56.7)	(2.6,35.2)	(22.0,48.0)	(56.1,70.3)
1.00	(20.7,100)	(1.2, 100)	(12.0,100)	(43.3,100)
AP/%	44.74	4.65	31.44	63.63

TABLE 3. Experimental results.

	Mean Matching Error			
	Layer 2	Layer 3	Layer 4	Layer 5
Before Mapping Correction	1.72	3.99	7.84	12.68
After Mapping Correction	1.66	3.63	6.83	10.25

algorithm without relying on screening. The corresponding matching accuracy of the proposed algorithm also reaches 43.3%, which is much higher than the 20.7% of the ORB algorithm, the 1.2% of the SIFT algorithm, and the 12.0% of the SURF algorithm. The AP can reflect the comprehensive performance of the algorithm at various screening rates. The AP value of the proposed algorithm reaches 63.63%, which is 18.89% higher than that of the ORB algorithm. It can also be clearly seen from Figure 11 that the matching loss distribution of the proposed algorithm is mostly concentrated on the right side of the coordinate system, indicating that most of the matches are correct.

D. COMPARATIVE EXPERIMENT ON COORDINATE MAPPING CORRECTION

In order to validate the effectiveness of coordinate mapping correction, we recorded the coordinate information of matching points before and after mapping correction on the experimental images, and plotted the distribution of matching losses for each layer of feature points, as illustrated in Figure 10. It can be observed from the figure that the mapping effect is relatively weak for the second layer of feature points, but from the third layer onwards, the distribution of matching errors after mapping correction shows a clear rightward shift, indicating a certain reduction in matching errors after coordinate correction. The experimental results are presented and compared in Table 3.

From Table 3, it can be observed that the effect of mapping correction is most significant in the 5th layer, with a reduction in error of up to 19.2%. In contrast, the effect is least significant in the 2nd layer, with a reduction in error of only 3.5%. The error reduction in the 3rd and 4th layers is 9.0% and 12.9%, respectively.

E. COMPARATIVE EXPERIMENT ON REAL-WORLD APPLICATION SCENARIOS

To validate the performance of the algorithm in practical application scenarios, we conducted comparative tests on the matching accuracy and matching time of the algorithm in the fields of autonomous driving and industrial automation. The test images were from the KITTI dataset and the actual industrial production pot bottom label dataset. The KITTI dataset is currently the largest computer vision algorithm evaluation dataset in autonomous driving scenarios, containing real image data collected in urban, rural, and highway scenes, with various degrees of occlusion and truncation in each image. The dataset is divided into five categories:

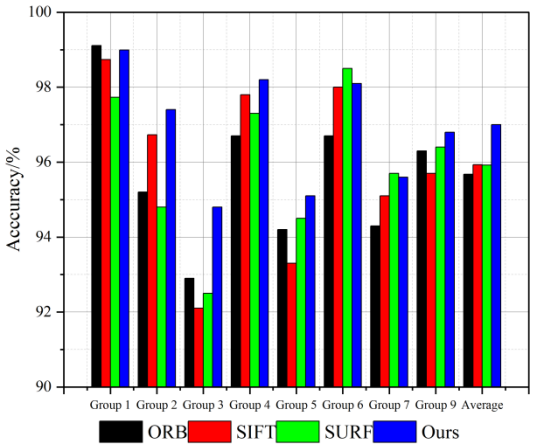

FIGURE 15. Histogram of matching accuracy summary for each algorithm.

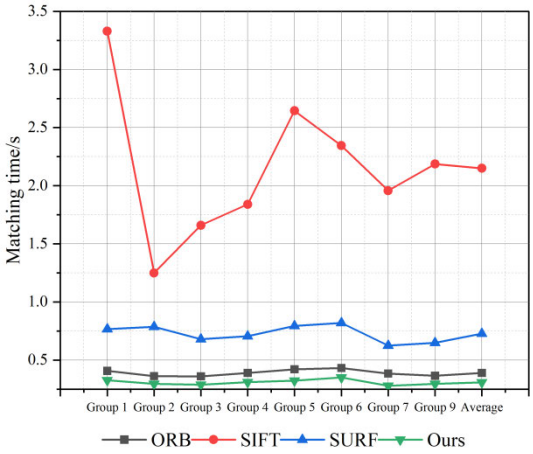

FIGURE 16. Summary of matching time for each algorithm.

'Road', 'City', 'Residential', 'Campus', and 'Person'. The pot bottom label dataset contains various pot bottom label images in actual production, with different degrees of lighting changes and surface reflections for each pot type. To ensure consistency of the parameters, we unified the screening threshold rate of each algorithm to 0.8 for the experiment. We compared the matching accuracy and matching time of each algorithm, and the experimental results are shown in Figures 11-16, and the summary results are shown in Tables 4-7.

According to the data from Table 4 and Table 5, it can be seen that our algorithm performs the best on the KITTI dataset, with a matching accuracy of 88.70% and a matching time of only 0.143 seconds. Compared to the ORB algorithm, our algorithm improves the matching accuracy by 10.15% and reduces the matching time by 17.34%. However, in the Residential category, our algorithm's matching accuracy is slightly inferior to that of the SURF algorithm. This is because there are certain and complex texture information in the image, which makes it difficult for the algorithm to accurately describe it in high-level images.

TABLE 4. Experimental sample matching accuracy (KITTI Dataset).

	Road	City	Residential	Campus	Person	Average
ORB	64.15	74.87	79.14	80.97	88.64	77.55
SIFT	72.84	82.75	78.96	74.39	73.56	76.50
SURF	75.77	84.78	91.12	84.61	86.41	84.54
Ours	83.91	87.59	89.37	90.43	92.18	88.70

TABLE 5. Matching time of experimental samples (KITTI Dataset).

	Road	City	Residential	Campus	Person	Average
ORB	0.181	0.187	0.173	0.175	0.148	0.173
SIFT	1.021	1.614	1.317	1.463	0.914	1.266
SURF	0.432	0.469	0.517	0.581	0.411	0.482
Ours	0.142	0.159	0.137	0.154	0.121	0.143

TABLE 6. Experimental sample matching accuracy (Cookware Bottom Label Dataset).

	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Average
ORB	99.11	95.23	92.94	96.68	94.22	96.74	94.28	96.25	95.68
SIFT	98.74	96.73	92.15	97.76	93.31	98.12	95.06	95.64	95.94
SURF	97.73	94.81	92.54	97.25	94.51	98.63	95.76	96.882	96.01
Ours	98.99	97.41	94.75	98.23	95.14	98.16	95.61	96.75	96.88

TABLE 7. Experimental sample matching accuracy (Cookware Bottom Label Dataset).

	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 8	Group 8	Average
ORB	0.407	0.362	0.364	0.391	0.421	0.432	0.384	0.365	0.391
SIFT	3.336	1.257	1.661	1.837	2.645	2.341	1.958	2.187	2.153
SURF	0.767	0.786	0.679	0.705	0.794	0.819	0.625	0.648	0.728
Ours	0.327	0.296	0.289	0.311	0.323	0.251	0.279	0.295	0.296

According to the data from Table 6 and Table 7, our algorithm performs well on the pot bottom label dataset. In terms of real-time performance, our algorithm's average speed is about 2.5 times faster than that of the SURF algorithm, and it only takes 75.70% of the time of the ORB algorithm. In terms of accuracy, our algorithm's average matching accuracy on each group of samples is 96.88%, which is better than the ORB algorithm's 95.68%, the SIFT algorithm's 95.94%, and the SURF algorithm's 96.13%.

IV. CONCLUSION

This paper proposes an improved feature point extraction algorithm based on multi-scale feature fusion to address the low accuracy and poor robustness issues of traditional ORB algorithm. In the establishment of feature scale pyramid, the algorithm extracts features at each layer of the image to achieve multi-scale feature extraction. Furthermore, in the fusion of feature coordinate information, the algorithm corrects the mapped coordinates by establishing FAST-SCORE maps at different scales to alleviate the accuracy degradation caused by downsampling, thus improving the accuracy and robustness of feature extraction.

Comparative experiments were conducted on ORB algorithm, SIFT algorithm, SURF algorithm, and the proposed algorithm. The results show that the proposed algorithm has significantly better resistance to interference and has significant advantages in matching time and matching accuracy in industrial practical applications.

However, it was found in the experiments that the proposed algorithm has certain requirements for the scale of the image, and the advantages of multi-scale feature extraction can only be fully utilized in larger images. Therefore, in further optimizing the algorithm, particle swarm optimization (PSO) can be used to determine the position of feature points, making the distribution of feature points more uniform and reducing the dependence on image size, thus further improving the feature extraction capability of the algorithm.

REFERENCES

[1] B. Liu, F. Yang, Y. Huang, Y. Zhang, and G. Wu, "Single-shot three-dimensional reconstruction using grid pattern-based structured-light vision method," *Appl. Sci.*, vol. 12, no. 20, Oct. 2022, Art. no. 10602.

[2] S. Quan, K. Yin, K. Ye, and K. Nan, "Robust feature matching for 3D point clouds with progressive consistency voting," *Sensors*, vol. 22, no. 20, p. 7718, Oct. 2022.

[3] I. Misra, M. K. Rohil, S. M. Moorthi, and D. Dhar, "A novel country-level integrated image mosaic system using optical remote sensing imagery," *Earth Sci. Informat.*, vol. 15, no. 4, pp. 2181–2193, Dec. 2022.

[4] J. Ni, X. Wang, T. Gong, and Y. Xie, "An improved adaptive ORB-SLAM method for monocular vision robot under dynamic environments," *Int. J. Mach. Learn. Cybern.*, vol. 13, no. 12, pp. 3821–3836, Dec. 2022.

[5] J. Ma, A. Fan, X. Jiang, and G. Xiao, "Feature matching via motion-consistency driven probabilistic graphical model," *Int. J. Comput. Vis.*, vol. 130, no. 9, pp. 2249–2264, Sep. 2022.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[7] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 3951, Jul. 2006, pp. 404–417.
[8] 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.
[9] J. Gao and Z. Sun, "An improved ASIFT image feature matching algorithm based on POS information," *Sensors*, vol. 22, no. 20, p. 7749, Oct. 2022.
[10] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[11] 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.
[12] M. Wu, "Research on optimization of image fast feature point matching algorithm," *EURASIP J. Image Video Process.*, vol. 2018, no. 1, pp. 1–27, Dec. 2018.
[13] C. Lv, J. Li, Q. Kou, H. Zhuang, and S. Tang, "Stereo matching algorithm based on HSV color space and improved census transform," *Math. Problems Eng.*, vol. 2021, pp. 1–17, Jul. 2021.
[14] C. Ma, X. Hu, J. Xiao, H. Du, and G. Zhang, "Improved ORB algorithm using three-patch method and local gray difference," *Sensors*, vol. 20, no. 4, p. 975, Feb. 2020.
[15] Z. Yan, H. Wang, Q. Ning, and Y. Lu, "Robust image matching based on image feature and depth information fusion," *Machines*, vol. 10, no. 6, p. 456, Jun. 2022.
[16] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, vol. 2, Feb. 2009, pp. 331–340.
[17] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2227–2240, Nov. 2014.
[18] J. Bian, W. Lin, Y. Matsushita, S. Yeung, T. Nguyen, and M. Cheng, "GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2828–2837.
[19] S. Kumar, N. Kumar, and K. Alam, "A nonlinear anisotropic diffusion equation for image restoration with forward-backward diffusivities," *Recent Adv. Electr. Electron. Eng. (Formerly Recent Patents Electr. Electron. Eng.)*, vol. 14, no. 4, pp. 428–434, Jun. 2021.
[20] R. Taranco, J.-M. Arnau, and A. González, "LOCATOR: Low-power ORB accelerator for autonomous cars," *J. Parallel Distrib. Comput.*, vol. 174, pp. 32–45, Apr. 2023.
[21] S. Li, Q. Wang, and J. Li, "Improved ORB matching algorithm based on adaptive threshold," *J. Phys., Conf. Ser.*, vol. 1871, no. 1, Apr. 2021, Art. no. 012151.
[22] 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.
[23] G. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3974–3983.
[24] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

CHENGXIAN YAO received the bachelor's and master's degrees in engineering from the Shanghai University of Engineering Science. He is currently focused on developing feature matching algorithms to improve image processing and feature matching performance in complex environments, with applications in industrial automation and autonomous driving. His research interests include machine vision, image processing, and feature matching.

HAIFENG ZHANG received the bachelor's degree in computer science and technology and the master's degree in computer application technology from Chang'an University, in 2000 and 2004, respectively, and the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, in 2007. From 2017 to 2018, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, Kansas State University. He is currently an Associate Professor with the School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, specializes in machine vision technology and applications, and image processing. He has led several research projects, including visual inspection of brake joint appearance, design of an automatic internal thread appearance inspection and control systems, and development of a visual inspection and control system for plug-in quick installation plates. His research results have been published in various domestic and international journals.

JIA ZHU received the B.E. degree from the Shanghai University of Engineering Science, in 2020, where she is currently pursuing the master's degree. Her research interests include the development of algorithms and software for computer vision, image processing, and measurement and control systems. Her research aims to improve the accuracy and efficiency of computer vision and image processing systems, with applications in fields such as robotics, automation, and medical imaging.

DIQING FAN received the master's degree in mechanical design and theory from Shanghai Ocean University, in 2006. From 1998 to 2003, he was a Technical and Quality Supervisor with companies, such as Zhuhai Dongda Group Company Ltd. and Shanghai Science Atlanta Company Ltd. As the project leader, he completed the "Automatic Detection Equipment for Square Joint Sealing Surface," applied for five invention patents, authorized one patent, and authored one textbook. He is skilled in the development of automation assembly, processing and testing equipment, and intelligent equipment. His research interests include mechanical and electrical control systems and engineering, and machine vision.

YU FANG received the Ph.D. degree in mechanical manufacturing and automation from the Harbin Institute of Technology, in 2006. He conducted Postdoctoral Research with the School of Mechanical Engineering, Donghua University, in 2006. Since 2008, he has been with the Shanghai University of Engineering Science, where he is currently the Vice Dean of the Urban Rail Transit College, the Director of the Academic Affairs Office, the Research Office, and the Technology Transfer Center, and the Dean of the School of Mechanical and Automotive Engineering. His main research interests include special robots and intelligent equipment, machine vision, and image detection.

LIN TANG is currently pursuing the Bachelor of Engineer degree from Xi'an Jiaotong University. Her current major is automation, and her research interests include computer vision, image processing, machine learning and so on.