IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 13, 2020, accepted March 22, 2020, date of publication March 26, 2020, date of current version April 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2983488

3D Semantic Map Construction Using Improved ORB-SLAM2 for Mobile Robot in Edge Computing Environment

XU CUI, CHENGGANG LU, AND JINXIANG WANG
Department of Computer Science and Technology, Yanbian University, Yanji 133002, China

Corresponding authors: Chenggang Lu (cglu@ybu.edu.cn) and Jinxiang Wang (124961535@qq.com; wangjinxiang@ybu.edu.cn)

ABSTRACT Although the existing localization and mapping (SLAM) technology of indoor mobile robot has made great development, its intelligence and environmental perception ability still cannot meet the needs of service and inspection. Therefore, based on edge computing environment, a 3D semantic map construction of mobile robot based on improved ORB-SALM2 is proposed. Firstly, the improved yolov3 algorithm is used to detect indoor objects, and then the real-time semantic segmentation network model based on deep learning is used to segment indoor objects to achieve the classification of pixel points of objects on two-dimensional images, and BAFF feature fusion algorithm is introduced to improve the accuracy of semantic segmentation model. Then, through the SLAM system, we estimate the pose of the image in the result of semantic segmentation, and use the depth information to project it into the three-dimensional environment to build the three-dimensional semantic map. Finally, the experiment platform of mobile robot is built to verify the stability of ORB-D and thermal imaging sensor registration technology, the accuracy and real-time of building three-dimensional environment thermal field map, and the accuracy of robot positioning using thermal infrared and depth image.

INDEX TERMS Mobile robot, edge computing, 3D semantic map, improved ORB-SALM2, pose estimation, image semantic segmentation, deep learning.

I. INTRODUCTION

Due to the lack of simple labor, people's living standard is getting higher and the population is aging, it has brought greater pressure on medical and care of society [1]. Driven by the above factors, service robots show great development potential and social needs. More and more countries have begun to focus on the development of the robot field. The number of countries involved in the development of service robots is also increasing, and the key technologies are regarded as the key science and technology for future development [2], [3]. SLAM (Simultaneous localization and mapping) is a process in which a robot is equipped with vision, laser, odometer and other sensors to build a map of an unknown environment and achieve self-localization [4], [5]. It plays a key role in robotic autonomous navigation task [6]. After decades of theoretical research and technological precipitation, SLAM technology has achieved rich results, especially a series of breakthrough advances made by visual SLAM in recent years. The desire of mobile robots to achieve autonomous navigation and human-machine integration has gradually become reality.

The current research method of SLAM problem is to estimate the motion information of the robot body and the characteristic information of the unknown environment by installing multiple types of sensors on the robot body and use information fusion to achieve accurate estimation of the robot pose and spatial modeling of the scene [7].

Due to information accumulation error of odometer and other sensors, back-end optimization becomes particularly important. The key is to detect the closed loop correctly [8], [9]. Compared with the single spatial structure perception information of the laser sensor, the visual sensor has huge advantages and potential in improving the accuracy of inter-frame estimation and the accuracy rate of closed-loop detection with rich perceptual information such as color and texture [10]–[12]. Visual SLAM (SLAM) is a SLAM system with image as the main source of environmental perception information. It can be applied to applications such

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

as unmanned driving and augmented reality. It is a popular research direction in recent years [13]–[15].

II. RELATED WORKS

Visual SLAM based on SIFT (scale invariant feature transform) feature [16] and visual SLAM based on ORB (oriented FAST and rotated BRIEF) feature [17] are widely used in the field of visual SLAM relying on its good robustness, superior discrimination ability and fast processing speed. The SIFT-SLAM algorithm proposed in [18] determines the camera pose and map based on the degree error. It does not need to extract corner points and descriptors, but it cannot represent the global feature of an image. It is difficult to solve the problem of cumulative drift in closed-loop detection. The ORB feature detection operator is proposed based on FAST feature detection and BRIEF feature descriptors. Its running time is far better than SIFT and SURF, and it has scale and rotation invariance. It also has invariance in noise and its perspective transformation and can be applied to real-time feature detection. The typical vision SLAM algorithm takes the estimation of camera pose as the main goal and it reconstructs the 3D map through multi-view geometry theory. To improve the data processing speed, some vision SLAM algorithms first extract sparse image features and implement inter-frame estimation and closed-loop detection by matching between feature points. However, it is still an unsolved problem in computer vision that sparse image features represent image information optimally [19]. On the other hand, sparse image features still have more challenges in dealing with changes of lighting, dynamic target movement, changes of camera parameters and lack of texture or the single environment of texture.

Faced with these problems, [20] proposed a hierarchical image feature extraction method represented by deep learning technology in the field of visual SLAM, and it was successfully applied to SLAM inter-frame estimation and closed-loop detection. Deep learning algorithm is the mainstream recognition algorithm in the current computer vision field. They rely on multi-layer neural network to learn the hierarchical feature representation of images. Compared with traditional recognition methods, they can achieve higher recognition accuracy [21], [22]. Meanwhile, the algorithm in [23] can associate image with semantic and combine with SLAM technology to generate a semantic map of the environment. Then, it builds a semantic knowledge base of the environment for the robot to perform cognitive and task reasoning and improve the service capability of robot and intelligence of human-computer interaction [23]. Reference [24] proposed an end-to-end deep neural network architecture to predict changes in camera speed and direction. The main feature of the method is to use a single type of calculation module and learning rule to extract visual motion, depth information and odometer information, which is mainly divided into two steps. First is the extraction of image sequence depth and motion information. However, in term of accuracy, the algorithm has not yet reached the mainstream visual odometer accuracy. Reference [25] *et al.* used the convolutional neural network to learn the optimal feature representation of image data for visual odometer estimation and demonstrated the robustness of its algorithm in dealing with image motion blur and illumination changes. However, the experimental results also show the dependence of the proposed algorithm on training data, especially when the frame speed of the image sequence is too fast, the algorithm error is large. The reason is that the lack of high-speed training samples in the training set results in a large estimated rotation error. References [26] *et al.* have expanded based on spatial transform network and chose to regress classic computer vision method when designing network, such as end-to-end visual odometry and image depth estimation.

However, the existing mobile robots still have problems. For example, it mainly uses similar vision sensor and lacks the complementarity of multi-source sensors and perception information is relatively single. The ability of robots to understand and apply information is still weak. Therefore, an indoor 3D semantic map construction of mobile robot based on improved ORB-SLAM2 is proposed. The main innovations are summarized as follows:

(1) The accuracy of the existing semantic segmentation model is low. Therefore, the proposed method uses deep learning to achieve target detection based on the improved YOLOv3 algorithm. Then, it uses deep learning to segment indoor target objects and classify the target pixel points on a two-dimensional image. And it introduces the BAFF feature fusion algorithm to improve the accuracy of the semantic segmentation model.

(2) RGB camera has poor SLAM stability based on the ORB-SLAM2 algorithm in low-light environment. Therefore, the proposed method uses thermal infrared images and depth images that are less affected by light to estimate robot pose and improve robot positioning stability.

(3) To improve the understanding of mobile robot for the environment, the proposed method uses real-time semantic segmentation technology of deep learning to achieve the positioning and recognition of target objects. Then, it combines the poses obtained by the SLAM algorithm to project a two-dimensional image of semantic information and builds a three-dimensional semantic map to improve the intelligence and environment perception depth of robot.

III. THE OVERALL ARCHITECTURE OF THE PROPOSED METHOD

The overall architecture of SLAM based on edge computing is shown in Figure 1. The data involved in data collection, feature extraction, data matching, semantic segmentation and map building are stored to the edge by encryption, and the real-time processing of data is realized by using the edge computing module, at the same time, the computing with low real-time requirements is solved by cloud computing.

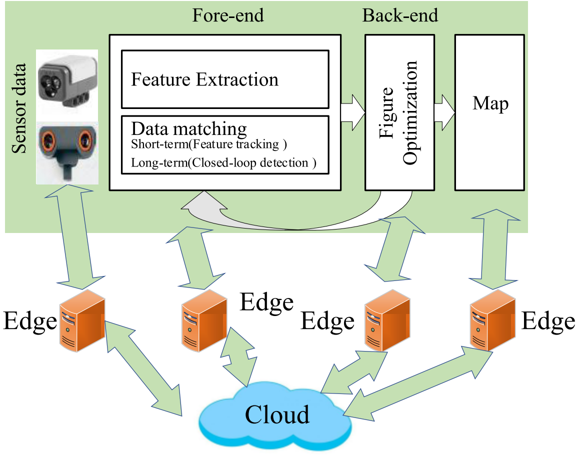

FIGURE 1. Front-end and back-end in a typical SLAM system.

The front-end inter-frame estimation and back-end optimization. The front-end inter-frame estimation solves the motion estimation of the robot in the time interval between acquiring the sensor information of the two frames before and after. The back-end optimization solves the optimization of the historical trajectory after the robot detects the closed loop of the path.

The mobile robot of the indoor environment uses the vision system and the motion system to obtain an image of the environment. After analyzing and processing the core processor by the robot, the three-dimensional coordinate of the robot is obtained to construct a 3D semantic map. The overall architecture of the proposed method is shown in Figure 2.

The vision system of mobile robot uses RGB-D camera and thermal imaging sensor and combines with the motion system to process the acquired images for target detection, semantic image segmentation, feature point matching, pose estimation and optimization, etc. And the 3D semantic map of mobile robot is obtained.

IV. CONSTRUCTION OF 3D SEMANTIC MAP BASED ON IMPROVED ORB-SLAM2

Based on the system framework of the proposed mobile robot 3D semantic map construction method, the proposed method uses the improved YOLOv3 for indoor target detection and uses the real-time semantic segmentation network model of deep learning to classify detected targets. Finally, the image in the result of semantic segmentation is estimated by SLAM system, and the depth information is used to project it into a three-dimensional environment to construct a three-dimensional semantic map.

A. OBJECT DETECTION USING IMPROVED YOLOV3

The feature extraction network Darknet53 of the YOLOv3 real-time target detection model is composed of 53 convolutional layers and 24 residual layers [27], [28]. The last 20 layers are the feature interaction layer of the YOLO network, which is divided into 3 scales. Within each scale, local feature interaction is realized by convolution kernel, which is similar as full connection layer. Local feature interaction between feature images is realized by convolution kernel (3 × 3 and 5 × 5), and the fully connected layer performs global feature interaction. The convolutional layer extracts the image features, and the fully connected layer predicts the image position and class estimated probability value. Based on the input image data, YOLOv3 uses regression analysis to output multiple sliding window positions of image data and the target categories detected in the window.

To improve the ability of detecting small targets of YOLOv3 network in infrared images, an IR-YOLO neural network structure is proposed. The characteristics of small infrared targets are low resolution (small infrared targets are generally 20′20 pixels), the fuzzy details and the lack of color features. Therefore, the improvement of the network structure focuses on the compression feature to extract the network depth. The shallow convolution feature perception field contains little background noise, which is suitable for extracting the semantic features of small target with low resolution and has better representation ability for infrared target. Deep convolution layers are more suitable for processing high-resolution detail features, and for low-resolution image features such as infrared images, there is more background noise in the field of view. Less effective information can be used, so compression can be performed [29]. Meanwhile, to further improve the real-time detection, the original detection layer of YOLOv3 uses an anchorless CenterNet structure. The structure of IR-YOLO neural network is shown in Figure 3.

In the IR-YOLO neural network structure, a total of 9 convolutional layers from 44 to 53 and the last 4 residual layers in the YOLOv3 dry feature extraction network (Darknet53) are clipped, reducing the original backbone feature extraction network from 74 to 61 layers and forming a compressed network structure. The CenterNet used by IR-YOLO is composed of cascading corner pooling and center pooling to obtain rich information from the upper left and lower right corner and obtain more identification information in the middle area. The structure of the detection part is shown in Figure 4.

The central pooling model in the CenterNet network consists of 2 convolution normalized residual fusion layers, 1 left pooling layer, 1 right pooling layer, 1 top pooling layer and 1 bottom pooling layer. It is used to predict the branch of the key point in the center, which is helpful for the center to obtain more central areas of the target and then perceive the central position of the proposed area more easily. This method is implemented by taking the maximum sum of the horizontal and vertical response values of the center position. The cascade corner pooling model consists of two convolution normalized residual fusion layers, one left pooled layer, one convolution normalized fusion layer and one top pooled layer, which is used to increase the function of the perceived internal information [30]. Combine the maximum response sum value of the interior and boundary directions of the target in the

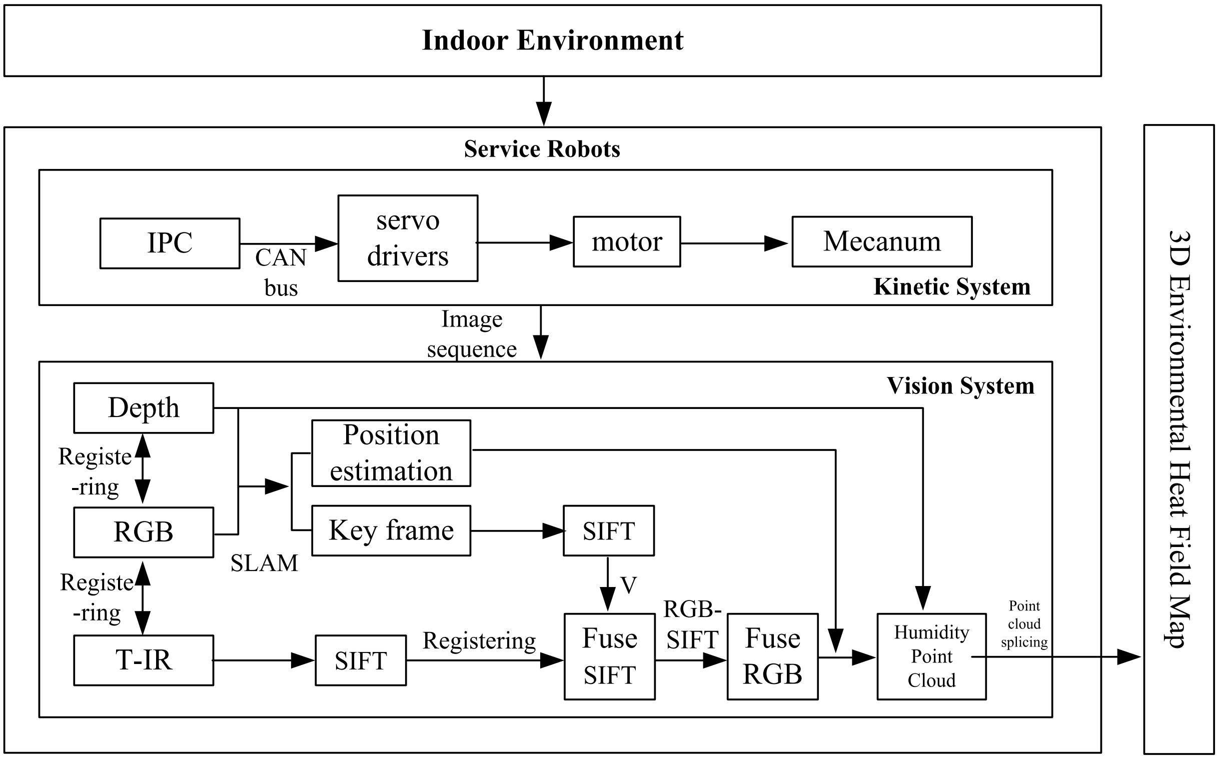

FIGURE 2. Overall framework of the proposed method.

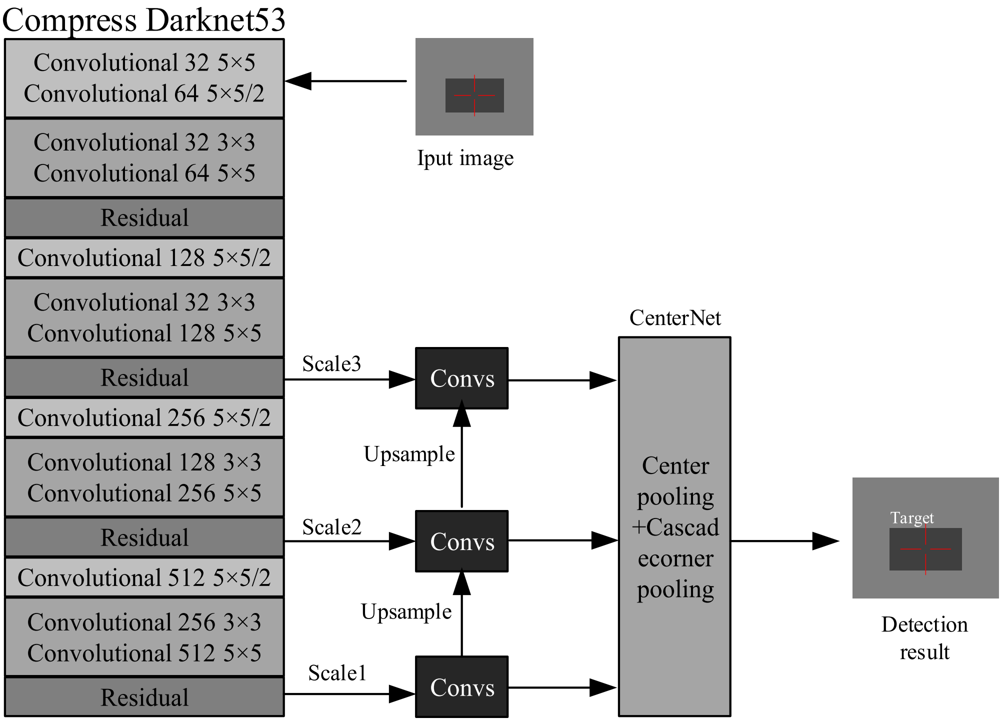

FIGURE 3. IR-YOLO neural network structure.

feature map to predict the upper left corner and the lower right corner. After fusing the output of the central pooling module and the cascaded corner pooling module, the target prediction location can be accurately obtained.

In the process of target detection, the original YOLO layer is replaced by the CenterNet structure on all three detection scales. Among them, CenterNet is large number of incorrect bounding boxes that often appear in anchor-based methods,

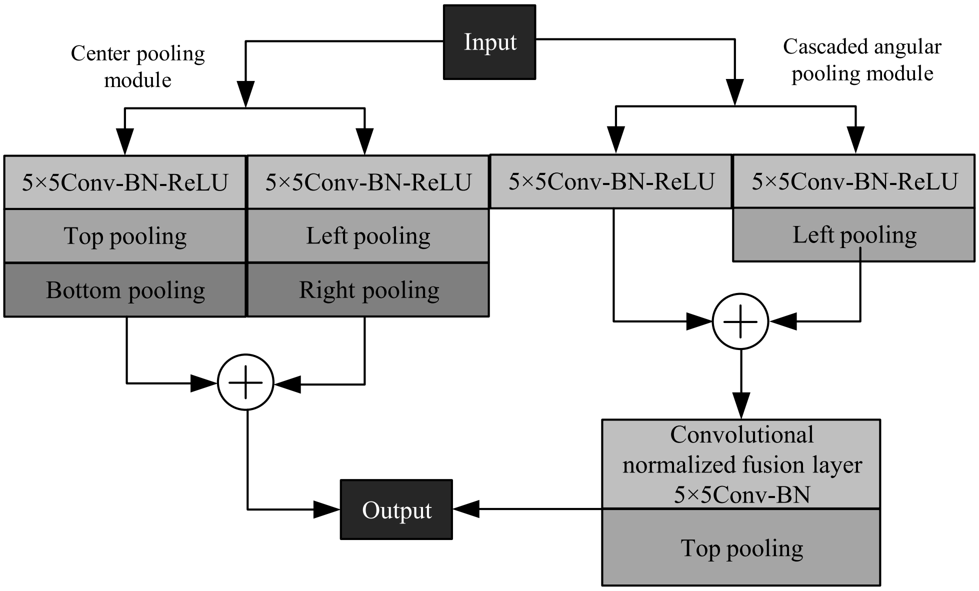

FIGURE 4. **Detection part structure.**

resulting from the lack of additional supervision of relevant clipping regions. Therefore, the original YOLOV3 requires large number of anchor points in the detection layer to target detection. CenterNet is a single-stage key-point detection model. Each target is detected as 3 key-points (center point, upper and lower diagonal point), avoiding generating large number of anchor points, reducing the amount of calculation, improving real-time performance and improving detection accuracy and recall ratio [31].

B. REAL-TIME IMAGE SEMANTIC SEGMENTATION USING DEEP LEARNING

The semantic segmentation model used in the proposed method is shown in Figure 5, where the coding structure uses a lightweight convolutional neural network MobileNet v2 and the decoding structure is built on a conventional jump connection structure. A BAFF feature fusion algorithm (BAFF -SkipNet) is introduced to improve the accuracy of the semantic segmentation model.

Because different feature layers reflect different overall and detailed features of the object, different features are weighted to highlight and retain image features under different receptive fields in the feature fusion process [32]. Firstly, to describe the different characteristics, under a fixed receptive field, a deep learning network is used to scale the same object to test images with different size and the average probability value of the target area is output. Then, considering the characteristic difference of different convolutional layers, Gaussian curves with different shapes and peak centers are used to describe the characteristics of the convolutional layers. The 8th and 20th layers in the network are weighted and fused. This process can be regarded as the complementary superposition of two types of Gaussian curves, expressed as:

$$P_r = \lambda a_1 \exp\left(b_1 (S - c_1)^2\right) + (1 - \lambda) a_2 \exp\left(b_2 (S - c_2)^2\right) \tag{1}$$

where S is the area of the object, a_1, b_1, c_1 are the parameters of the output curve fitting of the 8th layer, a_2, b_2, c_2 are the parameters of the 20th layer output curve fitting and $\lambda$is the weighted value.

Finally, considering the perspective of the scene in the actual image, the area occupied by the same object in the image may vary greatly. The receptive field size of a given convolutional layer in a deep learning network is fixed. To adjust the size of the receptive field in a local area to match the area of the object in the area, the proposed method proposes a block adaptive feature fusion (BAFF) method. The feature map is divided into blocks and weighted fusion is performed separately [33].

Assume that there are two convolutional layers in BAFF, each convolutional layer has l feature map. Assuming that the dimension of each feature map is $w \times h$ and the first convolutional layer is recorded as z_a, then the j feature map of the convolutional layer is expressed as z_a^j. The second convolutional layer is recorded as z_b, then the j feature map of the convolutional layer is represented as z_b^j. σ^j represents the j feature map in the output layer. The calculation process of BAFF is expressed as:

$$\begin{cases} h_1^j = z_a^j * w_1^j + b_1^j \\ h_2^j = z_a^j * w_2^j + b_2^j \\ h_3^j = \mu\left(h_1^j + h_2^j\right) \\ h_4^j = \mu\left(h_3^j * w_4^j + b_4^j\right) \\ h_5^j = up\left(h_4^j\right) \\ \sigma^j = h_5^j \circ z_a^j + \left(1_{K \times L} - h_5^j\right) \circ z_b^j, \quad 1 \le j \le l \end{cases} \tag{2}$$

where $up\,(x)$is the up-sampling function of bilinear interpolation, $\circ$ is the Hadamard product, b_1^j, w_1^j are the weight and offset of the j feature map in the first hidden layer. b_2^j, w_2^j, b_4^j, w_4^j respectively represent the convolution kernel and bias

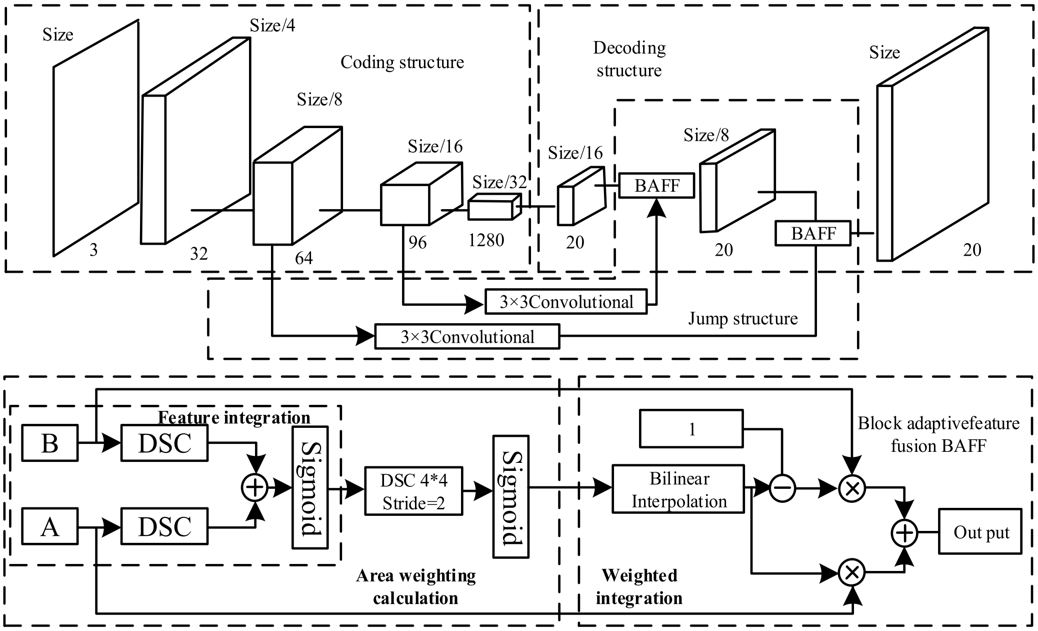

FIGURE 5. Real time semantic segmentation model.

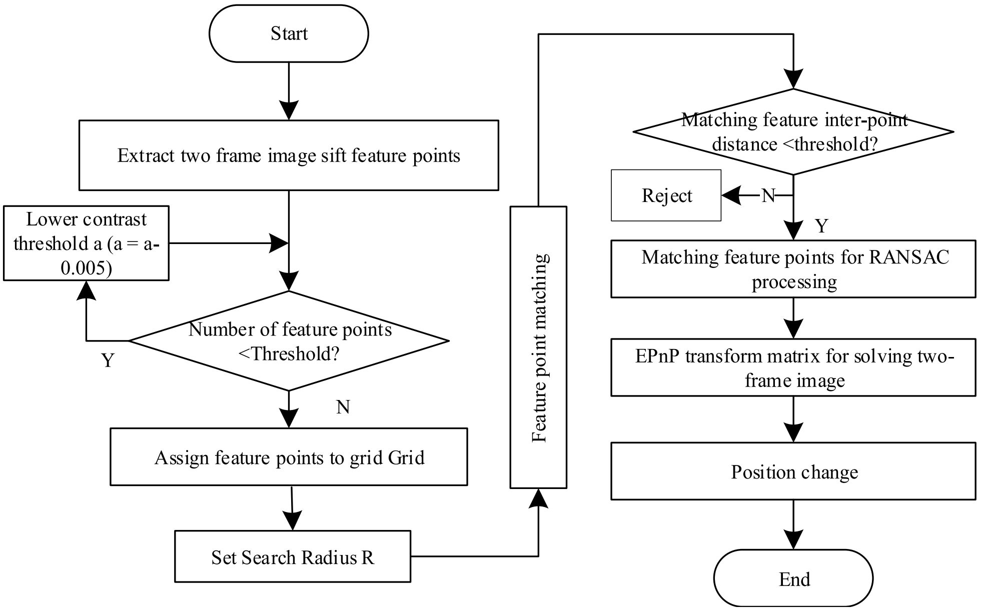
Start
Extract two frame image sift feature points
Lower contrast threshold a (a = a-0.005)
Number of feature points <Threshold?
Y
N
Assign feature points to grid Grid
Set Search Radius R
Feature point matching
Matching feature inter-point distance <threshold?
Reject
N
Y
Matching feature points for RANSAC processing
EPnP transform matrix for solving two-frame image
Position change
End

FIGURE 6. Process of pose estimation based on depth image and thermal infrared image.

of the j feature map in each layer. σ^j is the j feature map in output layer. The h_4^j is weight matrix. Up-sampling can obtain the expanded weight matrix h_5^j. Weighting h_5^j and input can get the fused output σ^j.

C. POSE ESTIMATION BASED ON DEPTH IMAGE AND THERMAL INFRARED IMAGE

The pose estimation process based on the depth image and thermal infrared image is shown in Figure 6.

Firstly, an improved Scale-invariant feature transform (SIFT) algorithm is used to extract the feature points of the image. The feature points are matched under the condition that the feature point threshold is met. Combined with the matched feature points, PnP is used to solve the pose between the thermal infrared images to obtain the 3D position of the image. Due to the phenomenon of overlap and misalignment, a random fern model is used to optimize the pose to obtain an accurate 3D image position.

The thermal imaging system is susceptible to environmental interference, and its own detection capability is low. As a result, the acquired thermal infrared image has low resolution, the image is blurred and there is more noise. Therefore, the SIFT algorithm is used to accurately extract the features of the thermal infrared image to estimate the robot's motion in a low-light environment.

The SIFT algorithm is a robust algorithm for detecting image feature points in computer vision. To extract SIFT features, it needs to convert the input image into a grayscale image. And then, the image gaussian pyramid is obtained through down-sampling and Gaussian convolution, as shown in Figure 7, that is, to obtain images of different scales and different degrees of blur. Then, a difference operation is performed on the images at the same scale to obtain a Gaussian difference pyramid. The detected extreme points are used as feature points. Because the image is transformed at different scales and different blur levels during the extraction of key points, the extracted feature points are minimally affected by factors such as light and rotation.

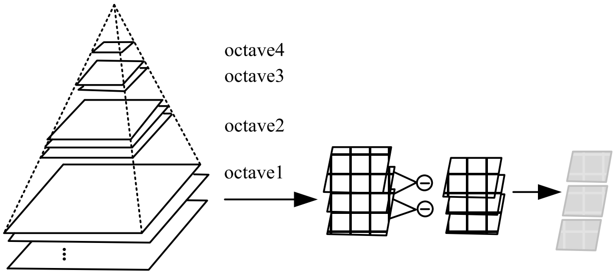

FIGURE 7. SIFT feature point extraction algorithm.

Due to the low resolution of the thermal infrared image, the fuzzy image texture and the fewer feature points, the SIFT extraction algorithm is improved. That is, after obtaining the initial candidate feature points, the SIFT algorithm performs a preliminary screening of the feature points. If the candidate points have a low contrast or are located at the edges, the algorithm will consider them as unstable points and remove them. However, if the contrast of the image is low and the imaging is blurred, a fixed threshold will remove more points, making it difficult to extract sufficient feature points. Therefore, parameters need to be adjusted according to the specific conditions of the image. Experiments show that the curvature threshold has little effect on the number of feature points, but the contrast threshold has a great effect.

The contrast of the image can be represented by texture information and image information entropy. But it has been found through experiments that these feature parameters do not have a particularly obvious functional relationship with the number of feature points. Therefore, the proposed method sets the minimum number of SIFT feature points (800) extracted by each frame of the image. If the number of extracted feature points is less than the threshold, the contrast threshold is decreased by 0.005 until the number of feature points is higher than the set value.

According to the assumption of smooth lateral motion, the images to be detected are introduced into a grid and divided into groups. Firstly, each grid is treated as an independent region. Adaptive feature point detection is performed on each independent region. Then, detected feature points are sorted by their response values. Finally, according to the pre-set value, the best feature points are selected from the feature point set, thus the detection of all areas of the whole image is completed. The algorithm flow is shown in Figure 8.

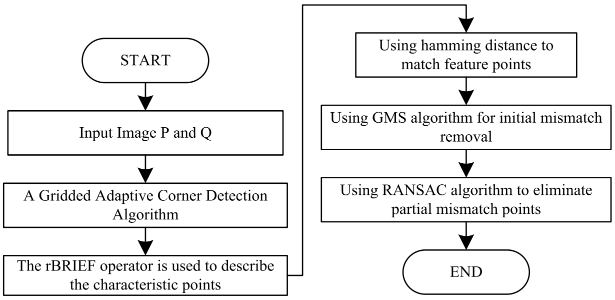
START
Input Image P and Q
A Gridded Adaptive Corner Detection Algorithm
The rBRIEF operator is used to describe the characteristic points
Using hamming distance to match feature points
Using GMS algorithm for initial mismatch removal
Using RANSAC algorithm to eliminate partial mismatch points
END

FIGURE 8. Algorithm flow of feature point matching.

Specific steps are as follows:

Step 1: Step 1: Set the maximum number of feature points for extraction to Q and the number of row and column of the grid to G_{col}, G_{row} respectively. Then, the number of pre-detected points of each grid is:

$$N = Q/\left(G_{col}G_{row}\right) \tag{3}$$

Step 2: According to the area divided by the grid, the coordinate position of the grid border of the $i(i = 1, 2, \cdots, G_{row}G_{col})$ area is:

$$\begin{cases} R_{row} = \left(\dfrac{(i/G_{row})\, w}{G_{row}}, \dfrac{((i/G_{row})+1)\, w}{G_{row}}\right) \\ R_{col} = \left(\dfrac{(i-(i/G_{row})\, G_{col})\, h}{G_{col}}, \dfrac{((i-(i/G_{row})\, G_{col})+1)\, h}{G_{col}}\right) \end{cases} \tag{4}$$

In the formula, the width and length of the image are w, h respectively.

The region is detected by a corner detection algorithm and the difference between the gray values of each candidate corner point and the surrounding 16 neighboring points is calculated by using the oFAST algorithm. If the number of pixel points M $>$ 12 is satisfied, the candidate point is considered as corner point.

$$M = \sum_{x\in c(p)} \left|I_x - I_p\right| > \alpha \left\{\frac{1}{n}\sum_{i=1}^{n}\left[I\left(x_i\right) - I\left(\bar{x}\right)\right]^2\right\} \tag{5}$$

where $c\,(p)$ is the area composed of 16 pixels adjacent to the candidate corner point p. I_p is the gray value of the candidate corner pint p to be measured. I_x is the image gray value of the 16 adjacent pixels of the candidate corner point p. n is the number of pixels in the area. α is the scale factor and $I\,(x_i)$, $I\,(\bar{x})$ are the gray value of each pixel in the area to be detected and the average gray value of the area.

Step 3: If the number of detected corner points is less than the set number of pre-detection points N, save and execute the detection of the next area, otherwise go to step 4.

Step 4: If the number of detected feature points is greater than or equal to N, the response values of all detected feature points are calculated by Harris corner response and then sorted in descending order according to the size of the response value. The top N best points are filtered out and saved from the feature point set.

PnP (Perspective-n-Point) is an important method to solve the pose of 3D-2D matching points because it requires fewer matching point pairs and has higher accuracy. Since the obtained thermal infrared image and the depth image have a one-to-one correspondence, the 3D positions of the feature points can be obtained from the depth image. Therefore, PnP can be used to solve the pose between the thermal infrared images. There are many methods to solve PnP problem. Because EPnP is fast and has high accuracy, EPnP is used to solve PnP problem.

The EPnP algorithm represents the 3D points in the world coordinate system as a weighted sum of a set of virtual control points. For general cases, the EPnP algorithm requires the number of control points to be 4 and these 4 control points cannot be coplanar. Assume that the feature point x_m^t in the image I_t is 3D coordinate X_q^t, $p = 1, 2, 3, \cdots, Q$ of the camera coordinate system and the feature point matching in the image I_{t+1} is 2D coordinate X_q^{t+1}, $q = 1, 2, 3, \cdots, Q$, and the coordinates of the four control points of the corresponding camera coordinates in I_t are C_i^t, $i = 1, 2, 3, 4$ and are expressed in the camera coordinate system as C_i^{t+1}, $i = 1, 2, 3, 4$.

3D feature point X_q^t is represented by 4 control points:

$$X_p^t = \sum_{i=1}^{4} \alpha_{ti} C_i^t, \quad \sum_{i=1}^{4} \alpha_{ti} = 1 \tag{6}$$

Assuming the relative pose of two frames is $[R\,|T\,]$, the relationship between the control points is:

$$C_i^{t+1} = [R\,|T\,]\, C_i^t \tag{7}$$

As can be seen from the above formula, solving the pose between cameras requires determining the control points in the two-frame camera coordinate system, that is, calculating the coordinates of the control point C_i^{t+1} in camera coordinate I_{t+1}.

The coordinate relationship between the 2D feature point x_q^{t+1} and the control point C_j^{t+1} is obtained according to the projection model of the camera:

$$\begin{gathered}
\forall p, \quad w_p \begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix} \sum_{i=1}^{4} \alpha_{ti} \begin{bmatrix} X_i^{t+1} \\ Y_i^{t+1} \\ Z_i^{t+1} \end{bmatrix} \\
\sum_{i=1}^{4} \alpha_{ti} f_u X_i^{t+1} + \alpha_{ti} \left(u_c - u_p\right) Z_i^{t+1} = 0 \\
\sum_{i=1}^{4} \alpha_{ti} f_v Y_i^{t+1} + \alpha_{ti} \left(v_c - v_p\right) Z_i^{t+1} = 0
\end{gathered} \tag{8}$$

By connecting q feature points in series, we get:

$$Mx = 0, \quad x = \left[C_1^{t+1},\ C_2^{t+1},\ C_3^{t+1},\ C_4^{t+1}\right]^T \tag{9}$$

Solving the above formula can get the control point coordinates and using the obtained control point coordinate, the 2D coordinate point x_q^{t+1} can be represented as X_q^{t+1} in 3D.

In the process of map creation, only the map optimization pose optimization method is used, and it has not been possible to identify areas that have been reached and complete the global closed loop of system. The effect of back-end optimization on the closed loop will gradually diminish over time and during the creation of large scene map. When returning to the area have reached or repeating the local mapping, overlap and dislocation will occur.

To solve the above problem, a random fern model is used in the optimized camera pose model. It uses image coding to determine the similarity between the two frames. The encodings of *mblock* constitute the encoding of each image C, and each *block* consists of *nferns*. Each *ferns* is determined by comparing the pixel value θ at each channel x with a threshold. The calculation of encoded key frames is as follows:

$$C = \{block_k\}_{k=1}^{m} \rightarrow block = \{ferns\}_{i=1}^{n} \tag{10}$$

Encode the obtained key frames. The key frame and its corresponding depth image have a total of four channels. The pixel value of each channel is compared with the selected threshold value to calculate the value of the *ferns* list in each *block*.

TABLE 1. Similarity matrix construction algorithms.

Image coding	ID number of key frames
……	……
{1010}	(0,3,5)
{1011}	(1,4)
{1100}	(2,7)
……	……

The pixel values of the 4 channels of the obtained image are compared with the selected threshold, and the code value of *block* is {1100}. The relationship between the image code value and the key frame ID is shown in Table 1. The left column of Table 1 is the calculated image code and the right column is the key frame number corresponding to the image code. If the same image encoding exists between key frames, the similarity between the two is high. By comparing neighboring keyframes with historical keyframes, the similarity between this image and all images can be calculated. Determine the similarity between key frames and decide whether the current key frame is added to the loopback. If there is a loopback, then register with the keyframe with the similarity and perform relocation.

$$\begin{gathered}
f\left(I, \theta, \tau\right) = \begin{cases} 1, & I\left(\theta\right) \geq \tau \\ 0, & I\left(\theta\right) < \tau \end{cases} \\
\theta = \{c, x\}
\end{gathered} \tag{11}$$

D. 3D SEMANTIC MAP FUSION UPDATED BY BAYESIAN

The three-dimensional indoor semantic map construction method combines pixel-level image semantic segmentation and three-dimensional simultaneous positioning based on deep learning with mapping. The three-dimensional scene reconstruction and image semantic segmentation discussed above are organically integrated into a whole through a fusion algorithm updated by Bayesia. The result of neural network recognition and 3D point cloud map are integrated into a unified semantic map to realize the construction of 3D semantic map for mobile robots [34].

When collecting data through sensors, semantic segmentation usually lacks consistency in consecutive adjacent key frames in an unknown environment due to its instability. Therefore, progressive semantic label fusion is used to associate semantic labels of multiple key frames and image semantic labels are updated in time with the association of scenes. If the current key frame is K_t, the distribution of semantic labels on 3D voxel V_d of K_t is l_k [35]. It needs to get the independent probability distribution of each 3D voxel on its semantic label set $P\left(v_d \to l_k \left| k_0^t\right.\right)$. The current key frame set is $K_0^t = \{K_0, K_1, \cdots, K_t\}$ and the recursive Bayesian is:

$$P\left(v_d \to l_k \left| k_0^t\right.\right) = P\left(K_t \left| K_0^{t-1}, l_k\right.\right) P\left(l_k \left| K_0^{t-1}\right.\right) \times \frac{1}{P\left(K_t \left| K_0^{t-1}\right.\right)} \tag{12}$$

Using Markovian assumptions on $P\left(K_t \left| K_0^{t-1}, l_k\right.\right)$ can get:

$$P\left(v_d \to l_k \left| k_0^t\right.\right) = \frac{P\left(K_t\right) P\left(l_k \left| K_t\right.\right)}{P\left(l_k\right)} P\left(l_k \left| K_0^{t-1}\right.\right) \times \frac{1}{P\left(K_t \left| K_0^{t-1}\right.\right)} \tag{13}$$

Assuming that $P\left(l_k\right)$ is not related to the time variable, when acquiring the next key frame image, the semantic label category of the 3D voxel can be updated by the following formula:

$$P\left(v_d \to l_k \left| k_0^t\right.\right) \sim P\left(l_k \left| K_0^t\right.\right) P\left(l_k \left| K_0^{t-1}\right.\right) \tag{14}$$

V. EXPERIMENTAL RESULTS AND ANALYSIS

To measure the effectiveness and feasibility of the proposed method, target detection experiment, image semantic segmentation experiment, feature matching experiment and semantic map construction experiment were performed respectively.

A. TARGET DETECTION EXPERIMENT

To verify the performance of the target detection method based on the improved YOLOv3, the proposed method is compared with the methods in [24], [25], and [27] in terms of detection accuracy and detection speed. Among them, the accuracy ratio, recall rate, harmonic mean and processing speed (frame/s) of evaluation index selection of detection accuracy are calculated as follows:

$$\begin{aligned} P &= \frac{T_P}{T_P + F_P} \times 100\% \\ R &= \frac{T_P}{T_P + F_N} \times 100\% \\ F &= \frac{2PR}{P + R} \times 100\% \end{aligned} \tag{15}$$

T_P is the number of targets that detected correctly and F_P is the number of mistakenly detected non-target as targets. F_N is the number of mistakenly detected targets as background.

The statistics of each index in the target detection experiment are shown in Table 2.

TABLE 2. Analysis of evaluation indexes of different methods.

Method	P /%	R /%	F /%	v_0 /(frame/s)
reference [24]	86	87	86.5	21
reference [25]	90	90	90	7
reference [27]	92.5	90.3	91.4	47
The proposed method	98.1	93.7	95.9	51

It can be seen from the above table that the proposed method has improved the accuracy and recall of target detection by 5.6 and 3.4 percentage points, respectively, compared with the target detection model in [27] and the average of the reconciliation has increased by 4.5 percentage points. The average speed improves 4 frames/s. The other two target detection methods have a high number of falsely detected non-targets as targets in the test sample set, which affects the overall accuracy. The average processing speed is much lower than the proposed method. Therefore, compared with other methods, the proposed method has better performance in target detection.

B. IMAGE SEMANTIC SEGMENTATION EXPERIMENT

To evaluate the accuracy of the image semantic segmentation model, the Mean Intersection Over Union (MIoU) is used to evaluate the accuracy of the segmented object. The higher the value, the better the segmentation effect. The formula is as follows:

$$MIoU = \frac{1}{l+1} \sum_{j=0}^{l} \frac{TT_j}{FT_j + TF_j - TT_j} \tag{16}$$

In the formula, l is the number of categories and $0 \leq j \leq l-1$. TT_j is the number of correct prediction samples in the j category. TF_j and FT_j are the number of true false and false true samples in the j category, that is, the samples of incorrect prediction.

In real-time semantic segmentation, one of the important indicators is the prediction speed of the network. If the time complexity is too high, it will cause a lot of time for model training and prediction, and fast real-time prediction cannot be achieved. Therefore, time complexity determines the model training and prediction time. Its time complexity is calculated as follows:

$$T = \sum\nolimits_{i=0}^{Co} M_{1,i} \cdot N_{1,i} \cdot \left(K_{1,i}\right)^2 \cdot C_{1,i}^{i} \cdot C_{1,i}^{0} + \sum\nolimits_{j=0}^{De} M_{2,j} \cdot N_{2,j} \cdot C_{2,j}^{i} \cdot \left(\left(K_{2,j}\right)^2 + C_{2,j}^{0}\right) \quad (17)$$

where Co is the number of regular convolution in the model and in the i regular convolution layer, $M_{1,i}$ and $N_{1,i}$ are the length and width of the output feature map. $K_{1,i}$ is the dimensions of the convolution kernel. $C_{1,i}^{i}$ is the number of channels input by the convolution layer. $C_{1,i}^{0}$ is the number of channels output by the layer. De is the number of model depth-separated convolutions. In the j depth-separated convolution layer, $M_{2,j}$ and $N_{2,j}$ are the length and width of the output feature map. $K_{2,j}$ is the dimension of the convolution kernel. $C_{2,j}^{i}$ is the number of channel input by the convolution layer. $C_{2,j}^{0}$ represents the number of output channels of the convolution layer.

To verify the effectiveness of the proposed method, the proposed model and the comparison model SkipNet were trained 24000 times on the CityScapes dataset. The accuracy comparison of each category of semantic segmentation is shown in Figure 9. The CityScapes dataset contains 19 image categories (roads, buildings, signs, sky, cars, etc.).

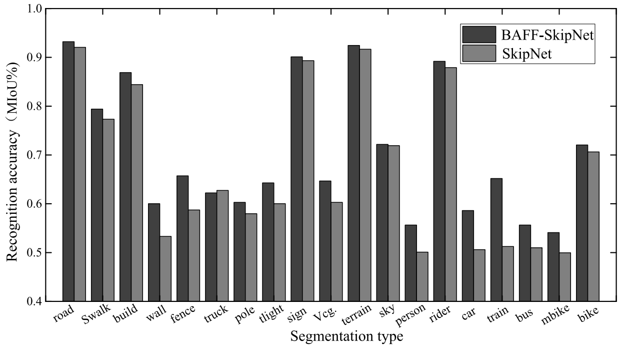

FIGURE 9. Accuracy comparison of semantic segmentation categories.

As can be seen from the figure above, compared with other methods, the proposed method improves the accuracy of object segmentation greatly, which is mainly reflected in the types with complex features, such as: rider, mbike (motorcycle), train (train), wall (wall), fence (fence), etc. The reason is that the proposed BAFF method can adaptively fuse shallow and deep features.

Further, the time complexity of the model is analyzed. The experimental results obtained are shown in Table 3.

From the experimental comparison results, it can be found that the proposed model has high segmentation accuracy with a small number of parameters, and the computing time of the model is relatively small. Therefore, the proposed model achieves a good balance between the fastness and accuracy of image segmentation. It is a real-time semantic segmentation algorithm with excellent performance.

TABLE 3. Accuracy comparison of real-time semantic segmentation model.

method	MIoU /%	time /ms	Parameter quantity / 10^6
Reference [28]	68.1	19.56	2.18
Reference [29]	57.5	11.89	0.33
The proposed method	71.3	19.02	0.84

C. FEATURE MATCHING EXPERIMENT

To verify that the proposed feature matching method can improve the matching quality and effectively eliminate false matching, it is compared with the methods in [12] and [19] in terms of matching accuracy and running time. Six groups of images in different environments were randomly selected for the experiment, and the results are shown in Table 4.

TABLE 4. Comparison of matching number and running time of different methods.

group	Reference [12]		Reference [19]		The proposed method	
	Matching number	time/s	Matching number	time/s	Matching number	time/s
1	256/0	59	237/0	58	214/0	50
2	49/2	58	42/1	57	37/0	49
3	157/3	30	136/2	29	122/0	17
4	658/0	86	614/1	85	573/0	69
5	875/0	82	830/0	79	791/0	63
6	435/0	33	408/0	31	386/0	18

It can be seen from the above table that when there are fewer image feature points, the algorithms in [12] and [19] will have mismatches, and the proposed algorithm can eliminate the mismatch points completely. Although the overall number of matches is slightly less than the algorithms in the other two references, it guarantees enough accuracy. In addition, the proposed algorithm runs faster than the other two matching algorithms, mainly because the corner detection algorithm used in the proposed method is faster. compared with the methods of [12] and [19], the running time of overall algorithm increases by 21.6% and 23.2% respectively, which has a greater advantage in real-time performance.

D. SEMANTIC MAP CONSTRUCTION EXPERIMENT

The platform of this experiment is equipped with a thermal imaging sensor, a Kinect2 sensor and an industrial computer and wheeled mobile system. According to the resolution and field view of the two vision sensors, two sensors are selected to be installed side by side. Among them, TX2 is selected as the computing platform. Its AI embedded processor with super-computing capabilities is particularly suitable for the calculation of intelligent devices such as robots. The GPU has 256 CUDA cores to support parallel processing and has 8G of memory. The CPU consists of 64-bit CPU cluster with two ARM architectures. Kobuki is selected as the mobile platform. It is the most widely used mobile chassis in mobile robot research. Its load weight is 5kg and the accuracy of

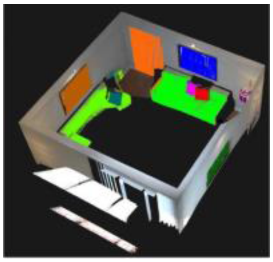

FIGURE 10. Generating 3D point cloud map with color annotation of object instance by experiment.

the encoder is 11.7ticks/mm. The accuracy of the single-axis heading gyroscope is 110deg/s and the maximum moving speed is 70cm/s. The maximum angular velocity is 180deg/s and the mobile power source is a nickel-metal hydride battery with 2200m A. The built-in controller of the chassis is written using the ROS system. Its source code is open and can be freely obtained on the Internet. The chassis can communicate with TX2 through the data cable. The specifications of the camera are shown in Table 5.

TABLE 5. Camera specifications.

sensor	thermal imaging sensor	RGB-D camera
type	DS-2TD2636T-10	Kinect2
field angle	$37.7^0 \times 28.7^0$	$84.1^0 \times 53.8^0$
resolution	388*284	1920*1080
frame rate	25FPS	30FPS
wave length	8-14um	
temperature measurement range	-20~150^{0}C	

In addition, SLAM2 algorithm is mainly implemented by C++ programming. Compared with the socket TCP/IP communication method, time delay is avoided to the greatest extent and time delay processing is not required for data.

3D semantic map construction experiments of indoor scenes were performed for the ICL-NUIM dataset. The data subset is Living Room 'lr kt3', a total of 1242 RGB-D images. The camera circled the room for 42 s. There are 7 types of items in the dataset scene: sofa, pillow, door, TV, mural, table lamp and potted plant. Figure. 10 is a three-dimensional point cloud map containing object color labels. For display convenience, the point cloud in the ceiling area of the point cloud map is removed.

The proposed algorithm is used to solve the pose between adjacent frames, and the result is represented by the rotation matrix R and translation matrix T. The rotation matrix R can be converted into a quaternion q. Let $R = \{m_{ij}\}$, i, $j \in [1, 2, 3]$, then:

$$\begin{aligned} q_0 &= \frac{\sqrt{tr(R)+1}}{2}, \quad q_1 = \frac{m_{23}-m_{32}}{4q_0} \\ q_2 &= \frac{m_{31}-m_{13}}{4q_0}, \quad q_3 = \frac{m_{12}-m_{21}}{4q_0} \end{aligned} \tag{18}$$

Let the real rotation matrix of camera is R_{true} and its quaternion is expressed as q_{true}. The real translation matrix is T_{true}. Then, the rotation error between the two frames is:

$$\begin{aligned} E_{rot} &= \|q_{true} - q\|_2 \\ E_{trans} &= \|T_{true} - T\|_2 \end{aligned} \tag{19}$$

In the experiment, three scenes were selected for evaluation. In scene one, the difference of temperature is smaller. The contrast of the image is low. The temperature difference in scene two is increased and the sharpness of the image is high. The temperature difference in scene three is more obvious. Feature points were extracted and matched for the three scenes respectively. After feature point matching and pose estimation optimization, the experimental results were finally obtained.

The proposed algorithm and the ICP algorithm using only depth information are used to calculate the motion estimation between the images and the experimental results obtained by the two algorithms are analyzed using the error formula. The robot can record the camera movement through the wheeled motion system during the movement. The error is shown in Figure 11. Meanwhile, the time consumption of the two algorithms is compared. The result is shown in Figure 12. It can be seen from the result that the compute time of the pose estimation algorithm based on the feature point method is shorter than the ICP algorithm and it is more conducive to real-time operation.

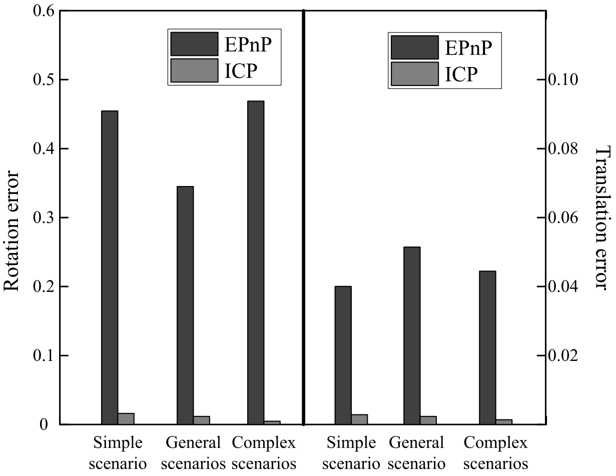

FIGURE 11. Error comparison chart of pose estimation.

To more fully verify the pose estimation algorithm based on the thermal infrared image, 150 frames of continuous images were collected. The continuous RGB, depth and thermal infrared images were recorded when the images were collected. And the registered RGB and depth images were input to ORB-SLAM2 algorithm to calculate the pose of each frame as the true value of this experiment. Then, the proposed algorithm was used to estimate the pose of each frame of the image. Compare the obtained result with the "true value" obtained by the ORB-SLAM2 algorithm and calculate

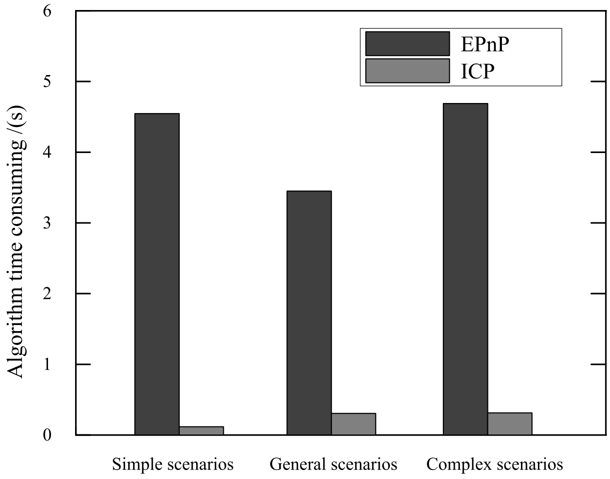

FIGURE 12. Algorithm time comparison.

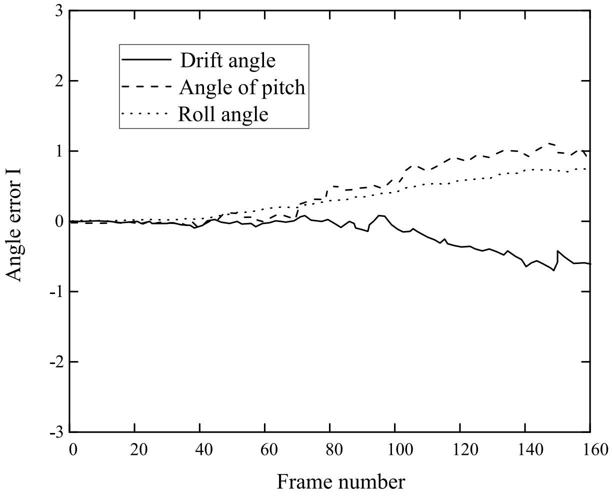

FIGURE 13. Rotation angle error.

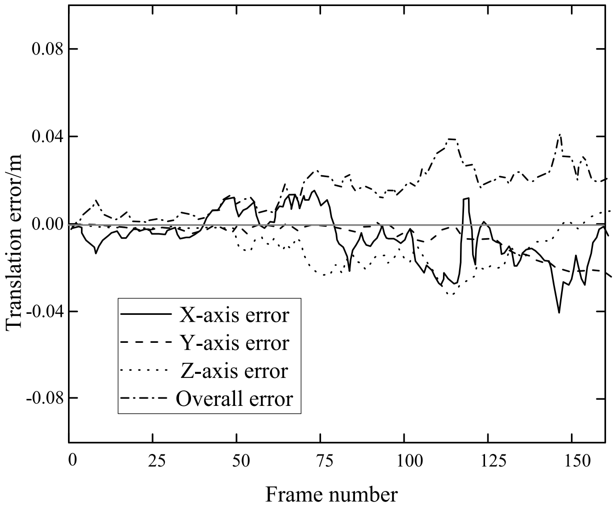

FIGURE 14. Translation error of motion.

rotation and translation errors. To indicate the magnitude of the error, Euler angle is selected to represent the rotational movement. The yaw angle, pitch angle and roll angle are calculated and the differences are calculated. The calculation result is shown in Figure 13. Then calculate the translation errors in the X, Y, and Z directions respectively, and the error result is shown in Figure 14.

It can be seen from the rotation and translation error graphs that the angle error is within 1 °. The specified error index is within 1.5 ° and it meets the requirement. The overall translation error is about 0.04m and it meets the requirement that the specified error is within 0.05m. However, there is still a tendency for error to gradually increase due to cumulative errors. Therefore, based on local pose optimization, back-end optimization is needed to reduce cumulative error.

VI. CONCLUSION

To ensure that mobile robots have a certain ability to sense target in indoor environment and improve the degree of intelligence of robots, based on edge computing environment, a 3D semantic map construction method of mobile robots using improved ORB-SLAM2 is proposed. Combining RGB-D camera with thermal imaging sensor, it performs object detection, image semantic segmentation, feature point matching, pose estimation and optimization based on the acquired image to build 3D semantic map of mobile robot. Finally, the accuracy of the proposed method in pose estimation and robot motion and the efficiency of the algorithm are demonstrated through experiments, which improves the positioning stability of the robot in low light and other environments. It can be applied to actual indoor environments.

However, the pose estimation based on the thermal infrared image and the depth image is currently limited to a small area and a region with obvious features. And the positioning effect for a large range of scene needs to be improved. In addition, the accuracy and generalization ability of semantic segmentation of indoor object need to be improved. The segmentation result can add other attribute associations, so that the robots understand and use the environment can more fully. In future work, we will explore and improve the proposed 3D semantic map construction method for mobile robots.

REFERENCES

[1] M. Mende, M. L. Scott, J. van Doorn, D. Grewal, and I. Shanks, "Service robots rising: How humanoid robots influence service experiences and elicit compensatory consumer responses," *J. Marketing Res.*, vol. 56, no. 4, pp. 535–556, Aug. 2019.

[2] Y. F. Zhou and N. Chen, "The LAP under facility disruptions during early post-earthquake rescue using PSO-GA hybrid algorithm," *Fresenius Environ. Bull.*, vol. 28, no. 12A, pp. 9906–9914, 2019.

[3] M. Jörling, R. Böhm, and S. Paluch, "Service robots: Drivers of perceived responsibility for service outcomes," *J. Service Res.*, vol. 22, no. 4, pp. 404–420, Nov. 2019.

[4] M. Labbé and F. Michaud, "RTAB-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *J. Field Robot.*, vol. 36, no. 2, pp. 416–446, Mar. 2019.

[5] M. Sualeh and G.-W. Kim, "Simultaneous localization and mapping in the epoch of semantics: A survey," *Int. J. Control, Autom. Syst.*, vol. 17, no. 3, pp. 729–742, Mar. 2019.

[6] Q. Shi, S. Zhao, X. Cui, M. Lu, and M. Jia, "Anchor self-localization algorithm based on UWB ranging and inertial measurements," *Tsinghua Sci. Technol.*, vol. 24, no. 6, pp. 728–737, Dec. 2019.

[7] H. Gao, W. Huang, and X. Yang, "Applying probabilistic model checking to path planning in an intelligent transportation system using mobility trajectories and their statistical data," *Intell. Autom. Soft Comput.*, vol. 25, no. 3, pp. 547–559, 2019.

[8] R. Gomez-Ojeda, F.-A. Moreno, D. Zuniga-Noel, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 734–746, Jun. 2019.

[9] M. J. Schuster, K. Schmid, C. Brand, and M. Beetz, "Distributed stereo vision-based 6D localization and mapping for multi-robot teams," *J. Field Robot.*, vol. 36, no. 2, pp. 305–332, Mar. 2019.

[10] L. Han, L. Xu, D. Bobkov, E. Steinbach, and L. Fang, "Real-time global registration for globally consistent RGB-D SLAM," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 498–508, Apr. 2019.

[11] X. Ma, H. Gao, H. Xu, and M. Bian, "An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, p. 249, Nov. 2019, doi: 10.1186/s13638-019-1557-3.

[12] H. Gao, W. Huang, Y. Duan, X. Yang, and Q. Zou, "Research on cost-driven services composition in an uncertain environment," *J. Internet Technol.*, vol. 20, no. 3, pp. 755–769, 2019.

[13] D. Zou, P. Tan, and W. Yu, "Collaborative visual SLAM for multiple agents: A brief survey," *Virtual Reality Intell. Hardw.*, vol. 1, no. 5, pp. 461–482, Oct. 2019.

[14] H. Gao, Y. Duan, L. Shao, and X. Sun, "Transformation-based processing of typed resources for multimedia sources in the IoT environment," *Wireless Netw.*, Nov. 2019, doi: 10.1007/s11276-019-02200-6.

[15] H. Gao, Y. Xu, Y. Yin, W. Zhang, R. Li, and X. Wang, "Context-aware QoS prediction with neural collaborative filtering for Internet-of-Things services," *IEEE Internet Things J.*, early access, Dec. 2, 2019, doi: 10.1109/JIOT.2019.2956827.

[16] J. Cheng, Y. Sun, and M. Q.-H. Meng, "Improving monocular visual SLAM in dynamic environments: An optical-flow-based approach," *Adv. Robot.*, vol. 33, no. 12, pp. 576–589, Jun. 2019.

[17] C.-C. Tsai, C.-F. Hsu, X.-C. Lin, and F.-C. Tai, "Cooperative SLAM using fuzzy Kalman filtering for a collaborative air-ground robotic system," *J. Chin. Inst. Engineers*, vol. 43, no. 1, pp. 67–79, Jan. 2020.

[18] Y. Chen, G. Wang, and L. Wu, "Research on feature point matching algorithm improvement using depth prediction," *J. Eng.*, vol. 2019, no. 23, pp. 8905–8909, Dec. 2019.

[19] Q. Fu, H. Yu, L. Lai, J. Wang, X. Peng, W. Sun, and M. Sun, "A robust RGB-D SLAM system with points and lines for low texture indoor environments," *IEEE Sensors J.*, vol. 19, no. 21, pp. 9908–9920, Nov. 2019.

[20] H. Liao, "Facial age feature extraction based on deep sparse representation," *Multimedia Tools Appl.*, vol. 78, no. 2, pp. 2181–2197, Jan. 2019.

[21] Q. An, Z. Pan, L. Liu, and H. You, "DRBox-v2: An improved detector with rotatable boxes for target detection in SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8333–8349, Nov. 2019.

[22] M. Geng, S. Shang, B. Ding, H. Wang, and P. Zhang, "Unsupervised learning-based depth estimation-aided visual SLAM approach," *Circuits, Syst., Signal Process.*, vol. 39, no. 2, pp. 543–570, Feb. 2020.

[23] J. Li, Z. Li, Y. Feng, Y. Liu, and G. Shi, "Development of a human–robot hybrid intelligent system based on brain teleoperation and deep learning SLAM," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1664–1674, Oct. 2019.

[24] F. Wang, H. Wang, H. Wang, G. Li, and G. Situ, "Learning from simulation: An end-to-end deep-learning approach for computational ghost imaging," *Opt. Express*, vol. 27, no. 18, p. 25560, Sep. 2019.

[25] J. Hu, J. Fang, Y. Du, Z. Liu, and P. Ji, "A security risk plan search assistant decision algorithm using deep neural network combined with two-stage similarity calculation," *Pers. Ubiquitous Comput.*, vol. 23, nos. 3–4, pp. 541–552, Jul. 2019.

[26] J. Deng, A. Roussos, G. Chrysos, E. Ververas, I. Kotsia, J. Shen, and S. Zafeiriou, "The menpo benchmark for multi-pose 2D and 3D facial landmark localisation and tracking," *Int. J. Comput. Vis.*, vol. 127, nos. 6–7, pp. 599–624, Jun. 2019.

[27] J. Tang, L. Ericson, J. Folkesson, and P. Jensfelt, "GCNv2: Efficient correspondence prediction for real-time SLAM," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3505–3512, 2019.

[28] G. Tinchev, A. Penate-Sanchez, and M. Fallon, "Learning to see the wood for the trees: Deep laser localization in urban and natural environments on a CPU," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1327–1334, Apr. 2019.

[29] M. Mao, C. Beihang UniversityBeijing, H. Zhang, S. Li, B. Zhang, and C. Shenzhen, "SEMANTIC-RTAB-MAP (SRM): A semantic SLAM system with CNNs on depth images," *Math. Found. Comput.*, vol. 2, no. 1, pp. 29–41, 2019.

[30] Q. Liu and F. Duan, "Loop closure detection using CNN words," *Intell. Service Robot.*, vol. 12, no. 4, pp. 303–318, 2019.

[31] S. J. Lee, H. Choi, and S. S. Hwang, "Real-time depth estimation using recurrent CNN with sparse depth cues for SLAM system," *Int. J. Control, Autom. Syst.*, vol. 18, no. 1, pp. 206–216, Jan. 2020.

[32] T. Nguyen, J. Wozencraft, C. J. Taylor, V. Kumar, S. S. Shivakumar, I. D. Miller, J. Keller, E. S. Lee, A. Zhou, T. Ozaslan, G. Loianno, and J. H. Harwood, "MAVNet: An effective semantic segmentation micro-network for MAV-based tasks," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3908–3915, Oct. 2019.

[33] W. Zhou, A. Zyner, S. Worrall, and E. Nebot, "Adapting semantic segmentation models for changes in illumination and camera perspective," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 461–468, Apr. 2019.

[34] X. Yang, Y. Gao, H. Luo, C. Liao, and K.-T. Cheng, "Bayesian DeNet: Monocular depth prediction and frame-wise fusion with synchronized uncertainty," *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2701–2713, Nov. 2019.

[35] R.-T. Wu and M. R. Jahanshahi, "Data fusion approaches for structural health monitoring and system identification: Past, present, and future," *Struct. Health Monitor.*, vol. 19, no. 2, pp. 552–586, Mar. 2020.

XU CUI received the degree from Yanbian University, in 1998. He is currently pursuing the master's degree in computer science. He was with Yanbian University. He is currently an Associate Professor. His research interests include machine learning and intelligent information processing.

CHENGGANG LU received the degree from Yanbian University, in 2003. He is currently pursuing the master's degree in computer science. He was with Yanbian University. He is currently a Lecturer. His research interests include machine learning and intelligent information processing.

JINXIANG WANG received the degree from Yanbian University, in 2002. He is currently pursuing the master's degree in computer science. He was with Yanbian University. He is currently a Lecturer. His research interests include robot and intelligent information processing.