IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 3, 2020, accepted February 20, 2020, date of publication March 2, 2020, date of current version March 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2977841

Feature Point Extraction and Tracking Based on a Local Adaptive Threshold

HANG LI, HONGFAN YANG, AND KAIYANG CHEN
School of Mechatronics Engineering, Henan University of Science and Technology, Luoyang 471003, China

Corresponding author: Hongfan Yang (170317010060@ stu.haust.edu.cn)

This work was supported in part by the National Key Research and Development Program under Grant 2018YFB200502, in part by the Key Science and Technology Program of Henan Province under Grant 182102110420, and in part by the National Basic Scientific Research Project of China under Grant JCKY2019419D001.

ABSTRACT Navigation, environment perception and localization are important capabilities of intelligent vehicles. In this paper, environmental perception and localization from binocular vision are studied. First, an outdoor feature point extraction algorithm that uses a local adaptive threshold is proposed to acquire environmental information. The algorithm filters feature points by setting adaptive parameters and calculating each pixel threshold with a dynamic local threshold. Second, an accurate method for feature point tracking is proposed for localization. We present exhaustive evaluation in 4 major scenarios from the most popular datasets. Evaluating the proposed method with traditional and state-of-the-art extraction methods and experimental results demonstrates that when the brightness decreases or increases, the performance of the proposed method is stable in terms of the number of feature points, the calculation speed and the overall repetition rate. Our proposed tracking method outperforms state-of-the-art tracking methods in terms of the root mean square error (RMSE) and the errors in the dimensions in the scenarios.

INDEX TERMS Feature extraction, tracking, oriented FAST and rotated BRIEF(ORB), features from accelerated segment test (FAST), root mean squared error (RMSE).

I. INTRODUCTION

With the rapid development of the economy and the transportation system, intelligent vehicle technology, as an important part of the intelligent transportation system (ITS), has been widely studied [1], [2]. This technology is regarded as the key for solving traffic problems. Navigation is a research hotspot in the field of intelligent vehicles. Environmental perception and localization are important foundations for realizing accurate navigation [3]. Visual sensors, as the most similar tools to human environment perception [4], have substantial advantages over other sensors. According to the number of visual sensors, visual navigation can be divided into three types: monocular vision, binocular vision and multi-vision [5]–[8]. Binocular vision requires only two cameras for measuring the depth information of the image, as in the human visual system. In the field of unmanned vehicles with high cost-control, cost-effectiveness and real-time requirements, binocular vision navigation is a superior choice [9], [10]. Searching for image features is an important part of visual environment perception. Image features are divided into point features, line features and surface features [11]. Due to the unique advantage of point features that they remain unchanged when the camera angle of view changes slightly [12], they have attracted substantial attention. Our work is about improving the performance of feature points and tracking them.

The most commonly used feature point extraction algorithms are scale-invariant feature transform (SIFT) [13], speeded-up robust features (SURF) [14], [15], oriented features from accelerated segment test (FAST) and rotated binary robust independent elementary features (BRIEF) (ORB) [16] and convolution neural networks (CNNs) [17]. In [18], the distances from the environmental feature points to the origins of the camera coordinate system were calculated by matching the captured binocular image frames using SURF to obtain the relative vehicle locations. In [19], the SIFT algorithm was used to match the feature points of two images. It selects three corresponding feature points and uses them to construct a triangle. Then, the transformation relationship of two corresponding triangles is utilized to construct a mathematical model for identifying

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

the direction of motion. The SIFT and SURF algorithms are relatively mature; however, they cannot satisfy the higher requirements in terms of the runtime and real-time performance.

Deep learning methods have achieved remarkable results in the research field of image point feature extraction and tracking, benefiting from the powerful feature characterization of deep learning. Feature points are detected in an image, and the region around the feature point is given as input to a CNN to obtain the feature vector. In large-scale visual recognition experimentation, CNNs have shown exceptional performance with VGGNet [20] and GoogLeNet [21]. These networks have been proven to be very discriminative feature descriptors. Kavitha and Thirumala [22] used a deep CNN, AlexNet, for feature extraction in the first stage of image registration instead of handcrafted features. Cieslewski *et al.* [23] designed a feature point extraction method for point clouds, called a neighbor-binary landmark density descriptor (NBLD), and extracted the NBLD from detected keypoints to recognize places through a voting framework. The loss of key information and long-term memory content during the model reconstruction process becomes serious when the upsampling process is nonlinear mapping, as this causes the effect of superresolution to be reduced by the deep network. Most CNN applications are for specific scenes, such as underwater images, medical images, satellite images and images for facial recognition. The deep network model after learning has some limitations. If the convolutional neural network is shallow, the details of feature extraction will be lost for different kinds of scenes; if the convolutional neural network is deep, the computational complexity will increase, which will bring some problems such as nonreal-time performance.

The ORB algorithm reduces the accuracy and robustness to increase the computing speed and reduce the computing time, namely, it realizes a trade-off between quality and performance [24]. Several studies have utilized this algorithm and have focused on ORB feature point extraction. An ORB feature-based tracking method in real-time, called ORB-SLAM, was proposed in [25]. The median filter and RANSAC algorithms were utilized for extracting feature points and finding more accurate matching points in [26]. Qin *et al.* [27] proposed an improved ORB algorithm based on the SIFT algorithm. Huang *et al.* [28] proposed an architecture that considers the reuse of subimage data; a remainder-based method is firstly designed for reading the subimage, and a FAST detector and a BRIEF descriptor are combined for corner detection and matching. Dai *et al.* [29] detected features by the SURB (SURF-ORB) algorithm, which combines the ORB algorithm with the SURF algorithm. However, in the environment perception of an outdoor scene, the number and repetition rate of feature points that are extracted using a fixed threshold via the ORB feature extraction algorithm change sharply with the brightness, thereby resulting in many mismatched feature points. This phenomenon directly affects the tracking of feature points, which results in large errors in localization and inaccurate navigation.

In this paper, feature point extraction and tracking from binocular vision are studied. First, in Section II, the original ORB algorithm is improved by setting adaptive parameters to realize the precise and stable extraction of image feature points under varying brightness values. Second, an image feature point tracking method is introduced in Section III, in which four images are needed in a complete feature point tracking link to accurately obtain the three-dimensional coordinates of the tracking feature points at various times. Finally, in Section IV, we provide experimental evidence of the superior accuracy of the improved feature point-based method for feature point extraction and tracking on vehicle platforms and in popular KITTI datasets.

II. FEATURE POINTS EXTRACTION

A. ORIGINAL ORB METHOD

Since the original feature from accelerated segment test (FAST) algorithm does not consider directionality, a description of the rotation is added to the ORB algorithm. The realization method assigns directions to the FAST points to convert them into directional FAST points. The calculation of the directionality is based on the center of gravity method. The implementation steps are as follows: In a small image block S, the moment of the image block is defined as:

$$m_{pq} = \sum_{x,y \in S} x^p y^q F(x, y), \quad p, q = \{0, 1\} \tag{1}$$

where $F(x, y)$ is the intensity of point (x, y). $\boldsymbol{m_{ij}}$ is the $(i+j)$th moment matrix of the image, and the intensity centroid can be calculated from these moments as:

$$C = \left(\frac{\mathbf{m}_{10}}{\mathbf{m}_{00}}, \frac{\mathbf{m}_{01}}{\mathbf{m}_{00}} \right) \tag{2}$$

where $\boldsymbol{m_{00}}$ is a null matrix because for binary images, the null matrix represents its area, and $\boldsymbol{m_{10}}$ and $\boldsymbol{m_{01}}$ are 1×1 matrices. The ORB algorithm constructs a vector from the feature point to the centroid. The orientation angle of the feature point is:

$$\alpha = \arctan(\mathbf{m}_{01}/\mathbf{m}_{10}) \tag{3}$$

Therefore, the orientation angle of each ORB feature point can be represented by the angle between the centroid and the feature point.

The ORB algorithm uses the BRIEF descriptor to describe the feature points. The descriptor can be calculated when the orientation angle of each feature point is obtained. The rotation-invariant descriptor of ORB is a bit string description of an image patch constructed via a set of binary intensity tests. Consider a smoothed image patch p of a feature point. A binary test is defined as:

$$\tau = (p; x, y) = \begin{cases} 1, & p(x) < p(y) \\ 0, & p(x) \geq p(y) \end{cases} \tag{4}$$

where $p(x)$ and $p(y)$ denote the gray values of random points x and y, respectively. After n test point pairs have been selected,

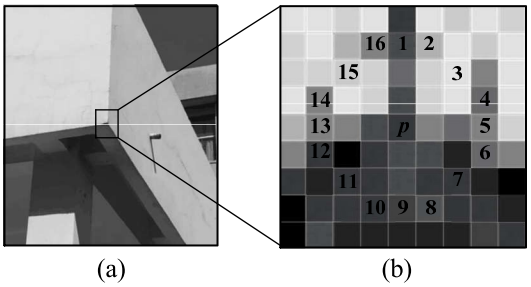

FIGURE 1. FAST detector. (a) A candidate corner. (b) The circular window near the enlarged candidate corner.

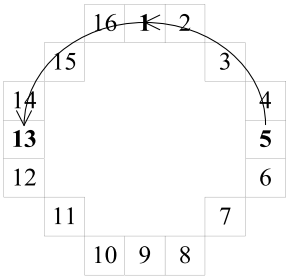

FIGURE 2. FAST detector strategy when Ip_1, Ip_5, and Ip_{13} belong to the darker or brighter group.

the descriptor is defined as a binary code list:

$$f_n(p; x, y) = \sum_{i=1}^{n} 2^{i-1} \tau(p; x, y) \tag{5}$$

The descriptor in formula (5) is not rotation-invariant. To overcome this problem, a $2\times n$ matrix is defined for any point (x_i, y_i) that participates in the binary tests:

$$\boldsymbol{Q} = \begin{pmatrix} x_1, \cdots, x_n \\ y_1, \cdots, y_n \end{pmatrix} \tag{6}$$

According to the feature point orientation angle α and the corresponding rotation matrix $\boldsymbol{R}_\alpha$, $\boldsymbol{S}_\alpha$ which is the test point set including the orientation property can be calculated:

$$\boldsymbol{S}_\alpha = \boldsymbol{R}_\alpha \boldsymbol{Q} \tag{7}$$

Then, the rotation-invariant feature point descriptor is represented as:

$$g_n(p, q) = f_n(p) | (x_i, y_i) \in \boldsymbol{S}_\alpha \tag{8}$$

B. ORB FEATURE POINT EXTRACTION METHOD BASED ON THE IMPROVED FAST ALGORITHM

The FAST detector test criterion analyzes a circle of sixteen pixels around the candidate corner p, as illustrated in Fig. 1. Fig. 1a shows a representative candidate corner on an image. Because the candidate corner is located at an edge and a real corner, intuitively, its curvature is sufficiently high and the curvature changes substantially. Hence, it is selected for analysis and explanation; however, the candidate corner is not necessarily a real corner. A circular window near the enlarged candidate corner is shown in Fig. 1b. The intensity of the candidate corner is I_p. We analyze 16 corners on the circumference of the circle centered at p with a radius of 3 pixels.

The FAST detector is expressed as:

$$S_{pn} = \begin{cases} d, & I_{pn} \le I_p - t \\ s, & I_p - t < I_{pn} < I_p + t \\ b, & I_p + t \le I_{pn} \end{cases} \tag{9}$$

where I_{pn}(n $= 1, 2, 3, \ldots, 16$) is the intensity of pixel n around the corner and t is the fixed threshold. If I_{pn} is equal to d, the pixel belongs to the darker group; if I_{pn} is equal to s, the pixel belongs to the similar group; and if I_{pn} is equal to b, the pixel belongs to the brighter group. If there are 9 continuous pixels that belong to the darker or brighter group, p is regarded as a corner (FAST-9 or FAST-12). This method must detect the intensities of at least 9 pixels continuously. In practice, to detect 9 continuous pixel intensities, it is necessary to detect them from pixels 1 to 9, 2 to 10,…, or 8 to 16. Therefore, 9 pixels must be calculated in the best-case scenario, and 54 pixels must be calculated in the worst-case scenario, which is a large number of calculations.

To reduce the computational complexity, I_{p1}, I_{p5}, I_{p9} and I_{p13} can be detected directly (I_{p2}, I_{p6}, I_{p10}, I_{p14}, etc., can also be selected, with an interval of three pixels). When three of the four selected pixels belong to either the dark or bright class, p may be a real corner; otherwise, it will be excluded directly. Next, the intensities of the remaining six pixels in the three-pixel enclosure ring are detected. If the remaining six pixels belong to the dark or bright class, p is a real corner.

The FAST corner detection diagram is shown in Fig. 2. As shown in Fig. 2, for example, when I_{p1}, I_{p5}, and I_{p13} belong to the darker or brighter group, then I_{p2}, I_{p3}, I_{p4}, I_{p14}, I_{p15}, and I_{p16} are detected continuously. If these remaining 6 pixels belong to the darker or brighter group, then p is a real corner.

The threshold in FAST extraction is artificially set to a percentage of the brightness I. With the changes in illumination and contrast in an outdoor environment, the feature points of an image will suffer from extraction errors. It is difficult to obtain ideal results using fixed global or fixed local thresholds. Instead, dynamic local thresholds can be used: a distinct threshold is set for each pixel in the image via autothreshold segmentation. The selection criterion for defining the threshold t for each pixel p is as follows:

$$t = \delta \times \left(\sum_{n=1}^{16} I_i - I_{\max} - I_{\min} \right) / I_a \tag{10}$$

where I_{max} is the maximum pixel intensity, I_{min} is the minimum pixel intensity on the circumference, I_a is the average intensity of the remaining 14 pixels after removing I_{max} and I_{min}, and δ is the adaptive parameter. Although the adaptive parameters δ are set artificially and fixed, t is a dynamic local threshold because I_{max}, I_{min} and I_a are constantly changing.

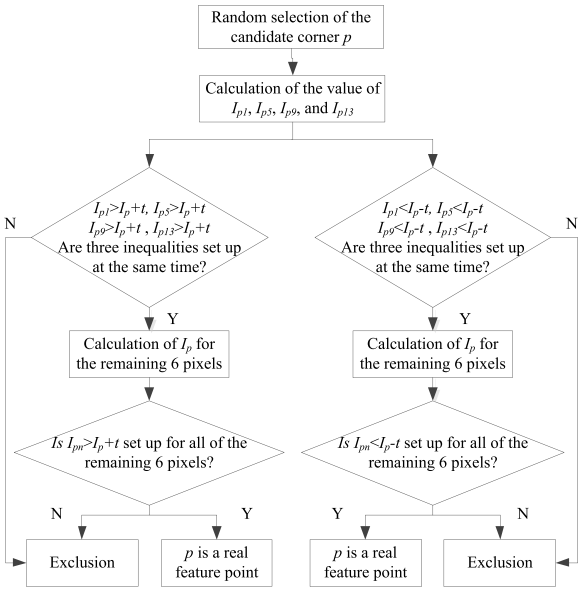

FIGURE 3. ORB feature point extraction algorithm procedure that is based on improved FAST.

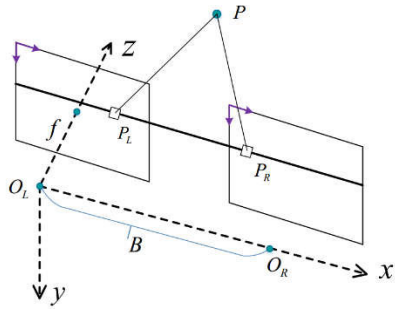

FIGURE 4. Imaging model of the binocular camera.

Fig. 3 illustrates the ORB feature point extraction algorithm procedure based on improved FAST as a flow chart.

III. BINOCULAR FEATURE POINTS TRACKING

A. HORIZONTAL BINOCULAR CAMERA MODEL

The mathematical model of a binocular camera can be regarded as two pinhole camera models, as shown in Fig. 4. The camera coordinate system of a binocular camera is defined as follows: take the optical center O_L of the left camera as the origin of the coordinate system, establish the X-axis horizontally to the right. The left camera establishes the Z-axis along the optical axis, and establish the Y-axis vertically down; B is the baseline of the binocular camera, which is the physical distance between the left and right optical centers.

The world coordinate system is an imaginary coordinate system, which can be defined freely as needed. In the binocular measurement system, the world coordinate system is generally defined to coincide with the camera coordinate system of the left camera.

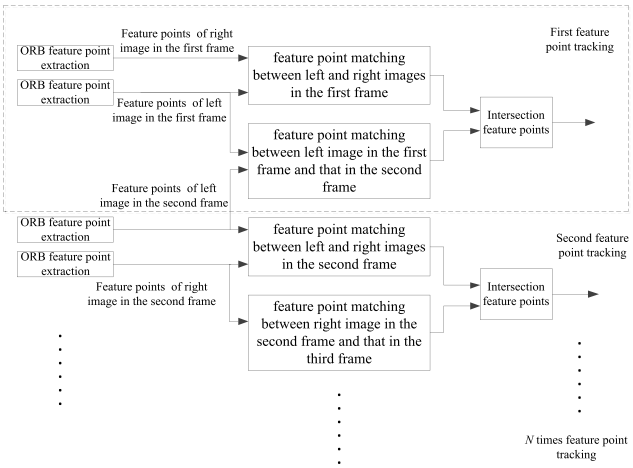

FIGURE 5. Binocular image feature point tracking flowchart.

Let the three-dimensional point of the binocular camera coordinate system be P, and its coordinate be (X, Y, Z). In the left and right cameras, the imaging points are P_L and P_R respectively, and the corresponding pixel coordinates are: (u_t, v_t) and (u_r, v_r), respectively. According to the principle of similar triangle and pinhole models, the following formula can be obtained:

$$\begin{cases} Z = \dfrac{fB}{d}, \quad d = u_L - u_R \\ X = \dfrac{u_t - c_x}{f_x} Z \\ Y = \dfrac{v_t - c_y}{f_y} Z \end{cases} \tag{11}$$

where f is the focal length, d is the parallax of P for binocular imaging, and the main point c is the intersection of the optical axis and the imaging plane, whose coordinates are (c_x, c_y).

B. FEATURE POINTS TRACKING METHOD

Image feature point tracking is the tracking of the feature points that correspond to the same spatial object point in the image sequence at various times. It is closely related to feature point matching. Feature point matching is the matching of feature points on two images, whereas tracking is the matching of feature points on multiple image sequences. Feature point tracking is conducted to prepare for the next step of motion parameter estimation. Feature point tracking of one link involves completing two feature point matching steps: (1) feature point matching between the left image in the first frame and that in the second frame captured by the binocular camera and (2) feature point matching between the left and right images in the first frame. Similarly, the second feature point tracking step is the same as the above method until the nth tracking is completed.

Fig. 5 presents a flow chart of the ORB feature point tracking method. The tracking procedure consists of the following steps: (1) The ORB feature points of the left and right images that are captured in the first frame are extracted, and matching is conducted. The coordinates of the matched feature points

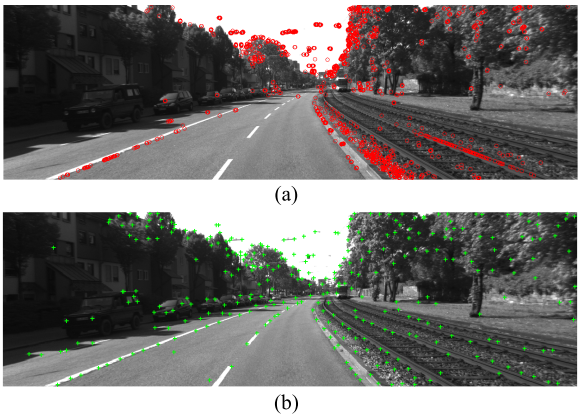

FIGURE 6. ORB feature point extraction performance in city areas. (a) The original ORB algorithm. (b) The proposed improved ORB algorithm.

in the left and right images are placed into corresponding two-dimensional arrays. The arrays of the coordinates for storing the matched feature point pairs are (u_t, v_t) and (u_r, v_r); (2) feature point matching between the left image in the first frame and that in the second frame is conducted. The array for the left-image coordinates of the matched feature point pairs that are stored is $(u_t^{'}, v_t^{'})$ in the first frame and $(u_r^{'}, v_r^{'})$ in the second frame; (3) $(u_t^{''}, v_t^{''})$ is formed by calculating the intersection of (u_t, v_t) and $(u_t^{'}, v_t^{'})$. The feature points in this new array are the tracked feature points. By substituting the pixel coordinates $(u_t^{''}, v_t^{''})$ into formula 11, the three-dimensional coordinates of the spatial points, which are regarded as coordinates of the vehicle in the world coordinate system, can be calculated.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. FEATURE POINT EXTRACTION EXPERIMENT

The extraction algorithm was implemented using MATLAB R2014a on a standard PC with Windows 7. The simulation tests were conducted on the Karlsruhe Institute of Technology and Toyota University of Technology (KITTI) datasets. In this experiment, 4 major scenarios, including city, residential, road, and campus scenarios, are tested.

According to a previous experiment, if the adaptive parameter δ is 10%∼30% in the improved ORB algorithm, the number of feature points will be large, and overlapping and aggregation will occur. If the number of feature points is less than 10% or more than 30%, the judgment conditions are too large or too small, respectively. Although there is no overlapping or aggregation of feature points, the number of feature points will be too small to accurately reflect the whole image. Therefore, in the following experiments, the default value of the adaptive parameter is set to $\delta = 20\%$. To enhance the contrast effect, nonmaximum suppression [30] is not utilized. First, we conduct experiments on city areas, and the result is presented in Fig. 6, which shows the difference in the feature point extraction performances of the improved algorithm proposed in this paper and the original ORB algorithm.

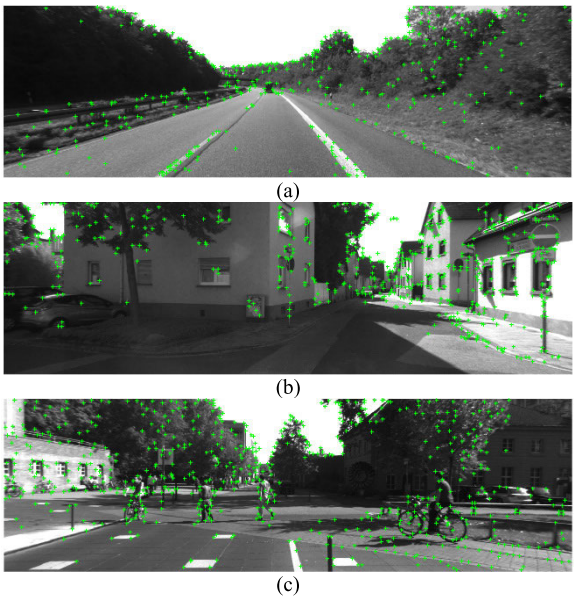

FIGURE 7. The proposed improved ORB feature point extraction performance in major scenarios from the raw KITTI dataset. (a) Road area. (b) Residential area. (c)Campus area.

According to Fig. 6a, the original ORB algorithm results in many overlapping feature points, and the feature points are clustered extensively. According to Fig. 6b, the number of feature points that are extracted by the improved ORB algorithm is substantially reduced, there are almost no overlapping feature points, and the feature points are evenly distributed, which inhibits the clustering of the feature points.

Then we tested our method in the remaining three scenarios as shown in Fig. 7. It can be observed that there are no overlapping feature points in each of these three scenarios. In Fig. 7a, lane lines, edge lines and distant cars can be detected. In Fig. 7b, the feature points of houses and vehicles in residential areas can be detected evenly. In Fig. 7c, pedestrians, vehicles and lane lines can be detected in the campus environment.

To further evaluate the adaptability of the improved ORB algorithm to variations in brightness, the extraction results are examined under gradual brightness variations of 20%, 40%, and 60% of the original brightness, as shown in Fig. 8 and Fig. 9.

According to Fig. 8, the number of ORB feature points extracted by the original ORB algorithm decreases sharply and many points overlap, whereas the number of feature points that are extracted by the improved ORB algorithm in Fig. 9 does not decrease sharply, nor are there many overlaps. Combined with the extraction results under the initial brightness, which are presented in Fig. 6b and Fig. 8, it is concluded that the number of ORB feature points that are extracted by our improved method increases slightly after the brightness increases or decreases by 20%. Therefore, the detection range of our method is enlarged, the distribution of the feature points is uniform, and no overlaps occur under small brightness changes.

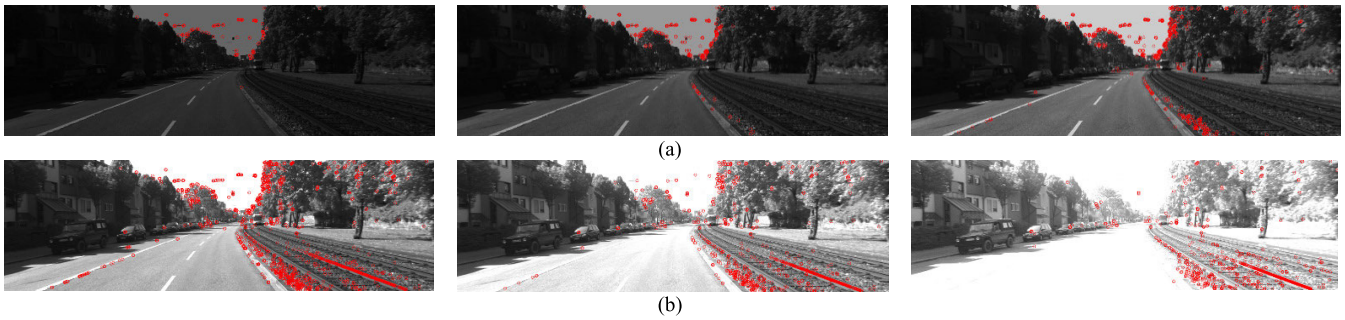

FIGURE 8. Extraction performance of the original ORB algorithm under varying brightness levels. (a) The brightness has been decreased by 60%, 40%, and 20%. (b) The brightness has been increased by 20%, 40%, and 60%.

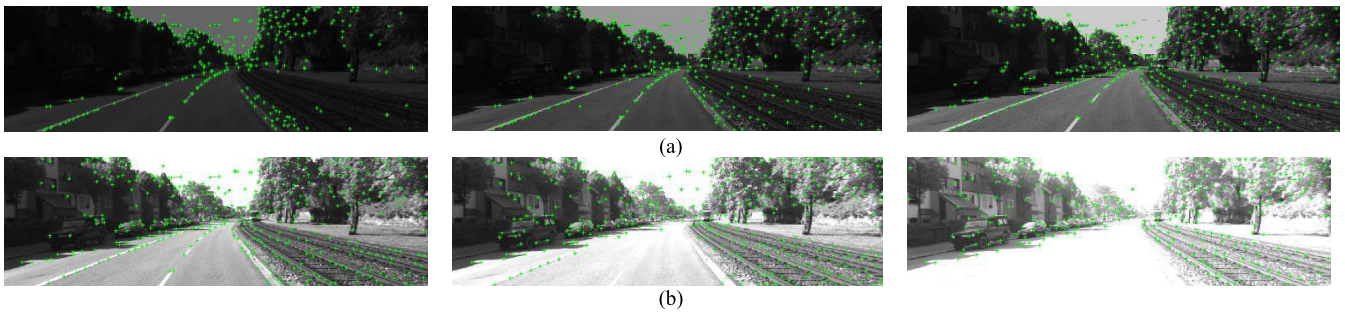

FIGURE 9. Extraction performance of the proposed ORB algorithm under varying brightness levels. (a) The brightness has been decreased by 60%, 40%, and 20%. (b) The brightness has been increased by 20%, 40%, and 60%.

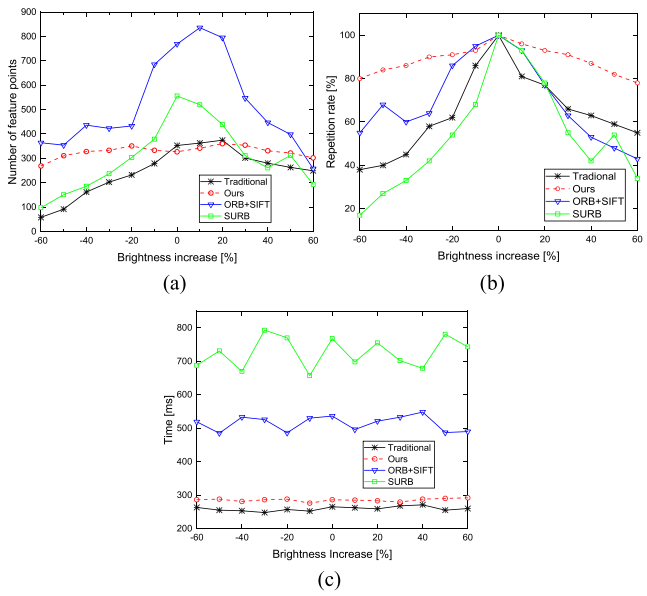

FIGURE 10. Extraction performance under varying brightness levels in the city area. (a) The number of feature points. (b) The repetition rates. (c) The computation times.

Based on the brightness of the original image, the numbers of feature points that are extracted by the two algorithms gradually increase and decrease by 10%, 20%, 30%, 40%, 50% and 60%, as shown in Fig. 10a. According to Fig. 10a, the number of feature points extracted by the original algorithm decreases dramatically with the change in brightness, whereas the number of feature points extracted by the improved algorithm becomes more stable. The maximum and minimum numbers of feature points extracted by the improved algorithm are 360 and 269, respectively. The range is only 91, which only accounts for 27.8% of the number of feature points extracted under the original brightness. The maximum and minimum numbers of feature points extracted by the original algorithm are 375 and 58, respectively. The range is 317, which accounts for 89.8% of the number of feature points extracted under the original brightness. Although the number of feature points extracted by Qin's method is higher than that of the algorithm proposed in this paper, its fluctuation range is large. After the brightness increases by 60%, the number of feature points decreases by 69% compared with the original brightness; Dai's method performs well under the original brightness and increases/decreases by 20%, but as the brightness continues to change, the number of feature points extracted drops sharply.

To further evaluate the performance of the improved algorithm, the extraction time t, and repetition rate r are selected as factors for quantifying the performance.

The repetition rate is defined as the percentage of the feature points that overlap between in the original image and the image after changing the brightness relative to the number of feature points in the original image:

$$r = N_r/N_f \times 100\% \tag{12}$$

where N_r represents the number of feature points that overlap between the original image and the image after changing the

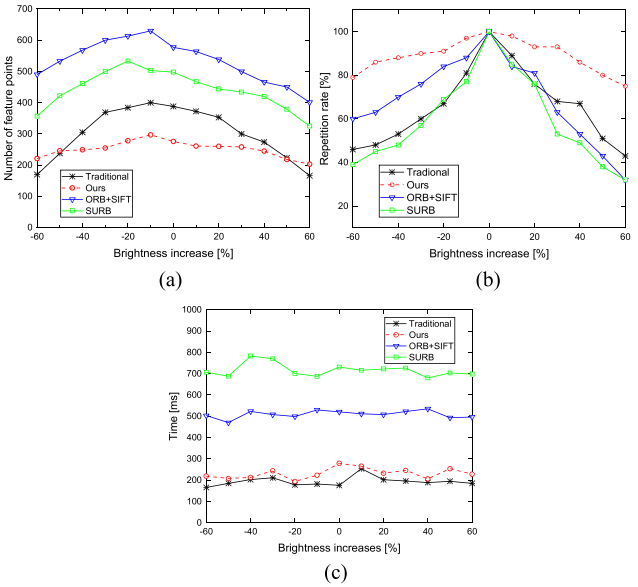

FIGURE 11. Extraction performance under varying brightness levels in the road area. (a) The number of feature points. (b) The repetition rates. (c) The computation times.

brightness and N_f represents the number of feature points in the original image.

The repetition rate point-line diagram for image brightness changes is shown in Fig. 10b. According to Fig. 9b, the repetition rate of the four algorithms is 100% in the image with the original brightness; however, with the decrease in brightness by 60% and the increase in brightness by 60%, the repetition rate of the original algorithm decreases sharply to 38% and 55%, respectively. Hence, the feature points extracted from the original image cannot be accurately detected in the image after the brightness changes. Although the repetition rate for the improved algorithm decreases after the brightness decreases or increases, it decreases by only 20%, and the repetition rate remains above 80%. In summary, the improved algorithm outperforms the original algorithm in terms of stability and adaptability to brightness changes.

The extraction time point-line diagram for image brightness changes is shown in Fig. 10c. The extraction time of the improved algorithm is between 276 ms and 292 ms, which is longer than that of the original algorithm; however, the increase is not more than 10% compared to the original algorithm. The extraction times of Qin's and Dai's methods are longer by 400 ms and 600 ms, respectively. Therefore, the extraction time of the improved algorithm is still far shorter than those of ORB+SIFT and SURB, which can satisfy the real-time requirements of the system.

Fig. 11 shows the qualitative comparisons of our method and three other methods with brightness changes in the road area. The growth trend in the three factors, which are the number of feature points, extraction time, and repetition rate, is similar to those in the city area. Due to the low complexity of the road area, which usually consists of only lane lines and trees, the growth in the road area seems to be more gradual, as there are no sharp changes in the diagrams.

FIGURE 12. Intelligent vehicle platform.

In these experiments, we have shown that our method can extract feature points in a relatively short amount of time, and the number and repetition rate are not easily affected by brightness changes in both the city and road areas, which could be useful to provide accurate information for feature point tracking in the next step.

B. BINOCULAR TRACKING EXPERIMENT

1) EXPERIMENTAL METHOD

The platform of an outdoor vehicle is shown in Fig. 12. The vehicle is a four-wheeled electric vehicle equipped with a binocular vision system and a 3D radar. The electric power-steering system, the main drive system and the control system of the electric vehicle have been modified. The binocular vision system, namely, model HNY-CV-002 by FpgaLena Co., Ltd., was mounted on the front of the vehicle, and the 3D radar was mounted on the top of the vehicle. This experiment used only the binocular vision system. A campus was selected as the experimental environment, and the collected scenery was diverse, which is conductive to the extraction of image features. After calibrating the camera via Zhang's camera calibration method in the MATLAB Camera Calibration Toolbox, the following procedure was conducted:

(a) The mobile robot was manually driven along the reference path. A video along the path was recorded by the binocular vision system, and a sequence containing 65 frames was collected. Each frame included one image captured by the left and right cameras at the same time.

(b) A personal computer (PC) was utilized to extract the feature points and to select an ordered set of key images $\{\ldots, I_k, I_{k+1}, I_{k+2}, \ldots\}$ from the video path to represent the video sequence of the reference path.

(c) The feature points were tracked via the procedures illustrated in Fig. 5, and the basic matrix was obtained and optimized to realize target localization.

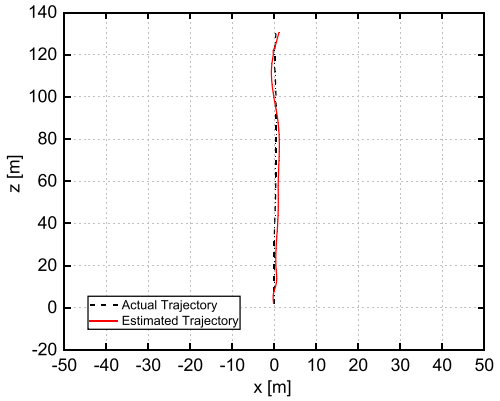

FIGURE 13. The actual and estimated trajectories.

2) RESULTS DISCUSSION

The pixel coordinates in the left-right matching graph of the two corresponding frames are obtained for the tracked feature points. According to Fig. 5 and formula 11, by substituting the pixel coordinates of the images that correspond to the current and next frames of the tracking points into the coordinate formula of spatial point P in the world coordinate system, the motion trajectory can be obtained. Qualitative comparisons of our trajectories and actual trajectory are shown in Fig. 13, where the black line represents the actual trajectory of the vehicle, which is the ground truth, and the red line represents the estimated trajectory of the vehicle, which is the simulation result.

To verify the localization performance, the localization errors are analyzed. Since the Y-coordinates of the recovered three-dimensional coordinates of the spatial points are in the upper and lower directions, which is shown in Fig. 12, the localization errors are not affected in the motion trajectory diagram; only the motion across the Y- and Z-axes are considered. Because it is difficult to take the whole tracking link into account when calculating the error from the starting position or the ending position, the result will be inaccurate, so the root mean square error (RMSE) is considered as the evaluation factor. Table 1 shows the true value, estimated value and median RMSE of the frame trajectories.

According to Table 1, the median RMSE is 1.43 m which means that the trajectory error is approximately 0.9% of its dimension, which is 1.2 $\times$ 131 m. The errors between the actual trajectory and the estimated trajectory of the robot and RMSE are controlled within an allowable range. Although the errors and RMSE are allowable, the reason is still analyzed, and there are three main reasons for the errors: first, the distortion of the lens, which is the systematic error; second, the calibration error; and third, the matching error. The camera parameters calculated by the calibration algorithm differ from the true values. The calibration error can be gradually reduced by adopting a more accurate calibration algorithm or by using more precise calibration equipment;. With the continuous improvement of the matching algorithm, the impact of the matching error will decrease.

TABLE 1. Results of our system.

X-axis			Z-axis			RMSE (m)
True (m)	Estimated (m)	Error (m)	True (m)	Estimated (m)	Error (m)	
0	-0.17	-0.17	2	2.15	0.15	1.43
0	-0.28	-0.28	4	4.15	0.15	
0	-0.27	-0.27	6	6.18	0.18	
0	-0.06	-0.06	8	8.20	0.20	
0	0.27	0.27	10	10.21	0.21	
0	0.50	0.50	12	12.22	0.22	
0	0.56	0.56	14	14.23	0.23	
0	0.52	0.52	16	16.28	0.28	
0	0.44	0.44	18	18.30	0.30	
…	…	…	…	…	…	
0.29	0.75	0.46	128	128.76	0.76	
0.39	1.15	0.76	130	130.76	0.76	

TABLE 2. Comparison of the different methods in the raw KITTI dataset.

Sequence	RMSE(m)		
	Our trajectory	ORB feature-based method	Median filter + RANSAC
Residential 2011_09_26_drive_0046	0.75	1.36	1.18
Road 2011_09_26_drive_0029	2.21	2.86	2.96
Campus 2011_09_28_drive_0105	0.94	1.10	1.17

To further evaluate the tracking method, we compared it with two other methods: (1) the ORB feature-based method and (2) the median filter + RANSAC method, which are state-of-the-art methods for using ORB feature points for tracking. We tested the methods in three different scenarios in raw KITTI datasets. These datasets are composed of many frames, including moving objects such as pedestrians and cars. The residential area contains 125 frames in 2011_09_26_drive_0046; the road area contains 130 frames in 2011_09_26_drive_0029; and the campus area contains 106 frames in 2011_09_28_drive_0105. Qualitative comparisons among the actual trajectory, our trajectories and the other trajectories are shown in Fig. 14.

Table 2 shows the RMSE for the three different methods over three executions in different scenarios in the raw KITTI datasets.

The results demonstrate that our proposed method outperforms both the ORB feature-based method and the median filter + RANSAC method in terms of the RMSE. Although the last coordinate of our trajectory is not the closest to the last coordinate of the actual trajectory compared with other methods, the RMSE of our trajectories is less than 1 m in the tested residential and campus areas, which is lower than that of compared methods in the different scenarios. When using the scenarios' dimensions to calculate the errors, our method has a very accurate trajectory error, which is typically approximately 0.19% in residential areas, 0.04% in road areas and 0.2% in campus areas.

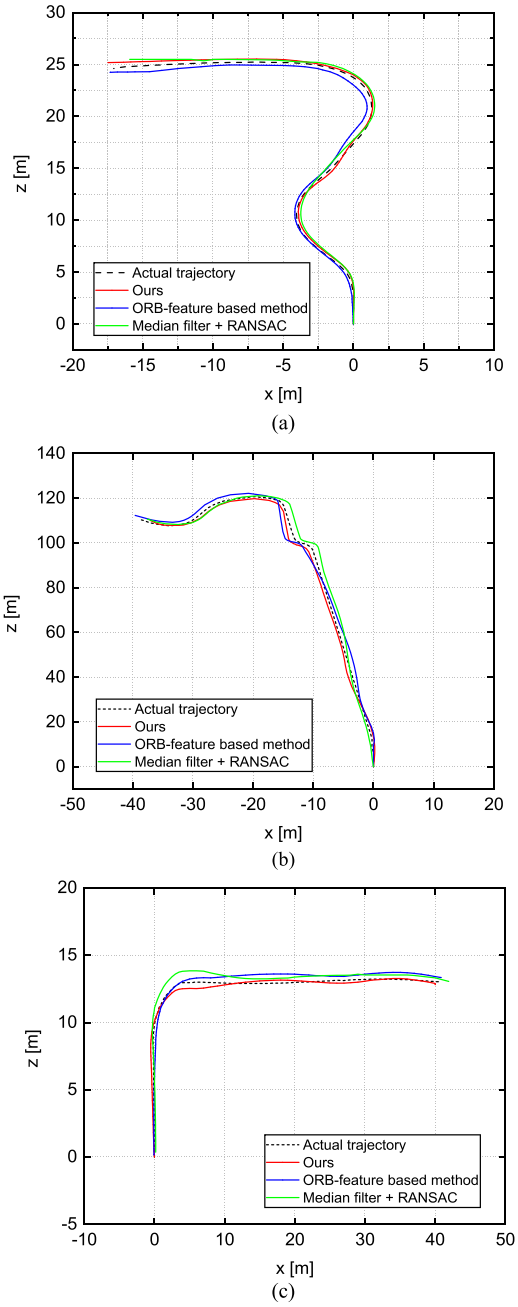

FIGURE 14. The trajectory performance in major scenarios from the raw KITTI dataset. (a) Residential. (b) Road. (c) Campus.

V. CONCLUSION

In this paper, we have proposed an improved feature point extraction method based on a local adaptive threshold for accurately and robustly extracting ORB feature points to solve the problem of sensitivity to changes in environmental factors, including brightness and noise. The experimental results of feature point extraction have demonstrated that compared with the original algorithm and two state-of-the-art methods, our proposed method has the following advantages: the number of feature points, the repetition rate, and the calculating speed fluctuate only slightly with brightness variations; the detection range expands and does not decrease dramatically with brightness variations; the distribution of the feature points is uniform, and they do not overlap.

Then, this paper presented a feature point tracking method based on the above work of improved feature point extraction. We presented extensive experiments both in intelligent vehicle platforms and popular KITTI datasets from many different scenarios, including pedestrians and moving vehicles, to evaluate our method and state-of-the-art methods. According to the results of the experiments, the RMSE for our method is typically less than that of similar methods, and the trajectory error is typically relatively small and in the controllable range when considering the dimensions; thus, it can conduct more accurate and stable feature point tracking.

REFERENCES

[1] L. Azpilicueta, C. Vargas-Rosales, and F. Falcone, "Intelligent vehicle communication: Deterministic propagation prediction in transportation systems," *IEEE Veh. Technol. Mag.*, vol. 11, no. 3, pp. 29–37, Sep. 2016.

[2] G.-J. Horng and S.-T. Cheng, "Using intelligent vehicle infrastructure integration for reducing congestion in smart city," *Wireless Pers. Commun.*, vol. 91, no. 2, pp. 861–883, Jul. 2016.

[3] M. Sefati, M. Daum, B. Sondermann, K. D. Kreiskother, and A. Kampker, "Improving vehicle localization using semantic and pole-like landmarks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 13–19.

[4] B. Tavli, K. Bicakci, R. Zilan, and J. M. Barcelo-Ordinas, "A survey of visual sensor network platforms," *Multimedia Tools Appl.*, vol. 60, no. 3, pp. 689–726, Jul. 2011.

[5] Y. X. Han, Z. S. Zhang, and M. Dai, "Monocular vision system for distance measurement based on feature points," *Opt. Precis. Eng.*, vol. 19, no. 5, pp. 1082–1087, 2011.

[6] Y. Ding and Y. Zhao, "No-reference quality assessment for stereoscopic images considering visual discomfort and binocular rivalry," *Electron. Lett.*, vol. 53, no. 25, pp. 1646–1647, Dec. 2017.

[7] F. Shao, K. Li, W. Lin, G. Jiang, and M. Yu, "Using binocular feature combination for blind quality assessment of stereoscopic images," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1548–1551, Oct. 2015.

[8] L. Zheng, S. Wang, J. Wang, and Q. Tian, "Accurate image search with multi-scale contextual evidences," *Int. J. Comput. Vis.*, vol. 120, no. 1, pp. 1–13, Mar. 2016.

[9] G. De Cubber and H. Sahli, "Augmented lagrangian-based approach for dense three-dimensional structure and motion estimation from binocular image sequences," *IET Comput. Vis.*, vol. 8, no. 2, pp. 98–109, Apr. 2014.

[10] L. Yang, B. Wang, and R. Zhang, "Analysis on location accuracy for the binocular stereo vision system," *IEEE Photon. J.*, vol. 10, no. 1, Feb. 2017, Art. no. 7800316.

[11] Y. Feng, Y. Wu, and L. Fan, "On-line object reconstruction and tracking for 3D interaction," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2012, pp. 711–716.

[12] B. Fan, F. Wu, and Z. Hu, "Rotationally invariant descriptors using intensity order pooling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 2031–2045, Oct. 2012.

[13] J.-Y. Choi, K.-S. Sung, and Y.-K. Yang, "Multiple vehicles detection and tracking based on scale-invariant feature transform," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2007, pp. 528–533.

[14] H. Bay, T. Tuytelaars, and L. J. V. Gool, "SURF: Speeded-up robust features," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[15] 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.

[16] J. Li, Y. Wang, and Y. Wang, "Visual tracking and learning using speeded up robust features," *Pattern Recognit. Lett.*, vol. 33, no. 16, pp. 2094–2101, Dec. 2012.

[17] S. Li, R. Fan, G. Lei, G. Yue, and C. Hou, "A two-channel convolutional neural network for image super-resolution," *Neurocomputing*, vol. 275, pp. 267–277, Jan. 2018.

[18] J.-W. Hsieh, L.-C. Chen, and D.-Y. Chen, "Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 6–20, Feb. 2014.
[19] H. Zhang, Y. Wang, L. Luo, X. Lu, and M. Zhang, "SIFT flow for abrupt motion tracking via adaptive samples selection with sparse representation," *Neurocomputing*, vol. 249, pp. 253–265, Aug. 2017.
[20] H. Ke, D. Chen, X. Li, Y. Tang, T. Shah, and R. Ranjan, "Towards brain big data classification: Epileptic EEG identification with a lightweight VGGNet on global MIC," *IEEE Access*, vol. 6, pp. 14722–14733, 2018.
[21] M. Al-Qizwini, I. Barjasteh, H. Al-Qassab, and H. Radha, "Deep learning algorithm for autonomous driving using GoogLeNet," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2017, pp. 89–96.
[22] K. Kavitha, B. Sandhya, and B. Thirumala, "Evaluation of distance measures for feature based image registration using AlexNet," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 10, pp. 284–289, 2018.
[23] T. Cieslewski, E. Stumm, A. Gawel, M. Bosse, S. Lynen, and R. Siegwart, "Point cloud descriptors for place recognition using sparse visual information," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 4830–4836.
[24] S. Wang, Y. Li, Y. Sun, X. Li, N. Sun, X. Zhang, and N. Yu, "A localization and navigation method with ORB-SLAM for indoor service mobile robots," in *Proc. IEEE Int. Conf. Real-Time Comput. Robot. (RCAR)*, Jun. 2016, pp. 443–447.
[25] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[26] Y. Lei, Z. Yu, and Y. Gong, "An improved ORB algorithm of extracting and matching features," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 8, no. 5, pp. 117–126, May 2015.
[27] Y. Qin, H. Xu, and H. Chen, "Image feature points matching via improved ORB," in *Proc. IEEE Int. Conf. Progr. Informat. Comput.*, May 2014, pp. 204–205.
[28] J. Huang, G. Zhou, X. Zhou, and R. Zhang, "A new FPGA architecture of FAST and BRIEF algorithm for on-board corner detection and matching," *Sensors*, vol. 18, no. 4, p. 1014, Mar. 2018.
[29] X. M. Dai, L. Lang, and M. Y. Chen, "Research of image feature point matching based on improved ORB algorithm," (in Chinese), *J. Electron. Meas. Instrum.*, vol. 30, no. 2, pp. 233–240, 2016.
[30] C. Sun and P. Vallotton, "Fast linear feature detection using multiple directional non-maximum suppression," *J. Microsc.*, vol. 234, no. 2, pp. 147–157, May 2009.

HANG LI received the Ph.D. degree from the Beijing Institute of Technology. He is currently a Professor with the School of Mechatronics Engineering, Henan University of Science and Technology. His research interests include mobile robot, parallel robot theory and technology, image recognition, and target tracking based on machine vision.

HONGFAN YANG was born in 1994. He received the bachelor's degree from the School of Mechatronics Engineering, Henan University of Science and Technology, in 2016, where he is currently pursuing the degree. His research interests include computer vision and image processing.

KAIYANG CHEN was born in 1996. He received the bachelor's degree from the School of Mechatronics Engineering, Henan University of Science and Technology, in 2017, where he is currently pursuing the degree. His research interests include signal processing and image processing.