China Instrument and Control Society
Instrumentation
OPEN ACCESS
INSTRUMENTATION, Vol. 11, No. 1, March 2024 https://doi.org/10.15878/j.instr.202300160

Article

Illumination Adaptive Identification Algorithm of a Reconfigurable Modular Robot

Fangyi Xing[1, 2], Cheng Xu[2, 3], Yanming Wu[4] and Hongwei Gao[1, 2, *]

[1] School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang 110158

[2] State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016

[3] School of Instrument Science and Engineering, Southeast University, Nanjing 210096

[4] School of Automation, Shenyang Aerospace University, Shenyang 110135

[*] Corresponding author email: 30963915@qq.com

Abstract: Reconfigurable modular robots feature high mobility due to their unconstrained connection manners. Inspired by the snake multi-joint crawling principle, a chain-type reconfigurable modular robot (CRMR) is designed, which could reassemble into various configurations through the compound joint motion. Moreover, an illumination adaptive modular robot identification (IAMRI) algorithm is proposed for CRMR. At first, an adaptive threshold is applied to detect oriented FAST features in the robot image. Then, the effective detection of features in non-uniform illumination areas is achieved through an optimized quadtree decomposition method. After matching features, an improved random sample consensus algorithm is employed to eliminate the mismatched features. Finally, the reconfigurable robot module is identified effectively through the perspective transformation. Compared with ORB, MA, Y-ORB, and S-ORB algorithms, the IAMRI algorithm has an improvement of over 11.6% in feature uniformity, and 13.7% in the comprehensive indicator, respectively. The IAMRI algorithm limits the relative error within 2.5 pixels, efficiently completing the CRMR identification under complex environmental changes.

Keywords: reconfigurable modular robot; visual identification; feature detection; feature matching

Citation: Fangyi Xing, Cheng Xu, Yanming Wu and Hongwei Gao. "Illumination Adaptive Identification Algorithm of a Reconfigurable Modular Robot." *Instrumentation* 11, no. 1 (March 2024). https://doi.org/10.15878/j.instr.202300160

0 Introduction

With the improvement of reconfigurable robots and identification algorithms, self-reconfiguration of mobile robots has an increasingly broad prospect in the fields of anti-terrorism, disaster relief, and exploration[1]. Reconfigurable robots could be applied in some harsh circumstances, thus guaranteeing people's working environment and life safety. A self-reconfiguration robot has many identical modules, and the robot deforms to different configurations to confront practical needs[2]. Among them, caterpillar self-reconfiguration robots have a good ability to overcome obstacles and ground adaptability, which provides aresearch hotspot in this field.

Since Fukuda et al. proposed the first reconfigurable mobile cell structured robot[3], reconfigurable technology has undergone rapid development. A Swarm-bot[4] with a wheel track was introduced, which had lightweight modules. Although the Swarm-bot is tight and flexible, its obstacle-traversing ability is poor. After that, Inoue et al. proposed a remotely operated vehicle[5], which had a well underwater operation ability. JL-2 tracked robot[6, 7] is a dual-track module robot with a robust locking mechanism. Liu et al. designed an AMOEBA-I robot[8] with three-track modules, which was well applied in the life rescue of the Ya'an earthquake[9]. Even though the above robots have great feasibility, most of their single-track module is unable to move or communicate independently,

thus limiting their flexibility and application scenarios.

The identification of modular reconfigurable robots is the key to achieving self-reconfigurable functions. The mainstream methods of target identification and positioning that have been implemented can be classified based on the sensors used: ultrasonic or infrared positioning, LiDAR distance positioning, and camera vision positioning. Yim et al. developed PolyBot equipped with infrared sensors[10]. To adjust the relative positions between modules, the offset information is calculated continuously by the centering method. CONRO[11] docked by mountain climbing algorithm, but required two docking surfaces within its infrared sensing range. The JL-2 tracked robot[6,7] measured the geometric distance through ultrasonic sensors to obtain the offset information between modules for docking. These reconfiguration schemes mostly utilized infrared and ultrasonic for docking, andtheir accuracy is not satisfactory[12].

LiDAR could be used for indoor target localization and map modeling. Liu et al.[13] employed optimized HDBSCAN to detect and trace human legs. Through the clustering of the leg model, the obstacle interference is effectively eliminated. Yao et al.[14] utilized two-dimensional LiDAR and DW-SVDD to detect leg targets. The dynamic leg data could be obtained even with obstacles and partial occlusion.

The most extensive technical route in the field of robot positioning is currently the use of cameras to detect contours or colors to trace targets[15]. Vision-based object detection typically performs feature extraction and classification in each frame to determine the target position[16]. Robots such as M-TRAN[17] and CKbot[18] docking through cameras,but their docking surfaces require mutual induction, which limits the docking range. Yi et al.[19] proposed a method called Patch SVDD for image anomaly detection and segmentation. The algorithm checks the image at the patch level and is able to locate defects. Oriented FAST and Rotated Brief (ORB)[20] is an efficient feature detection algorithm, which is widely applied in target detection and identification. However, the feature points detected by ORB are unevenly distributed and excessively clustered under non-uniform illumination, which affects the subsequent positioning accuracy of reconfigurable modular robots. The three-generation evolutionary algorithms of ORB-SLAM improved the accuracy of mapping and localization[21-23]. Mur-Arta et al.[22] proposed to utilize quadtree in the ORB-SLAM2 algorithm, which enhanced the uniformity of feature distribution, with low real-time performance. By improving Mur-Arta's method, Yu et al.[24] presented a quadtree ORB algorithm that effectively eliminates redundant features. Sun et al.[25] improved feature uniformity by setting thresholds to limit the feature extraction number in each region. Fan et al.[26] utilized an adaptive threshold to extract features, but there still exists some manual parameters, thus unable to realize the adaptive feature extraction.

In this work, a chain-type reconfigurable modular robot (CRMR) is designed. The reconstruction and docking approaches of CRMR are divided into two steps: remote guidance and near-end docking. During the remote guidance phase, it is necessary to use an algorithm to identify the CRMR modules and guide them. This promotes mutual visibility between the active module and the target module, which is conducive to the next stage of near-end docking.

Therefore, an Illumination Adaptive Modular Robot Identification (IAMRI) algorithm is proposed to effectively identify the reconfigurable modular robot.The main difficulties and contributions of this paper are summarized as follows:

1) In order to improve the accuracy of feature point detection, the adaptive threshold oriented FAST（oFAST) algorithm is used to detect feature points.

2) To improve the uniformity of feature points, the optimized quadtree decomposition method is used to allocate feature points.

3) To improve the accuracy of matching recognition, an optimized RANSAC algorithm is used to eliminate erroneous matches and effectively locate the robot range.

The rest of this paper is organized as follows. Section 2 introduces our reconfigurable modular robot platform. Section 3 describes the design of the IAMRI algorithm. Section 4 provides experimental analyses of the IAMRI algorithm performances. Finally, concluding remarks are given in Section 5.

1 The Reconfigurable Modular Robot Platform

Owing to the multi-joint crawling, snakes have fantastic environmental adaptability and climbing ability. Based on this principle, a chain-type reconfigurable modular robot (CRMR) is designed (Fig.1). Compared with traditional wheeled or legged robots, snake-like crawling robots with 3D motion capabilities have adaptive travel advantages in complex environments[27]. Traditionally, modules of the snake-like crawling robot are connected through serpentine chain joints. However, this scheme has geometric interference, which limits the motion range between adjacent modules[28]. To solve this issue, a reconfigurable structure with offset joints is adopted, which could flexibly change the configuration. The number of CRMR modules is selected according to actual requirements, such as climbing, obstacle crossing, and so forth. Encouragingly, the deformable configurations increase exponentially with modules, which enriches the application scenario of CRMR.

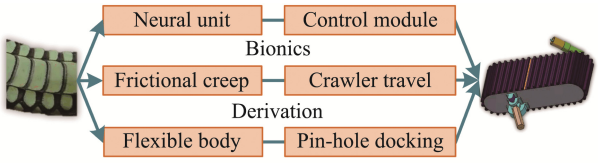

Fig.1 Modular robot with a snake-like structure

Concretely, a CRMR module is composed of a pole drive system, a caterpillar drive system, a pitch arm drive system, and an offset yaw arm drive system (Fig.2). A CRMR module weighing 8kg is driven by four DC motors, with a length of 425mm and a width of 270mm. Meanwhile, ZigBee communicates information between a computer and modules, allowing the individual module to operate independently (Fig.3). It is worth noting that two CRMR modules are defined as one active module. The active module interconnects freely with other modules to realize reconfiguration.

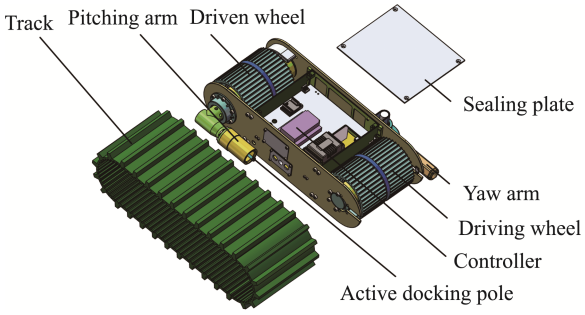

Fig.2 The structure of a CRMR module

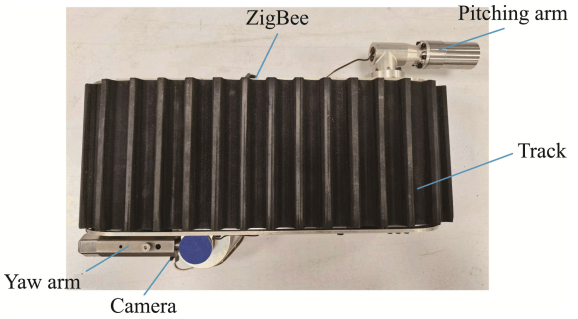

Fig.3 The CRMR unit module platform

2 Design of the IAMRI Algorithm

Non-uniform lighting often occurs indoors or outdoors, which affects the feature detection in visual missions. In this case, feature points collected by the traditional ORB algorithm are uneven and clustered, which is not conducive to the subsequent feature matching[20]. To this end, an Illumination Adaptive Modular Robot Identification (IAMRI) algorithm is proposed. Fig.4 shows the flowchart of the IAMRI algorithm. The IAMRI uses a revised ORB algorithm to detect features of the collected image, and Quadtree is applied to distribute the feature points uniformly, even with uneven illumination.

2.1 Feature Detection with Adaptive Threshold

Based on the principle of human visual perception, a K-layer image pyramid is constructed to maintain feature scale invariance. An image scaling factor T is set, and K-1 down-sampling is performed on the original Gaussian blurred image O_1. Therefore, the image O_k after down-sampling is represented as:

$$O_k = O_1 / T \cdot k \tag{1}$$

where $k \in \{1, 2, \cdots, K\}$, and K is calculated from the original image size and camera speed:

$$K = [\log_2(\min(M, N))] - [\mu / f] - 3 \tag{2}$$

where M and N are original image rows and columns, respectively, f is the camera frame rate, and μ is the scale factor, which is set to 12.

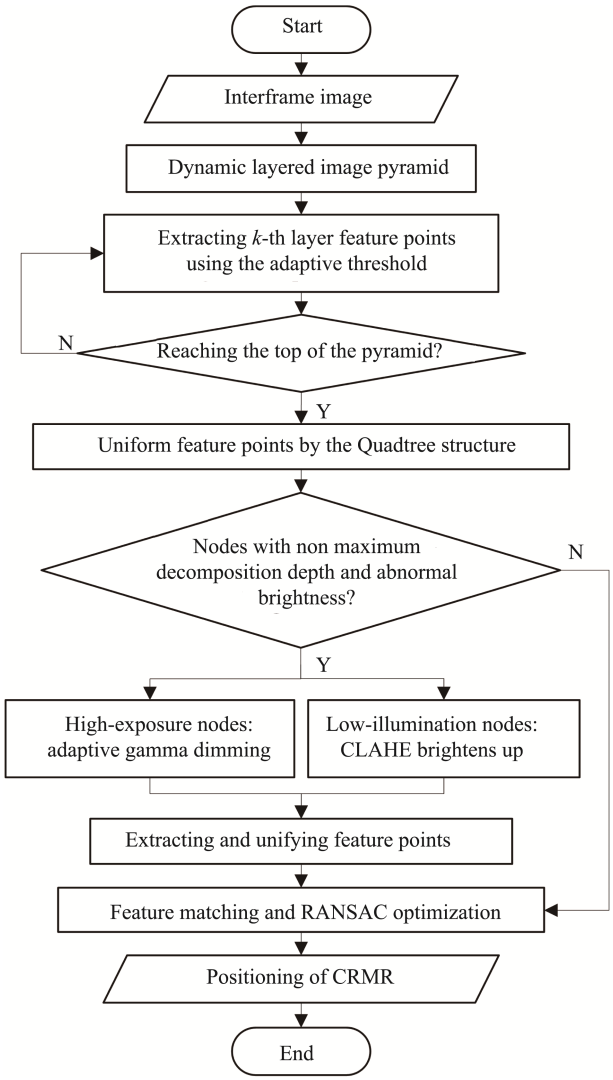

Fig.4 The route of the IAMRI algorithm

At the same time, the imagepyramid is gridded. Following the neighborhood detection rule of oFAST[20], each pyramid layer is evenly divided into grid regions of size 30×30. Areas that are not evenly divided by grid cells are subsumed into adjacent grids.

The ORB algorithm uses threshold t to extract oFAST features with different grayscales. However, under uneven illumination, image noise and contrast are variable in different areas, which means that the threshold t ought to be different. To this end, an adaptive threshold T_a is proposed for the oFAST feature detection:

$$T_a = \left(\frac{1}{n}\sum_{i=1}^{n}(I(x_i) - \overline{I(x)})^2\right) / \overline{I(x)} \tag{3}$$

where $I(x_i)$ is the ith pixel grayscale, $\overline{I(x)}$ is the grayscale average in the grid region, and n is the pixel number in the grid region.

To facilitate the understanding of feature detection, the oFAST feature detection schematic is shown in Fig.5. The adaptive threshold T_a is utilized to extract pixel-specific features in each pyramid grid region. In detail, the pixel P grayscale is set to I_p. Sixteen pixels are taken clockwise from a circle of radius three centered at P. If 12 consecutive pixel grayscale is greater than $I_p + T_a$ or less than $I_p - T_a$, then pixel P is recorded as a feature. Meanwhile, the gray centroid C of the region around the feature P is obtained using the gray centroid method[20]. Vector **PC** is used to add directional properties to features. After the above steps, if there is not enough feature collection in the grid area, T_a is adjusted to T_a / 2, and features are extracted again.

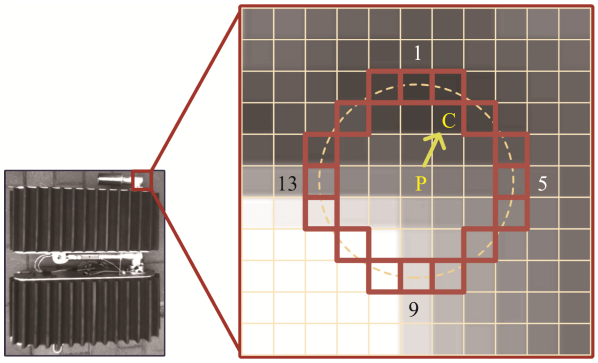

Fig.5 oFAST feature detection schematic

2.2 Feature Uniform Distribution

To handle the clustering and redundancy of detected features, the features are screened by an optimized quadtree algorithm. First, the maximum decomposition depth D of the quadtree is set to avoid excessive uniformity caused by too many iterations[29]. Each layer of the pyramid image is initially treated as a parent node. If there is more than one feature point in a node, the node needs to be divided into four nodes uniformly, until reaching the maximum depth D (Fig.6). Subsequently, the feature with the maximum Harris response is reserved in each node[30].

Image enhancement is carried out in areas with less feature detection. At the beginning, we calculate the brightness of the nodes that stop decomposing before reaching the maximum depth. Based on the pixel selection modes in Fig.7, the average brightness Avg of a node is calculated:

$$Avg = \frac{1}{m}\sum_{j=1}^{m}\overline{I_j(x_j, y_j)} \tag{4}$$

where m is the selected pixel number, $\overline{I_j(x_j, y_j)}$ is the average brightness of the collected pixels.

The node type is judged according to the average brightness: If Avg is higher than the threshold t_1, the high-brightness node is determined, which requires a decrease in brightness. If Avg is below threshold t_2, the low-brightness node is determined, which requires enhanced brightness. Besides, the node has normal brightness. Here, we set t_1 to 120 and t_2 to 100.

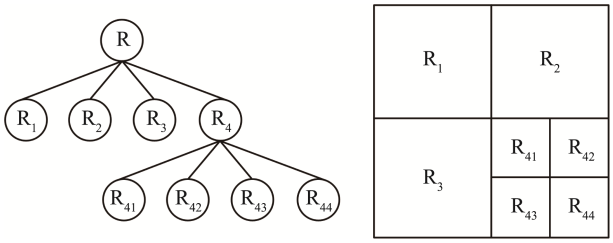

Fig.6 The decomposition of quadtree

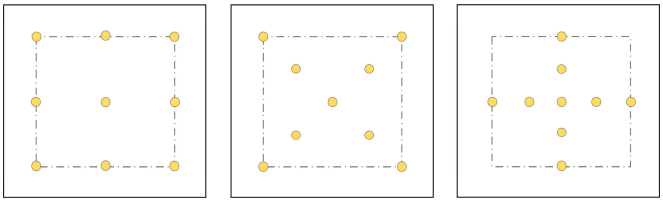

Fig.7 The pixel selection modes of a node

For the high-brightness node, we need to reduce luminance. The Gamma algorithm is usually applied to process partially bright images with a fixed correction factor γ[31]. Here, an adaptive gamma algorithm is designed to revise image details.The calculation is as follows:

$$G(s) = s_{\max} \times (s / s_{\max})^{\gamma(s)} \tag{5}$$

where $G(s)$ is the corrected image luminance, s is the input image luminance, $s_{\max}$ is the maximum luminance, and $\gamma(s)$ is the adaptive factor utilized for revision:

$$\gamma(s) = 1 - \sum_{s=0}^{s_{\max}}[P_\omega(s) / D(s)] \tag{6}$$

$$P_\omega(s) = (P(s) - P_{\min}) / (P_{\max} - P_{\min}) \tag{7}$$

$$D(s) = \sum_{s=0}^{s_{\max}} P_\omega(s) \tag{8}$$

where $P(s)$ is the image luminance probability density function, $P_{\min}$ and $P_{\max}$ are the maximum and minimum $P(s)$, respectively. Among them, $P(s)$ is expressed as follows:

$$P(s) = n_s / E \tag{9}$$

where n_s is the pixelnumber with luminance s, and E is the image's total pixel number.

For the low-brightness nodes, we need to strengthen their luminance. Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm is utilized for the enhancement[32]. Generally, CLAHE enhances contrast and suppresses local noise.

After image enhancement, these nodes undergo adaptive oFAST feature extraction and quadtree decomposition again. When the number of collected features reaches the expected number, the search and distribution of features are stopped.

2.3 Mismatch Elimination and Module Identification

After the search and allocation of features, the binary rBRIEF[20] feature descriptors are calculated. The

descriptors are compared with those detected in the robot sample. If the Hamming distance of the descriptor is less than twice the minimum matching distance, the pair of features is judged to be a correct match.

Then, the optimized random sample consensus (RANSAC) algorithm[33] is used to delete incorrect matches. Here, valid mathematical models are iterated over datasets containing anomalous data to eliminate outliers. The maximum number of iterations is set to 10. Based on the clustering and high matching characteristics of the features in the texture region, the direction filtering of the clustering points and the homography matrix filtering of the outliers are carried out.

The feature set ***F*** with close distance and small angle change of matching pair is counted. The transformation between the matching point pair (x, y) and (x', y') in the set ***F*** is shown in equation 10. The ***H*** in the formula is used to select other features to retain the correctly matched features.

$$\sigma\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{10}$$

Where, σ is a proportion factor, and h_{33}=1 is utilized to normalize ***H***.

After eliminating mismatches, a perspective transformation of the robot sample image is carried out through ***H***. The contour range of the robot is selected by a frame, and the center (x_b, y_b) of the contour is calculated as equations 11 and 12[34]:

$$x_b = \frac{\sum_{i=1}^{3} x_i^2 y_{i+1} + x_4^2 y_1 - \sum_{i=1}^{3} x_{i+1}^2 y_i - x_1^2 y_4 + \sum_{i=1}^{3} x_i x_{i+1} y_{i+1} + x_4 x_1 y_1 - \sum_{i=1}^{3} x_i x_{i+1} y_i - x_4 x_1 y_4}{3(\sum_{i=1}^{3} x_i y_{i+1} + x_4 y_1 - \sum_{i=1}^{3} x_{i+1} y_i - x_1 y_4)} \tag{11}$$

$$y_b = \frac{\sum_{i=1}^{3} x_i y_{i+1}^2 + x_4 y_1^2 - \sum_{i=1}^{3} x_{i+1} y_i^2 - x_1 y_4^2 + \sum_{i=1}^{3} x_i y_i y_{i+1} + x_4 y_4 y_1 - \sum_{i=1}^{3} x_{i+1} y_{i+1} y_i - x_1 y_1 y_4}{3(\sum_{i=1}^{3} x_i y_{i+1} + x_4 y_1 - \sum_{i=1}^{3} x_{i+1} y_i - x_1 y_4)} \tag{12}$$

3 Experimental Analyses of IAMRI Algorithm Performances

The binocular camera RealSense T265 is exploited to search the CRMR module. Table 1 shows the specifications of CRMR and camera. Experiments on feature uniformity and matching performance are conducted. The test images are derived from three self-made CRMR datasets, which involve different illumination changes, scale changes, and blur changes, and we name them Light, Scale, and Bulr, respectively.

Table 1 Specifications of the camera and CRMR

Specification	Description
Computer configuration	Intel i7-10750H (2.60GHz)
Operating system	Ubuntu16.04
Camera image size	848 pixels * 800 pixels
Camera frequency	30 Hz
Image format	.jpg
CRMR size	425mm*270mm*137mm
CRMR weight	8 Kg

3.1 Experiments on the Feature Uniformity

Based on the paper[35], there are five methods to divide an image into different areas, including vertical and horizontal directions. The number of features in each area is computed to form a regional statistical distribution vector. The variance V of the vector group is calculated and the uniformity u is as follows:

$$u = 101\log(V) \tag{13}$$

Here, the smaller the u, the higher the feature uniformity.

Meanwhile, the accuracy, precision, recall, and F1-score metrics[36,37] of feature detection are represented as follows:

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{14}$$

$$precision = \frac{TP}{TP + FP} \tag{15}$$

$$recall = \frac{TP}{TP + FN} \tag{16}$$

$$F1 - score = \frac{precision \times recall \times 2}{precision + recall} \tag{17}$$

where TP is the number of correctly detecting features, FP is the number of incorrectly detecting features, TN is the number of incorrectly excluded detecting features, and FN is the number of correctly excluded detecting features.

Three images are selected from the CRMR dataset with illumination variations. The ability to extract 1000 features is compared between ORB algorithm[20], Mur-Artal's algorithm (MA)[22], Yu's improved ORB algorithm (Y-ORB)[24], Sun's improved ORB algorithm (S-ORB)[25], and the IAMRI algorithm.

Fig.8 shows the feature extraction results. For the same images in the horizontal direction, features extracted by ORB algorithm are concentrated on the edges of the brick seam and the robot. The S-ORB algorithm has slightly improved through region partitioning. The features extracted by MA, Y-ORB, and the IAMRI algorithm are relatively uniform. In addition, for vertically aligned images with reduced light, the feature distribution of ORB and S-ORB algorithms is slightly reduced from the second figure. The feature positions of MA and Y-ORB algorithms changed greatly in the third figure. Although the feature distribution of the IAMRI algorithm is slightly reduced in the third image, this decrease is mostly in weak texture regions. In obvious texture areas, the feature reproduction and preservation effects are relatively good.

The uniformity and time consumption are shown in Table 2. Compared with ORB algorithm, the IAMRI

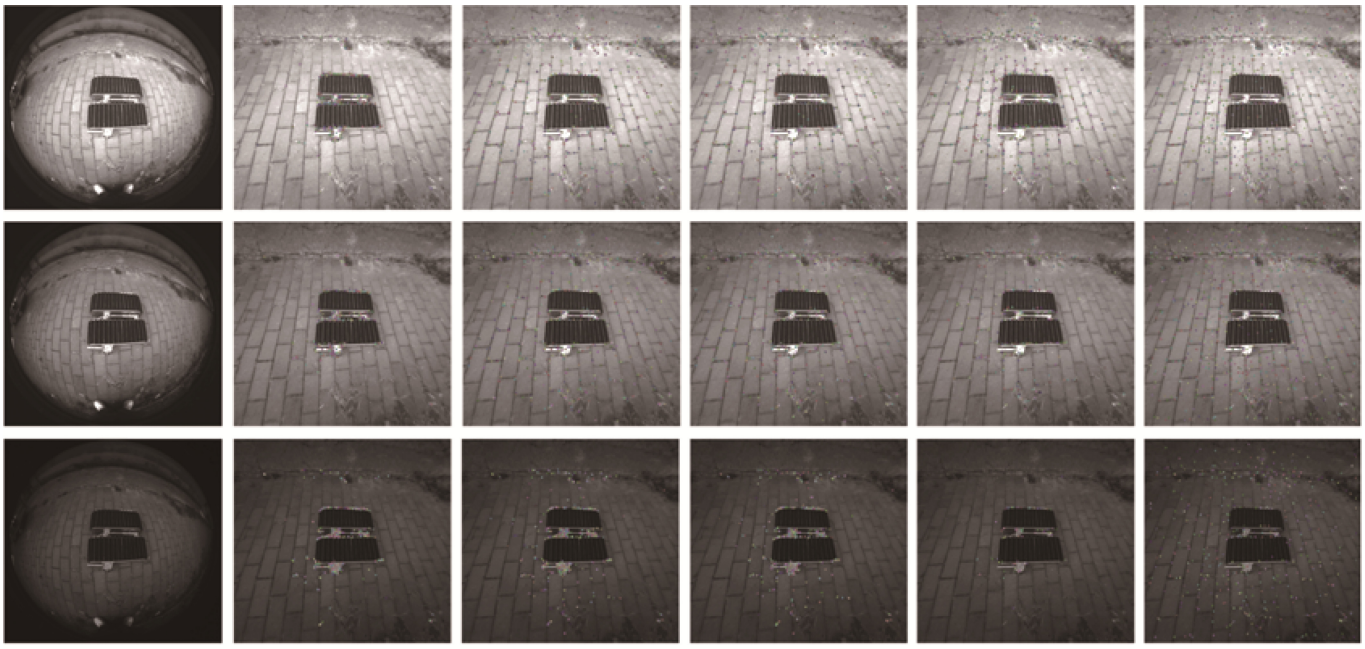

Fig.8 Results of feature extraction

Table 2 Uniformity and time consuming

Images	Uniformity, u					Time consumption, t/ms				
	ORB	MA	Y-ORB	S-ORB	IAMRI	ORB	MA	Y-ORB	S-ORB	IAMRI
1	233.49	182.91	182.68	183.52	165.33	28.79	33.52	33.41	37.75	28.34
2	236.35	165.45	164.19	169.56	164.27	30.31	32.83	35.71	29.77	26.64
3	230.25	220.48	220.73	208.41	166.71	29.51	33.21	31.29	27.74	26.85
Average	233.36	189.61	189.20	187.16	165.44	29.54	33.19	33.47	31.75	27.28

algorithm has a 29.1% promotion in uniformity. Compared to the other three algorithms, the IAMRI algorithm has an increase of over 11.6% in uniformity. In terms of time consumption, the IAMRI algorithm saves over 7.6% compared to the other four algorithms.

The experiments select the representative dataset Light, with changes in illumination, and compare the feature detection precision and recall (Fig.9). In terms of precision, the IAMRI algorithm has 24.7%, 21.1%, 4.5%, and 17.4% improvement over ORB, MA, Y-ORB, and S-ORB algorithms, respectively. In terms of recall, the IAMRI algorithm has 24.2%, 18.6%, 16.3%, and 15.2% improvement over ORB, MA, Y-ORB, and S-ORB algorithms, respectively. At the same time, the feature detection accuracy of the Light dataset and the F1-score of the three datasets are calculated (Fig.10). In terms of accuracy, the IAMRI algorithm has 24.4%, 20.7%, 2.7%, and 13.8% improvement over ORB, MA, Y-ORB, and S-ORB algorithms, respectively. In terms of F1-score, the IAMRI algorithm has 20.8%, 13.4%, 7.4%, and 10.7%

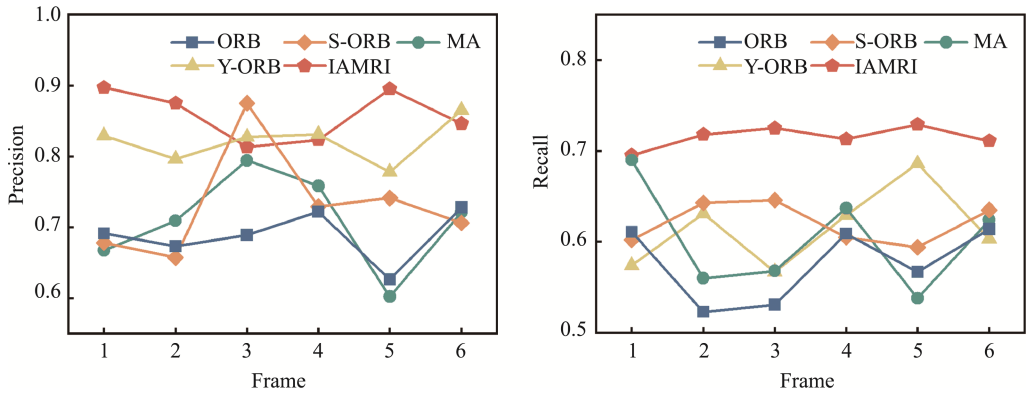

Fig.9 Feature detection precision and recall

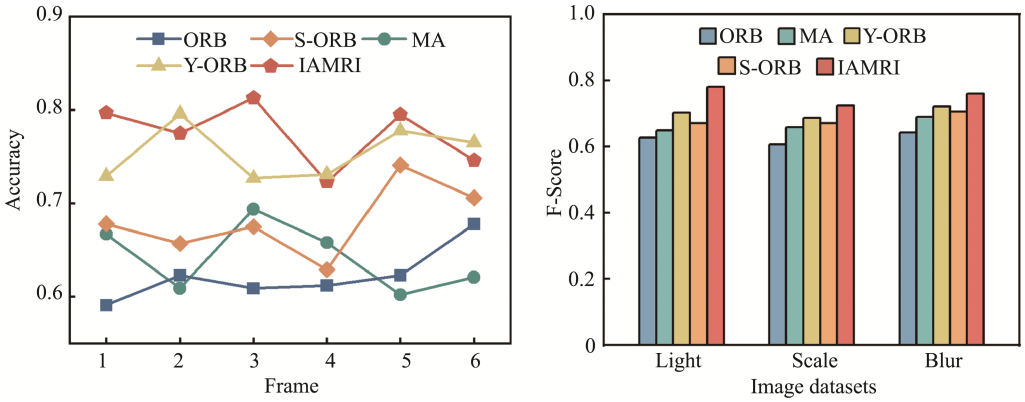

Fig.10 Feature detection accuracy and F1-score

improvement over ORB, MA, Y-ORB, and S-ORB algorithms, respectively. Therefore, the IAMRI algorithm has good stability in feature extraction, which is beneficial for feature matching and localization in complex environments.

3.2 Experiments on the Feature Matching and Identification

The matching precision P and recall rate R of five algorithms are compared. At the same time, the harmonic mean of P and R is taken as the comprehensive score F[38]:

$$P = P_T / (P_T + P_F) \tag{18}$$

$$R = P_T / (P_T + N_F) \tag{19}$$

$$F = (2 \cdot P \cdot R) / (P + R) \tag{20}$$

where P_T is the correct matching number, P_F is the non-matching number, and N_F is the excluded correct matching number.

The dataset with illumination variation is utilized to verify the matching effect (Fig.11). Compared with the other four algorithms, the IAMRI algorithm effectively deletes false matches while maintaining matching uniformity.

The F-scores of three datasets are counted. Moreover, the relative error between robot coordinate measurement and reality is calculated (Fig.12). Compared to the other four algorithms, the IAMRI algorithm has an improvement of over 13.7% in the F-score. In terms of time consumption, the IAMRI algorithm saves over 2.1% compared to the other four algorithms. The relative errors of ORB, MA, Y-ORB, and S-ORB algorithms are within 66.8, 30.2, 13.5, and 59.1 pixels, respectively. Encouragingly, the relative error of the IAMRI algorithm is controlled within 2.5 pixels, which is superior to the other four algorithms. The IAMRI algorithm meets the requirement of CRMR guidance error within 5 pixels. Therefore, our algorithm could identify CRMR more accurately.

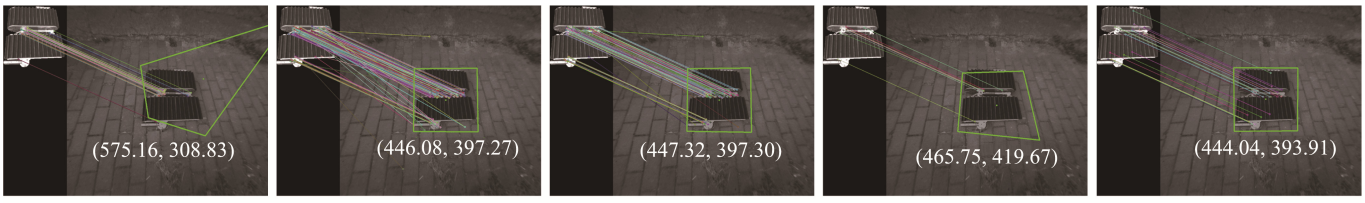

Fig.11 Matching and identification of CRMR

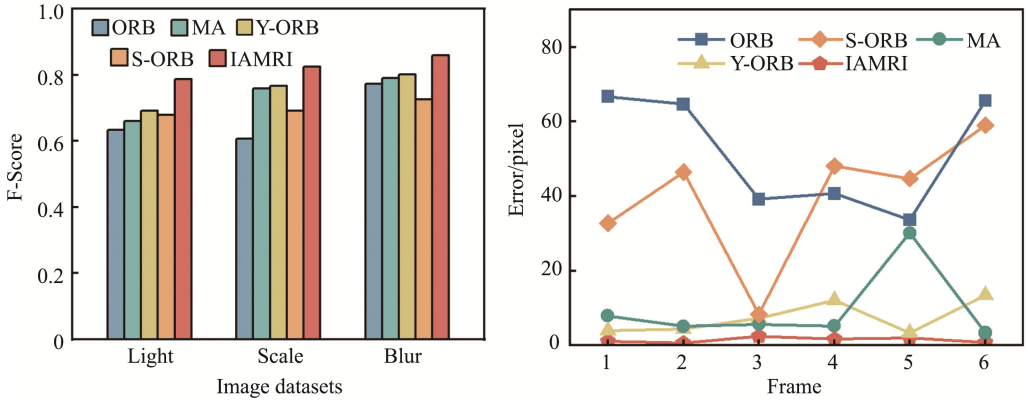

Fig.12 Evaluations of the F and the relative error

4 Conclusion

In this paper, we propose an illumination adaptive modular robot identification algorithm for a chain-type reconfigurable modular robot. The adaptive threshold method is devised to extract features, and an optimized quadtree decomposition method is invented to make feature distribution rational. After matching features, an improved RANSAC algorithm is employed to eliminate the mismatched features. The CRMR module is detected effectively through the perspective transformation.

The effectiveness of the IAMRI algorithm is verified through feature uniformity and matching experiments. In terms of feature extraction, the distribution of features is optimized through an improved quadtree decomposition method. Compared with ORB, MA, Y-ORB, and S-ORB algorithms, the IAMRI algorithm has an improvement of over 11.6% in feature distribution uniformity. Compared to the other four algorithms, the IAMRI algorithm saves over 7.6% in algorithm time. In terms of feature matching, the optimized RANSAC algorithm is utilized to eliminate mismatches. Compared to the other four algorithms, the IAMRI algorithm has an improvement of over 13.7% in the *F*-score. The relative error of the IAMRI algorithm is controlled within 2.5 pixels, which is superior to the other four algorithms.Therefore, the algorithm could identify CRMR more accurately. In the future, we will apply our scheme in practice and conduct more research on the reconstruction of modular robots.

Author Contributions:

Xing Fangyi mainly focuses on the design of research plans, data collection, processing, and analysis, as well as the writing of the manuscript. Xu Cheng is mainly responsible for the design and production of the CRMR unit module platform. Wu Yanming mainly assists in research design, method optimization, and guidance work. Gao Hongwei is responsible for the design and implementation of the entire study, oversees and coordinates the overall research, and provides economic and technical support.

Funding Information:

This work was supported by the National Key R&D Program of China (Grant No. 2018YFB1304600), the National Natural Science Foundation of China (Grant No. 62003337), the Open Fund for State Key Laboratory of Robotics (Grant No. 2023O03), and the Liaoning Province Joint Open Fund for Key Scientific and Technological Innovation Bases (Grant No. 2021-KF-12-05).

Data Availability:

The authors declare that the main data supporting the findings of this study are available within the paper.

Conflict of Interest:

No potential conflict of interest was reported by the authors.

Dates:

Received 01 November 2023; Accepted 18 December 2023; Published online 31 March 2024

References

[1] Zhang, T., Zhang, W J., Gupta, M M. (2018). An Underactuated Self-Reconfigurable Robot and the Reconfiguration Evolution. *Mechanism and Machine Theory*, 124, pp.248-258.

[2] Fei, Y Q., Wang, C Y. (2013). Self-Repairing Algorithm of Lattice-Type Self-Reconfigurable Modular Robots. *Journal of Intelligent & Robotic Systems*, 75, pp.193-203.

[3] Fukuda, T., Ueyama, T., Kawauchi, Y., et al. (1992). Concept of Cellular Robotic System (CEBOT) and Basic Strategies for Its Realization. *Computers & Electrical Engineering*, 18(1), pp.11-39.

[4] Dorigo, M., Trianni, V., Şahin, E., et al. (2004). Evolving Self-Organizing Behaviors for a Swarm-Bot. *Autonomous Robots*, 17(2-3), pp.223-245.

[5] Inoue, T., Shiosawa, T., Takagi, K. (2013). Dynamic Analysis of Motion of Crawler-Type Remotely Operated Vehicles. *IEEE Journal of Oceanic Engineering*, 38(2), pp.375-382.

[6] Wang, W., Yu, W P., Zhang, H X. (2010). JL-2: A Mobile Multi-robot System with Docking and Manipulating Capabilities. *International Journal of Advanced Robotic Systems*, 7(1), pp.9-18.

[7] Wang, W., Zhang, H X., Yu, W P. (2010). Design and Realization of Multimobile Robot System With Docking Manipulator. *Journal of Mechanical Design*, 132(11), pp.1-8.

[8] Liu, J G., Wang, Y C., Li, B., et al. (2007). Center-Configuration Selection Technique for the Reconfigurable Modular Robot. *Science in China Series F: Information Sciences*, 50(5), pp.697-710.

[9] Gao, Y., Liu, J G. (2014). China's Robotics Successes Abound. *Science*, 345(6196), pp.523.

[10] Park, M., Chitta, S., Teichman, A., et al. (2008). Automatic Configuration Recognition Methods in Modular Robots. *The International Journal of Robotics Research*, 27(3-4), pp.403-421.

[11] Khoshnevis, B., Will, P., Shen, W M. (2003). Highly Compliant and Self-Tightening Docking Modules for Precise and Fast Connection of Self-Reconfigurable Robots. In: *IEEE International Conference on Robotics & Automation*. Taiwan: IEEE, pp.2311-2316.

[12] Ma, L., Wu, X Y., Li, Z W. (2021). High-Precision Medicine Bottles Vision Online Inspection System and Classification Based on Multifeatures and Ensemble Learning Via Independence Test. *IEEE Transactions on Instrumentation and Measurement*, 70, pp.1-12.

[13] Liu, P., Yao, H., Dai, H., et al. (2022). The Detection and Following of Human Legs Based on Feature Optimized HDBSCAN for Mobile Robot. *Journal of Physics: Conference Series*, 2216(1), pp.1-11.

[14] Yao, H., Peng, J., Liao, Z., et al. (2023). Leg Detection for Socially Assistive Robots: Differentiating Multiple Targets with 2D LiDAR. In: *International Conference on Cognitive Systems and Signal Processing. Singapore: Springer*, pp.87-103.

[15] Algabri, R., Choi, M-T. (2020). Deep-Learning-Based Indoor Human Following of Mobile Robot Using Color Feature. *Sensors*, 20(9), pp.1-19.

[16] Gupta, M., Kumar, S., Behera, L., et al. (2017). A Novel Vision-Based Tracking Algorithm for A Human-Following Mobile Robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(7), pp.1415-1427.

[17] Murata, S., Kakomura, K., Kurokawa, H. (2006). Docking Experiments of A Modular Robot by Visual Feedback. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing: IEEE, pp.625-630.

[18] Yim, M., Shirmohammadi, B., Sastra, J., et al. (2007). Towards Robotic Self-Reassembly After Explosion. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Diego: IEEE, pp.2767-2772.

[19] Yi, J., Yoon, S. (2020). Patch SVDD: Patch-Level SVDD for Anomaly Detection and Segmentation. In: *Asian Conference on Computer Vision*. Springer, pp.1-16.

[20] Rublee, E., Rabaud, V., Konolige, K., et al. (2011). ORB: An Efficient Alternative to SIFT or SURF. In: *IEEE International Conference on Computer Vision*. Barcelona: IEEE, pp.2564-2571.

[21] Mur-Artal, R., Montiel, J M M., and Tardos, J D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5), pp.1147-1163.

[22] Mur-Artal, R., Tardos, J D. (2017). ORB-SLAM2: An Open-Source Slam System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5), pp.1255-1262.

[23] Campos, C., Elvira, R., Rodríguez, J J G., et al. (2021). ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Transactions on Robotics*, 37(6), pp.1874-1890.

[24] Yu, X Y., Zhan, Y A., Zhu, F., et al. (2018). Improved ORB Feature Extraction Algorithm Based on Quadtree Encoding. *Computer Science*, 45(Z11), pp.222-225.

[25] Sun, H., Wang, P. (2020). An Improved ORB Algorithm Based on Region Division. *Journal of Beijing University of Aeronautics and Astronautics*, 46(9), pp.1763-1769.

[26] Fan, X N., Gu, Y F., Ni, J J. (2019). Application of Improved ORB Algorithm in Image Matching. *Computer and Modernization*, 282(2), pp.5-10.

[27] Li, B., Liu, J G., Tan, D L. (2005). Research on the Tipover Stability of a Reconfigurable Modular Robot. *Robot*, 27(3), pp.241-283.

[28] Liu, J G., Wang, Y C., Ma, S G., et al. (2006). Analysis of Tipover Stability for Novel Shape Shifting Modular Robot. *Chinese Journal of Mechanical Engineering*, 19(2), pp.187-192.

[29] Chen, Q., Yao, L., Xu, L., et al. (2022). Horticultural Image Feature Matching Algorithm Based on Improved ORB and LK Optical Flow. Remote Sensing, 14(18), pp.1-18.

[30] Cheng, J., Jin, Y., Zhai, Z., et al. (2022). Research on Positioning Method in Underground Complex Environments Based on Fusion of Binocular Vision and IMU. *Sensors*, 22(15), pp.1-19.

[31] Wang, P., Liu, F L., Yang, C F., et al. (2018). Parameter Estimation of Image Gamma Transformation Based on Zero-Value Histogram Bin Locations. *Signal Processing: Image Communication*, 64, pp.33-45.

[32] Li, T., Zhou, X C., Zhang Y., et al. (2023). Underwater Image Enhancement Based on MSRCR and CLAHE-WGIF. *Instrumentation*, 10(2), pp.19-29.

[33] Cheng, L., Li, M C., Liu, Y X., et al. (2012). Remote Sensing Image Matching by Integrating Affine Invariant Feature Extraction and RANSAC. *Computers & Electrical Engineering*, 38(4), pp.1023-1032.

[34] Xue, X R., Huang, S C., Zhang, G S., et al. (2022). Optimization of Fuel Supply Strategy for Aircraft Longitudinal Motion Based on Complex Method. *Journal of Nanjing University of Aeronautics & Astronautics*, 54(2), pp.297-303.

[35] Ma, C Q., Hu, X G., Xiao J., et al. (2021). Homogenized ORB Algorithm Using Dynamic Threshold and Improved Quadtree. *Mathematical Problems in Engineering*, 2021, pp.1-19.

[36] Zhu, X Z., Tang, C., Wang, P C., et al. (2018). Saliency Detection Via Affinity Graph Learning and Weighted Manifold Ranking. *Neurocomputing*, 312, pp.239-250.

[37] Gao, Z H., Zhang, Q F., Wang, G T., et al. (2023). Image Feature Point Detection Based on Resinv-Unet. *Foreign Electronic Measurement Technology*, 42(4), pp.1-7.

[38] Yuan, X X., Chen S Y., Yuan, W., et al. (2017). Poor Textural Image Tie Point Matching Via Graph Theory. *ISPRS Journal of Photogrammetry and Remote Sensing*, 129, pp.21-31.