Pattern Recognition 113 (2021) 107822

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Visual SLAM for robot navigation in healthcare facility

Baofu Fang[a,b,c], Gaofei Mei[a], Xiaohui Yuan[e,*], Le Wang[a], Zaijun Wang[d], Junyang Wang[a]

[a] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, 230009, China
[b] Intelligent Interconnected Systems Laboratory of Anhui Province (Hefei University of Technology), Hefei, 230009, China
[c] Anhui Province Key Laboratory of Affective Computing and Advanced Intelligent Machine, (Hefei University of Technology), Hefei, 230009, China
[d] Key Laboratory of Flight Techniques and Flight Safety, Civil Aviation Flight University of China, Guanghan, 618307, China
[e] Department of Computer Science and Engineering, University of North Texas, Denton, TX, 76207, USA

ARTICLE INFO

Article history:
Received 12 August 2020
Revised 27 November 2020
Accepted 22 December 2020
Available online 16 January 2021

Keywords:
COVID-19 pandemic
Visual SLAM
Dynamic scenes
Semantic descriptors
Knowledge graph

ABSTRACT

The COVID-19 pandemic has affected many countries, posing a threat to human health and safety, and putting tremendous pressure on the medical system. This paper proposes a novel SLAM technology using RGB and depth images to improve hospital operation efficiency, reduce the risk of doctor-patient cross-infection, and curb the spread of the COVID-19. Most current visual SLAM researches assume that the environment is stationary, which makes handling real-world scenarios such as hospitals a challenge. This paper proposes a method that effectively deals with SLAM problems for scenarios with dynamic objects, e.g., people and movable objects, based on the semantic descriptor extracted from images with help of a knowledge graph. Specifically, our method leverages a knowledge graph to construct a priori movement relationship between entities and establishes high-level semantic information. Built upon this knowledge graph, a semantic descriptor is constructed to describe the semantic information around key points, which is rotation-invariant and robust to illumination. The seamless integration of the knowledge graph and semantic descriptor helps eliminate the dynamic objects and improves the accuracy of tracking and positioning of robots in dynamic environments. Experiments are conducted using data acquired from healthcare facilities, and semantic maps are established to meet the needs of robots for delivering medical services. In addition, to compare with the state-of-the-art methods, a publicly available dataset is used in our evaluation. Compared with the state-of-the-art methods, our proposed method demonstrated great improvement with respect to both accuracy and robustness in dynamic environments. The computational efficiency is also competitive.

1. Introduction

The introduction of robots to healthcare facilities provides many services for dealing with the coronavirus crisis such as allowing health care workers to remotely take temperatures and measure blood pressure and oxygen saturation from patients hooked up to a ventilator. Robots could also be used to disinfect with ultraviolet light or bring food to the quarantined. These are just a few of dozens of ways robots to reduce contact and the risk of infection during the COVID-19 pandemic.

To enable robots to work in unknown or dynamic environments, simultaneous localization and mapping (SLAM) technology is developed to build a semantic map of healthcare facilities to obtain the spatial layouts and understand the surrounding environment. Using the SLAM technology, a robot uses its sensors in an unknown environment, locates its position and estimates posture through the observed environmental characteristics during the movements, and incrementally builds a map of the interior. Visual SLAM (VSLAM) uses cameras as sensors, including monocular, stereo, and RGB-D cameras. Benefiting from the fast developments of computer vision techniques and high-quality visual sensors [41], VSLAM has received widespread attention [9,24].

However, there are still problems with VSLAM that require further investigation. The existing VSLAM approaches derive no semantic information from the videos, partly because of the complexity of the analysis [14]. Instead, geometric methods are used to track the movements and generate maps, which faces the difficulty of differentiating objects of various geometric entities [8]. This is critical in complex environments such as healthcare facilities, where patients are moved around and equipment is replaced from time to time. For robots to understand the environments and provide better help for the treatment of the COVID-19 pandemic, semantic information needs to be introduced. With the develop-

* Corresponding author.
E-mail address: xiaohui.yuan@unt.edu (X. Yuan).

https://doi.org/10.1016/j.patcog.2021.107822
0031-3203/

ment of convolutional neural networks [11], there are many high-performance networks for image semantic/instance segmentation [12,43]. The combination of these networks and the VSLAM system is helpful to improve the scene understanding ability of robots [31]. In addition, the existing VSLAM methods usually assume that the environment is static. The presence of moving objects in the hospital scene, such as medical staff, patients, and moving medical equipment, greatly affects the accuracy of VSLAM, and may even lead to the failure of traditional methods. Therefore, how to detect dynamic objects and eliminate their influence on pose estimation has become the key to VSLAM's handling of dynamic scenes. Pose estimation means estimating the camera's position and direction.

In this paper, we extend ORB-SALM2 [24]. This paper extends the ORB feature points and proposes a semantic descriptor for map construction. Using semantic descriptor and knowledge graph, a high-level semantic relationship can be established. By that, we detect and remove dynamic objects in the environment, to improve the accuracy of pose estimation. The main contributions of this paper are as follows:

1. Use a knowledge graph to construct the relationship between entities and obtain high-level semantic information.
2. A descriptor of semantic information is constructed by leveraging Mask R-CNN to describe the semantic information around key points and, together with the knowledge graph, accurately detect the dynamic objects.
3. In the process of feature point matching, dynamic feature points are eliminated to improve the accuracy of pose estimation.
4. Extensive evaluation with real hospital scene to build a semantic map and the effectiveness of the algorithm is verified through comparative experiments on the TUM dataset.

The rest of this paper is organized as follows: Section 2 introduces related work, Section 3 details the main work, Section 4 presents experimental results, and Section 5 concludes the paper with a summary.

2. Related work

In the treatment of the COVID-19 pandemic, researchers have proposed methods of using robot-assisted medical treatment. Ye et al. [39] evaluated the feasibility of using a robot-assisted remote ultrasound system for cardiopulmonary assessment of COVID-19 patients. Yang et al. [38] developed the use of wearable motion capture devices to remotely control robots to complete tasks such as medicine delivery and remote operation of medical equipment. The robots are equipped with a remote video conferencing system for doctor-patient communication. Robots rely on maps to move and need to understand the surroundings to complete various advanced tasks. This paper establishes a semantic map through visual SLAM technology to serve the application of robots in COVID-19 treatment.

Over the past decades, researchers have developed many methods in the field of visual SLAM. Traditional visual SLAM systems, such as PTAM [18], Fang [10], SPM-SLAM [25] and UcoSLAM [23], are based on geometry properties of the image. However, these methods ignore the semantic information from the surroundings. It is, therefore, difficult to accomplish complex tasks such as navigating within high-traffic facilities, e.g., hospitals. In recent years, semantic SLAM that associates geometric entities with semantic information has drawn the attention of researchers. McCormac et al. [22] obtain semantic information through neural network segmentation and use incremental semantic label fusion to build a semantic map based on ElasticFusion [37] framework. Zhao et al. [42] use 2D texture information and 3D geometric information jointly to build a novel semantic segmentation network to get more accurate segmentation results. Hu et al. [13] employ deep networks that leverage 3D features achieve an improved classification and identification. The above methods only focus on semantic mapping, while semantic information is not well used in other parts of SLAM. Alonso et al. [1] construct a lightweight semantic segmentation network, namely MiniNet, and make use of it for keyframe selection. However, their methods regard objects segmented by neural networks as independent individuals, with no connections between them. In this paper, connections between objects will be established using a knowledge graph. Thus we can obtain high-level semantic information.

The majority of the approaches are based on the assumption of static environments. When the space is highly dynamic, e.g., people moving around, the existing methods tend to have a serious deviation in pose estimation and trajectory tracking. To deal with dynamic scenarios, Klappstein et al. [17] calculate motion information between frames by optical flow method to detect moving objects. N. D. Reddy et al. [27] add support for stereo cameras by extending the method in [17]. Tan et al. [34] projected the map onto the current frame to verify the consistency of appearance and structure, thereby detecting changes in the scene.

In addition to the above methods, researchers have also proposed methods for processing SLAM problems in dynamic scenes using semantic information. Kaneko et al. [15] use the segmentation results of DeepLab v2 [6] to remove feature points detected in the sky and cars. Berta et al. [4] combine multi-view geometric models with MASK R-CNN segmentation results to detect and exclude dynamic objects. Riazuelo et al. [28] deal with human activities in the scene by recognizing and tracking "people". Sun et al. [33] detect the dynamic area by calculating the difference between the current and last frame. Kim and Kim [16] propose a robust background model-based dense-visual-odometry (BaMVO) algorithm in dynamic environments to reduce the impact of dynamic objects. Li and Lee [19] apply a static weighting method for edge points in the keyframes. The static weight indicates the likelihood of one point being part of a static environment. Yu et al. [40] combine SegNet [2] with motion consistency detection to filter dynamic feature points and build a static semantic map. Wang et al. [35] not only exclude dynamic objects but also constructed a unified framework for mutual promotion of SLAM localization and semantic segmentation, which improved the accuracy of both. In this paper, we directly describe the surrounding of objects using a semantic descriptor, which is used to detect and remove dynamic objects. In contrast, the aforementioned methods face problems of handling complex scenarios due to the lack of the ability of semantic analysis.

3. Method

This section presents our method in detail from five aspects. Firstly, the overall framework of our system based on the ORB-SLAM2 is described. Secondly, we introduce the establishment of a knowledge graph in the hospital scene to build a priori movement relationship between objects. Thirdly, a detailed description of the Semantic Descriptor established on the semantic segmentation network, which combines ORB features to describe the semantic information of frames, is presented. Subsequently, the knowledge graph is combined with semantic descriptors to detect and remove dynamic objects to improve the accuracy of pose estimation. Finally, we introduce our idea of constructing a semantic map.

3.1. System framework

The ORB-SLAM2 system has an excellent performance in most cases and achieves a good balance of real-time performance and

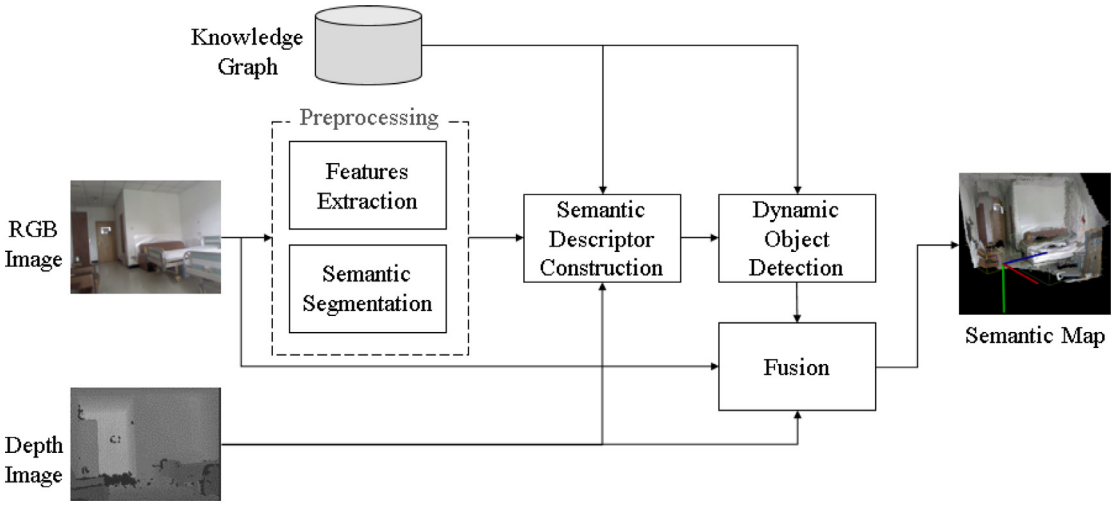


Fig. 1. Framework diagram of our system.

accuracy. Therefore, this paper chooses to work on the ORB-SLAM2 framework with an RGB-D camera in the hospital scene. The overall framework of our system is shown in Fig. 1.

Before the system runs, the Knowledge Graph is used to establish the relationship between various object entities in advance. Images captured by the Microsoft Kinect camera are processed by the tracking thread and the semantic segmentation thread at the same time. The tracking thread first extracts ORB feature points and then waits for the segmentation results of the semantic segmentation thread. After receiving the segmentation results, the tracking thread constructs semantic descriptors and then combines the pre-established entity relationships to mark dynamic feature points and exclude them, leaving only stable static feature points which are used for feature matching and semantic map building.

3.2. Knowledge graph

During COVID-19 treatment, there is a complicated movement relationship between objects. It is easy to get that: doctors, nurses, etc. are dynamic objects while ventilators, temperature guns, teacups, etc. are static objects; however, the process of temperature guns being taken away by nurses is also dynamic. In the past, semantic SLAM regards objects segmented by neural networks as independent individuals so only low-level semantic information can be obtained.

This low-level semantic information can easily recognize dynamic objects like doctors and nurses, but the situation of "a temperature gun was taken away by the nurse' needs to be handled by combining with other geometric methods. This paper intends to establish relationships between objects by building a knowledge graph, to process complex dynamic scenes, which is a type of higher-level semantic information. A knowledge graph is a structured representation of facts, consisting of entities, relationships, and semantic descriptions. We can define a knowledge graph as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$, where $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{F}$ are sets of entities, relations and facts, respectively. A fact is denoted as a triple of head (h), relation (r) and tail (t): $(h, r, t) \in \mathcal{F}$, where $h \in \mathcal{E}$, $r \in \mathcal{R}$ and $t \in \mathcal{E}$.

For ease of description, the set of feature points belonging to "objects with motion attributes" in the image is marked as P_{moving}, such as doctors, nurses, etc. The set of feature points belonging to "objects with attributes that can be moved" is marked as $P_{movable}$, such as temperature guns, cups, etc. These do not have motion attributes themselves, but they can be operated on or moved by medical staff. That is, these objects can be moved by different P_{moving} objects to be in motion. Taking medical staff and some common objects in hospitals, for example, the knowledge graph is established as shown in Fig. 2. Health care personnel, patient, and visitors are part of the set P_{moving}; movable medical equipment belongs to $P_{movable}$; whereas beds and stationary medical equipment are static objects. We refine the relationships between doctors, patients, medical equipment, and others from real hospital scenarios. At present, the knowledge graph established in this paper is sufficient to meet the demand for robot assistants during COVID-19 treatments. However, it is possible to extend the knowledge graph in more complex indoor or outdoor scenarios, such as in a living room or a downtown area.

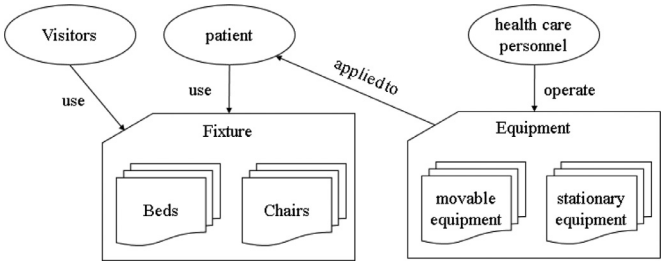


Fig. 2. An illustration of a knowledge graph.

3.3. Local semantic descriptor

In this paper, we have constructed a new type of descriptor, called a local semantic descriptor. Common descriptors, such as the descriptors in the SIFT [7] feature, the SURF [3] descriptor, and the BRIEF [5] descriptor, all describe the photometric information around key points in some forms without semantic information. ORB [29] feature takes BRIEF as its descriptor and obtains direction information. The semantic descriptor proposed is a descriptor that specifically describes the semantic information around key points.

To construct local semantic descriptors, we adopt the MASK R-CNN network to extract semantic information. MASK R-CNN demonstrated highly competitive image instance segmentation accuracy and can obtain instance-level semantic information. MASK R-CNN was trained on the MS COCO dataset [20] and classifies objects of 80 categories. Specifically, we employ the implementation with Tensorflow by Matterport [21].

As shown in Eq. (1), we use a set C to represent all possible object categories, in which c_i $(1 \le i \le m)$ represents the ith object, and c_0 represents the unclassified object. In this paper, m is set to

Fig. 3. Semantic map and 5×5 descriptor. (a) RGB image (b) semantic map (c) semantic descriptor. p and q in (a) are two key points and the zoom-in views of these two points and the neighborhood patches are shown in (c).

80, which is the number of MASK R-CNN classifications.

$$C = \{c_0, c_1, \ldots, c_m\}. \tag{1}$$

For a frame during the SLAM procedure, assume that p is one key point. We take an image patch B_p with side length n and the point p is at its geometric center. We record the object category corresponding to each pixel in B_p to form a semantic descriptor $D^{n\times n}$:

$$D^{n\times n} = \begin{bmatrix} d_{x_1,y_1} & \ldots & d_{x_1,y_n} \\ & \ldots & \\ d_{x_n,y_1} & \ldots & d_{x_n,y_n} \end{bmatrix} \tag{2}$$

where $d_{x_1,y_1} \in C$, $x_i, y_i \in B_p$, and $1 \le i, j \le n$. A semantic descriptor is a matrix, in which each element is a scalar representing object category id.

Fig. 3 shows an image, its semantic segmentation, and an illustration of 5×5 semantic descriptors. The segmented objects, e.g., staff, bed, etc., are marked in different colors and the areas not marked are classified as "background". The black grid shows the semantic descriptors of the two key points P and Q, where each cell records the semantic category c_i $(0 \le i \le m)$ of the pixel in the corresponding position. Among them, point P's category is "bed", and its descriptor includes bed, nurse, and background, while the point Q's category is "person", and its descriptor includes person, bed, and background. We learn from this that the nurse and another person are close to the bed. Hence, the semantic descriptor effectively describes the surroundings of key points.

The semantic descriptor is rotation invariance. In ORB's FAST [30] corner point extraction stage, direction information is obtained. We utilize this information to calculate the rotated semantic descriptor to make it rotation-invariant. This is similar to that of ORB-SLAM.

The direction information of the key points is obtained by the gray centroid method [26]:

1. In an image patch B, the moment is computed as follows:
$$m_{pq} = \sum_{x,y\in B} x^p y^q I(x,y) \tag{3}$$
where $p, q \in \{0, 1\}$ and $I(x, y)$ represents pixel value at coordinate (x, y).
2. Find the centroid of the image patch $(\bar{x}, \bar{y})$ using the moments
$$(\bar{x}, \bar{y}) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{4}$$
3. Draw a line between the centroid and the geometric center of B (key points) to obtain the vector, and calculate the angle between this vector and the x-axis of the image coordinate system:
$$\theta = \arctan\left(\frac{m_{01}}{m_{10}}\right) \tag{5}$$

	x_1	x_2	x_3	x_4	x_5
y_1	c_1	c_1	c_1	c_3	c_3
y_2	c_2	c_1	c_1	c_3	c_3
y_3	c_2	c_2	P	c_3	c_4
y_4	c_2	c_i	c_i	c_4	c_4
y_5	c_i	c_i	c_4	c_4	c_4

	x'_1	x'_2	x'_3	x'_4	x'_5
y'_1	c_1	c_1	c_3	c_3	c_3
y'_2	c_2	c_1	c_1	c_3	c_3
y'_3	c_2	c_2	P'	c_3	c_4
y'_4	c_2	c_i	c_i	c_4	c_4
y'_5	c_i	c_i	c_4	c_4	c_4

Fig. 4. Two examples of the semantic descriptor. The left and right are semantic descriptors of P and P', respectively.

4. rotates the semantic descriptor by θ.

Since the semantic information comes from the segmentation result of the neural network, the semantic descriptor is robust to illumination changes, which extends the ORB descriptor. The illumination changes refer to the changes of indoor lightning and daylight through window or door and the mixture of both. In the case of unstable neural network segmentation results, calculating the distance between semantic descriptors of two frames as auxiliary information can further improve the accuracy of feature points matching. When the accuracy of neural network segmentation is high enough, using only semantic descriptors is enough for matching, i.e., matching is performed by comparing the semantic information around key points. This matching method has high consistency with human judgment methods.

If the semantic descriptor is used for feature matching, the distance between two semantic descriptors is calculated as follows:

$$\sum \mathbb{1}(|f_p(x_i, y_j) - f_{p'}(x'_i, y'_j)| < 1), \tag{6}$$

where $f_p(x_i, y_j), f_{p'}(x'_i, y'_j) \in C$ and $f_p(x_i, y_j)$ and $f_{p'}(x'_i, y'_j)$ represent semantic information of p and p' at (x_i, y_j) and (x'_i, y'_j), respectively. $\mathbb{1}(\cdot)$ is an indicator function, which returns 1 if the distance between f_p and $f_{p'}$ is less than one; otherwise, it returns zero. The computation involves all elements in the two patches. An example is shown in Fig. 4, in which P and P' are two key distinct points in two adjacent frames and c_1, c_2, c_3, c_4, and c_i denote different values. Following the above distance formula, the distance between these two descriptors is 2.

3.4. Dynamic object detection

Detection of dynamic objects starts after the system extracts feature points from a frame and completes semantic segmentation. In a hospital scene, we leverage a priori information to decide dynamic objects. For feature points belonging to P_{moving} are dynamic,

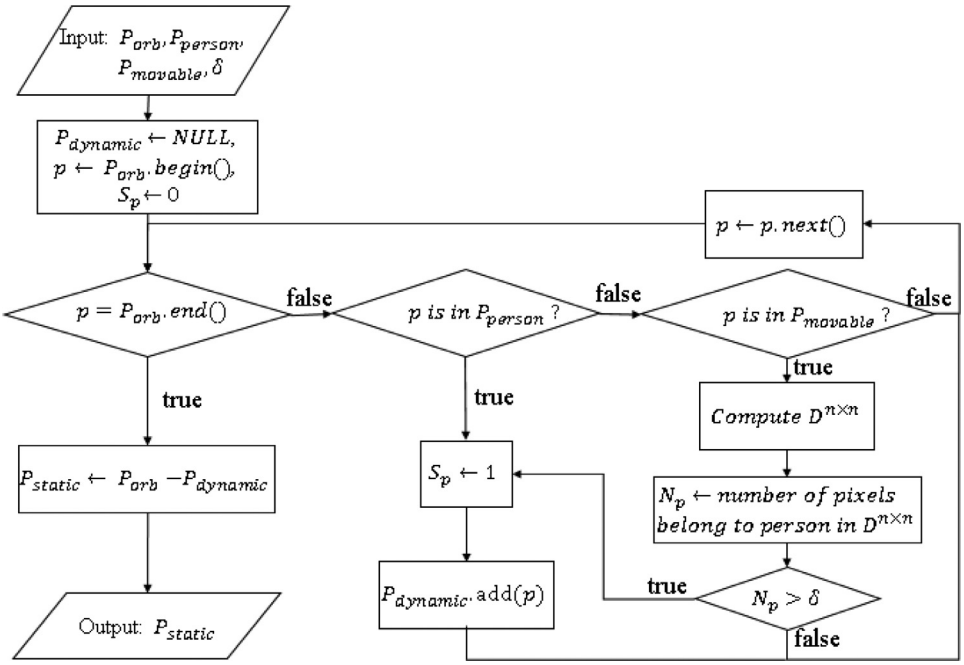


Fig. 5. Flowchart of dynamic object detection and rejection.

they are removed. In case when feature points belonging to $P_{movable}$ are close enough to correspond P_{moving} objects in the knowledge graph, they are also considered to be dynamic. For example, when the temperature gun is very close to a nurse, it is regarded as moving by the nurse.

To decide if $P_{movable}$ objects are close enough to P_{moving} objects, we construct semantic descriptors. For every key point belonging to the $P_{movable}$ in each frame, we construct a semantic descriptor $D^{n \times n}$. N is used to record the number of pixels in it whose semantic category belongs to the ones with motion attributes. Take the category "person" for example, which is represented by c_{person} in Eq. (7), N is calculated. That is, we use semantic descriptors to describe whether there are P_{moving} objects near key points.

$$N = \sum \mathbb{1}(d_{x_i y_j} = c_{person}) \tag{7}$$

We use the semantic descriptor to determine whether the feature point is dynamic. Also take the category "person" for example, as shown in Eq. (8), p represents one feature point in a frame; P_{person} represents the set of points belonging to category "person"; N_p is calculated using Eq. (7); δ represents a threshold; S_p represents the state of the point, where 1 represents dynamic and 0 stands for static. When the feature point category is "person", it is thought dynamic; when the feature point belongs to an object that can be moved by a person, and the number of pixels which are "person" in its semantic descriptor exceeds a given threshold, it is considered to be a dynamic point since it is close enough to people; otherwise, it is considered static. At this point, the labeling of dynamic feature points is completed, and the set of dynamic feature points is recorded as $P^i_{dynamic}$.

$$S_p = \begin{cases} 1, & p \in \{P_{person}, P_{movable}\} \quad \text{and} \quad N_p > \delta \\ 0, & otherwise \end{cases} \tag{8}$$

We remove the marked dynamic feature points from the original ORB feature points set. In Eq. (9), P_{orb} represents the set of all ORB feature points in a frame, and P_{static} represents the set of static feature points. We use P_{static} points to perform feature points matching and pose estimation and complete the follow-up work. In this way, we remove all dynamic feature points when appropriate, and reduce the possibility that $P_{movable}$ objects considered dynamic when they are static. They are dynamic only when they are close enough to objects represented by P_{moving}.

$$P_{static} = P_{orb} - P_{dynamic} \tag{9}$$

The flowchart of the dynamic object detection and rejection process is illustrated in Fig. 5.

3.5. Semantic map construction

For the robot to complete tasks such as delivering medicines and meals, it is necessary to know the type, size, and spatial location of objects in the environment in advance. In response to this need, this paper constructs a dense semantic point cloud map.

We generate an instance-level semantic database and semantic map by the following steps. Firstly, we use the keyframes and the transformation matrix between them along with semantic segmentation results to generate a semantic local point cloud map. Therefore, each point cloud generated has corresponding semantic attributes, such as people, tables, beds, etc. Secondly, we remove the point cloud belonging to the dynamic object according to the method described in Section 3.4 so that a static local point cloud map is created. According to the semantic information, the point cloud cluster of each instance in the segmentation result is obtained. After that, the instance-level semantic database of keyframes is established, including index, category, semantic color, probability, centroid, cluster, and cluster coordinates boundary. The database is used for the robot to quickly retrieve the objects in the current map. To filter out repeated information of the same object in different frames, the database of existing keyframes is updated according to the cluster's semantic category and centroid distance as well as coincidence degree between the current keyframe and the local point cloud map. Whenever a new keyframe arrives, we repeat the above steps to build a semantic map incrementally.

Fig. 6. Examples of semantic maps. From top to bottom, the panels on each row depict the RGB image, the depth image, the rendered point cloud with color, and the semantic map. The left column shows a case with only stationary objects and the right column shows a case with non-stationary objects, in which the non-stationary objects are removed.

4. Experimental Results and Discussion

4.1. Semantic map in hospital scene

In the real hospital scenarios, a total of 15 images sequences were collected by the Kinect camera, including scenes such as ward, corridor, nurse station, etc. They are divided into three parts: isolation area, buffer zone, and clean area. These sequences include dynamic scenes with people walking around and static scenes without people.

Fig. 6 illustrates two scenarios of a static scene and a dynamic scene in each column. The first column depicts a scenario of a ward with no patient or health care personnel and the second column depicts a scenario of a ward with a person. The first two rows present the RGB images and the corresponding depth images. The third row shows the reconstructed 3D views from point cloud textured with color information from the RGB image. The bottom row

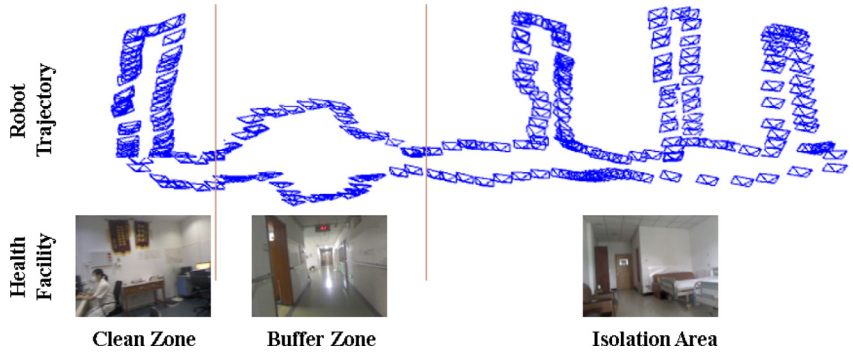


Fig. 7. Robot Trajectory and health facility scenes.

Table 1
Trajectory error in terms of average RMSE (aRMSE), absolute error distance, and median error distance.

	ORB-SLAM2			Ours		
Sequence	aRMSE	Mean (STD)	Median	aRMSE	Mean (STD)	Median
fr3-walking-static	0.3575	0.3243 (0.1490)	0.2870	0.0104	0.0090 (0.0052)	0.0079
fr3-walking-xyz	0.6770	0.5826 (0.3423)	0.5223	0.0164	0.0139 (0.0087)	0.0116
fr3-walking-half	0.5186	0.4567 (0.2424)	0.4282	0.0923	0.0857 (0.0344)	0.0838
fr3-sitting-static	0.0082	0.0072 (0.0039)	0.0066	0.0065	0.0060 (0.0033)	0.0054
fr3-sitting-xyz	0.0094	0.0079 (0.0051)	0.0070	0.0088	0.0079 (0.0043)	0.0070
fr3-sitting-half	0.0205	0.0159 (0.0130)	0.0136	0.0145	0.0125 (0.0074)	0.0115

Table 2
The aRMSE of ATE for TUM RGB-D dataset in dynamic environments.

Sequences	Sun [33]	Li [19]	Wang [35]	Wang [36]	Ours
fr3-walking-static	0.0656	0.0261	**0.0059**	0.3080	0.0104
fr3-walking-xyz	0.0932	0.0601	0.0190	0.3047	**0.0164**
fr3-walking-half	0.1252	0.0489	**0.0285**	0.3116	0.0923
fr3-sitting-static	-	-	0.0078	0.0066	**0.0065**
fr3-sitting-xyz	0.0482	0.0397	0.0098	-	**0.0088**
fr3-sitting-half	0.0470	0.0432	0.0217	0.0196	**0.0145**

Table 3
aRMSE of translational drift (RPE) for TUM RGB-D dataset in dynamic environments [m/s].

Sequences	Sun [33]	Li [19]	Kim [16]	Wang [36]	Ours
fr3-walking-static	0.0842	0.0327	0.1339	0.1881	**0.0150**
fr3-walking-xyz	0.1214	0.0651	0.2326	0.2158	**0.0241**
fr3-walking-half	0.1672	**0.0527**	0.1738	0.1908	0.1369
fr3-sitting-static	-	0.0231	0.0248	**0.0077**	0.0115
fr3-sitting-xyz	0.0330	0.0219	0.0482	**0.0117**	0.0131
fr3-sitting-half	0.0458	0.0389	0.0589	0.0245	**0.0189**

Table 4
aRMSE of rotational drift (RPE) for TUM RGB-D dataset in dynamic environments[°/s].

Sequences	Sun [33]	Li [19]	Kim [16]	Wang [36]	Ours
fr3-walking-static	2.0487	0.8085	2.0833	3.2101	**0.3269**
fr3-walking-xyz	3.2346	1.6442	4.3911	3.6476	**0.6481**
fr3-walking-half	5.0108	2.4048	4.2863	3.3321	**1.5543**
fr3-sitting-static	-	0.7228	0.6997	**0.2595**	0.3514
fr3-sitting-xyz	0.9828	0.8466	1.3885	**0.4997**	0.5639
fr3-sitting-half	2.3748	1.8836	2.8804	0.5643	**0.5577**

shows the semantic maps. The left column presents a static case; whereas the right column presents a dynamic case and a movable object, i.e., the person, is removed after semantic segmentation and object detection. Three-dimensional boxes are plotted over the 3D models with only stationary objects. In the semantic maps, the detected objects are marked by different boxes, which enables scene understanding and navigation for medical assistant robots.

We carried out a long-distance mapping of an entire floor in a local hospital. Fig. 7 shows the trajectory of our robot and zoning with example images including clean zone, buffer zone, and isolation area. It shows a part of an infectious medical facility for Covid-19.

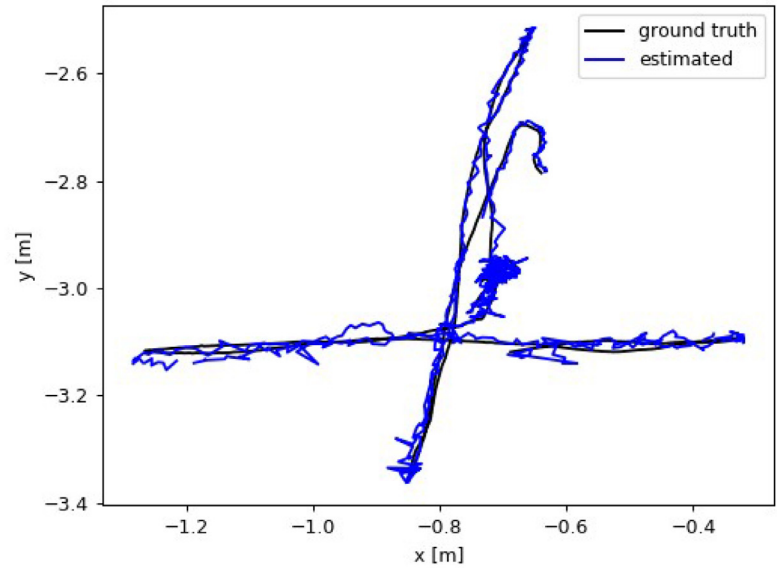
4.2. Pose estimation accuracy

The TUM dataset [32] collected image sequences of 39 different indoor scenes. Each sequence contains 640×480 8-bit RGB images with timestamps and 640×480 16-bit depth images as well as accurate real camera trajectories.

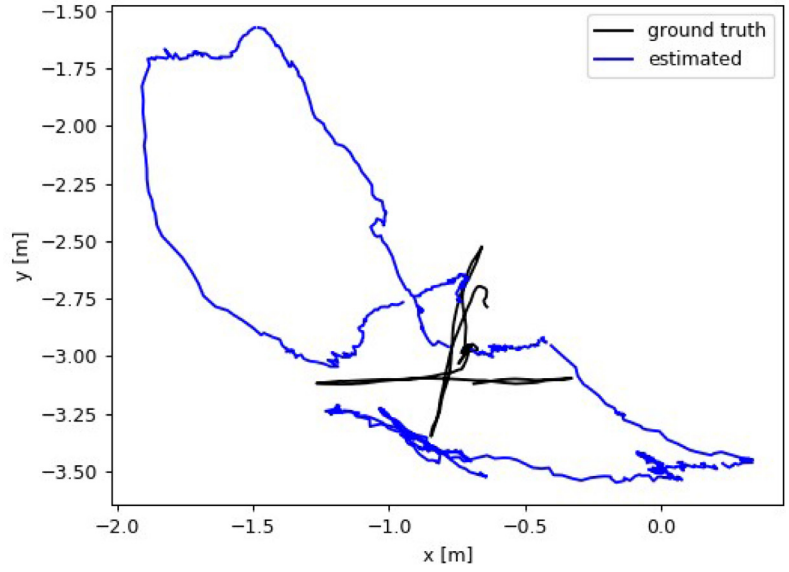
These scenes can be divided into three categories: static scenes, low dynamic scenes, and high dynamic scenes. In a static scenario, since no P_{moving} object can be detected, the tracking thread is consistent with ORB-SLAM2. A low dynamic scene refers to a scene with a small motion range of objects. The fr3-sitting-static sequence in the TUM dataset is just such a scene, in which two people are sitting and communicating. There are objects with a large range of motion in high dynamic scenes. Our experiments include three sequences of high dynamic scenes, fr3-walking-static, fr3-walking-xyz, and fr3-walking-half-sphere.

For the Absolute Trajectory Error (ATE) indicator, this paper selects six sequences from the data set for testing: fr3-walking-static, fr3-walking-xyz, fr3-walking-half-sphere, fr3-sitting-static, sitting-xyz, sitting-half-sphere. The size of the semantic descriptor is set to 21×21 and the threshold δ is set to 55. We conducted extensive experiments by comparing the test results with ORB-SLAM2 and other dynamic SLAM systems to evaluate the performance of our algorithm. Because the system has a certain degree of uncertainty, we run ten times on each experimental sequence and the median value is taken to obtain objective and accurate results.

The comparison results of this algorithm against ORB-SLAM2 are shown in Table 1, including average root mean square error (aRMSE), mean, standard deviation (STD), and median errors. The aRMSE is computed by computing the average RMSE of the estimated trajectory against the ground truth trajectory of ten repetitions. The mean and medium are the mean and medium of the absolution difference w.r.t. the ground truth in all repetitions. A



(a) our method



(b) ORB-SLAM2

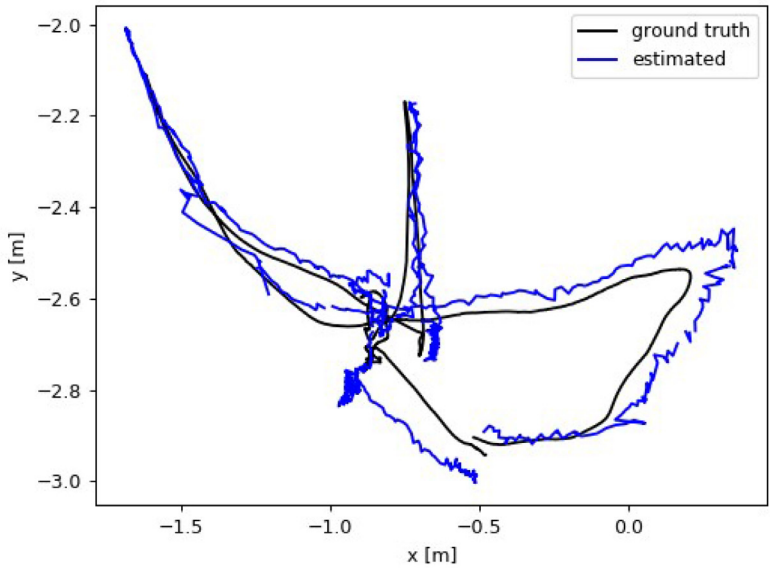
Fig. 8. Robot trajectory from test case fr3-walking-xyz.

Table 5
RMSE of the trajectories with different values of parameters.

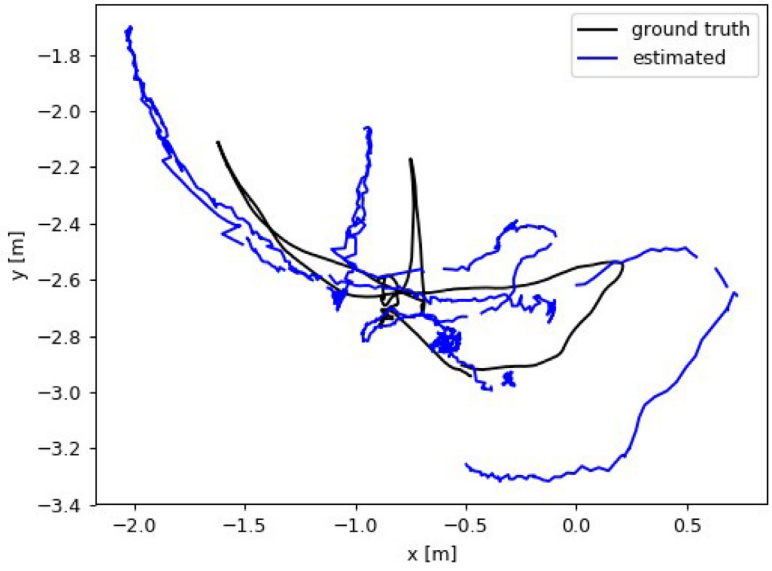
Sequences	$n = 5, \delta = 3$	$n = 55, \delta = 378$	$n = 21, \delta = 55$
fr3-walking-static	0.0198	0.0163	**0.0104**
fr3-walking-xyz	0.0382	0.0285	**0.0164**
fr3-walking-half	0.3020	0.2062	**0.0923**
fr3-sitting-static	0.0075	0.0074	**0.0065**
fr3-sitting-xyz	**0.0086**	0.0093	0.0088
fr3-sitting-half	**0.0140**	0.0181	0.0145

small aRMSE, mean, and median indicate high accuracy of the estimated trajectory. For sequence fr3-walking-half-sphere, the improvement by our method reaches more than 80%. For sequence fr3-walking-static and fr3-walking-xyz, the improvement by our method reaches more than 95%. And for sequence fr3-sitting-static, fr3-sitting-xyz and fr3-sitting-half-sphere, there are also improvements. Compared with ORB-SLAM2, it is evident that the proposed algorithm gains great improvement in high dynamic scenarios. In cases with fewer or no dynamic objects, e.g., sequence fr3-sitting-static, the results of our method are as good as ORB-SLAM2 with slight improvements. The advantage of our method is much significant in dynamic scenarios. By comparing the STD, the results of our method are much less than that of the ORB-SLAM2 in most cases. This demonstrates that the proposed method exhibits greater robustness.

We conducted experiments comparing with other dynamic SLAM systems [19,33,36]. Table 2 presents the RMSE of ATE of the compared methods. In Sun [33], the number of clusters is set to



(a) our method

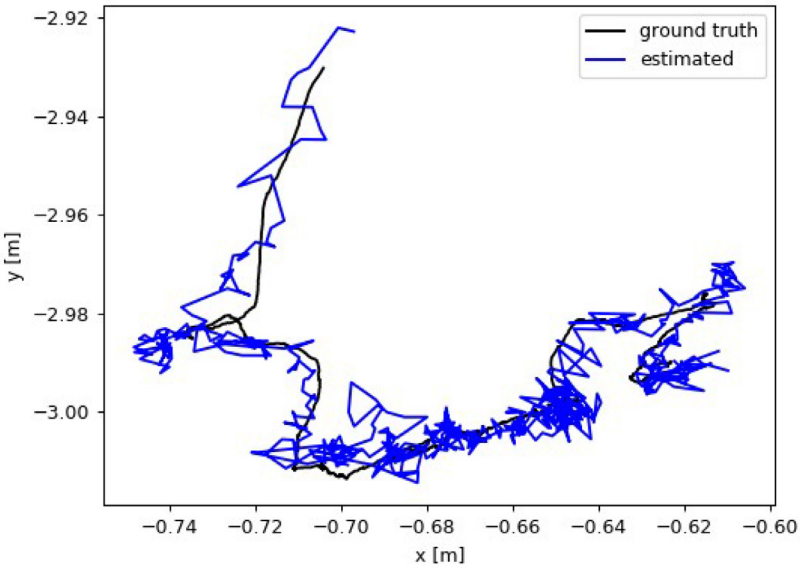


(b) ORB-SLAM2

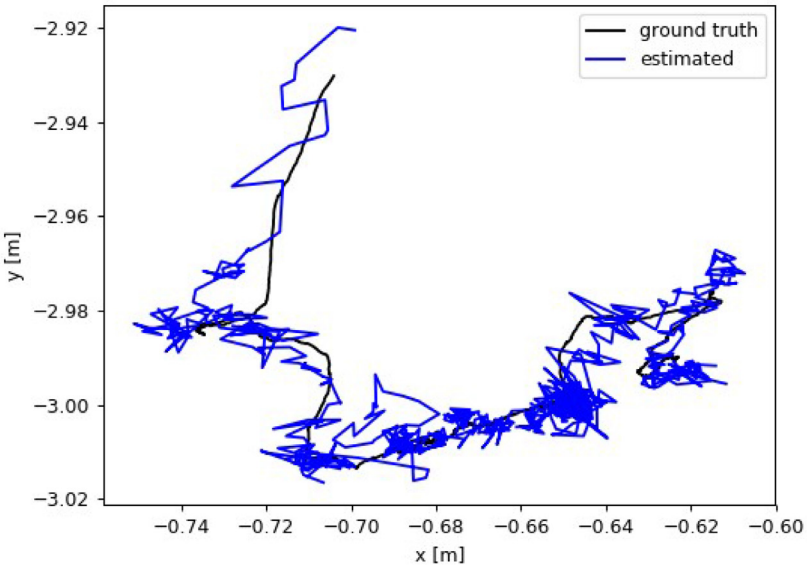
Fig. 9. Robot trajectory from test case fr3-walking-halfsphere.

Table 6
Absolute trajectory error of our algorithm with and without distance calculation.

	Ours without distance calculation				Ours with distance calculation			
Sequence	RMSE	Mean	Median	S.D.	RMSE	Mean	Median	S.D.
fr3-walking-static	0.0104	0.0090	0.0079	0.0052	0.0095	0.0083	0.0074	0.0045
fr3-walking-xyz	0.0164	0.0139	0.0116	0.0087	0.0158	0.0131	0.0107	0.0079
fr3-walking-half	0.0923	0.0857	0.0838	0.0344	0.0916	0.0850	0.0827	0.0332
fr3-sitting-static	0.0065	0.0060	0.0054	0.0033	0.0058	0.0051	0.0049	0.0030
fr3-sitting-xyz	0.0088	0.0079	0.0070	0.0043	0.0077	0.0072	0.0062	0.0045
fr3-sitting-half	0.0145	0.0125	0.0115	0.0074	0.0154	0.0133	0.0121	0.0078



(a) our method



(b) ORB-SLAM2

Fig. 10. Robot trajectory from test case fr3-sitting-static.

5 and the reprojection error threshold is set to 3 and the number of particles is set to 1000. In Li [19], the coefficient of the depth-dependent threshold and the depth discontinuity threshold are set to 0.015m and 0.04m, respectively; the degree of freedom of t-distribution is set to 10 and the mean value is set to 0; the coefficient of the static weight is set to 1 if the keyframe is the current frame, otherwise set to $0.5N/(N+v-k)$ where N represents the number of frames between latest and last keyframe while v and k represent the index of the current frame and latest keyframe respectively; the threshold of geometric proximity distance is set to 1.5m; the distance difference threshold and angle difference threshold for consistency check are set to 0.02m and 3°, respectively. In [36], the number of clusters is set to 10. As can be seen from the table, we get the best result on four out of six sequences. On the sequence fr3-walking-static, our result is the second best.

Tables 3 and 4 show the translation aRMSE and rotation aRMSE of relative pose error (RPE) respectively. In Table 3, we perform the best on three sequences and the second-best on the other three sequences. In Table 4, we gain the highest accuracy on four of the sequences and the second-highest on the other two sequences, which demonstrates that our algorithm achieved competitive performance.

We analyzed the value selection of the parameters. Results of different values of the parameters are shown in Table 5. n represents the side length of the semantic descriptor and δ is the threshold mentioned before. We find that if the size is set smaller than 10×10, the result will go worse because it is difficult for

Table 7
Average time expense (in milliseconds).

Case	ORB-SLAM2	Berta [4]	Yu [40]	Ours
Dynamic	-	235.98	48.31	**17.21**
Stationary	**20.10**	3,362.22	55.19	1,021.37

the under-sized semantic descriptor to capture surrounding information. Also, if the size is bigger than 49×49, the result will go worse too. This is because moving objects (e.g., humans) are captured too early by semantic descriptors when they are still far from movable objects such as books. That leads to the elimination of static feature points that affect feature points matching. The threshold δ should be kept relatively small. When δ is large, moving objects could be missed. On the other hand, δ should not be too small. This is because if the number of points in the semantic descriptor that belong to moving objects is too few, it could be the situation that the moving object is in a far distance or the result of the semantic segmentation network is inaccurate. Our experiments show that keeping δ the range of $1/8$ to $1/2$ of the semantic descriptor size yielded satisfactory results.

Figs. 8–10 illustrate the trajectories estimated by ORB-SLAM2 and our proposed method. The black trajectories represent the ground truth while the blue curves represent the estimated trajectories. In each Figure, (a) shows a comparison of the result of our method and the ground truth and (b) shows a comparison of the result of ORB-SLAM2 and the ground truth. It can be seen that the estimated trajectory in (a) is closer to the ground truth than that in (b). That is, our method is more accurate than ORB-SLAM2 in dynamic scenarios. This improvement is attributed to the elimination of dynamic feature points.

We also conducted experiments to show that the accuracy of feature point matching can be further improved by calculating the distance between two semantic descriptors of different frames. We calculate the semantic distance following Eq. (6). Based on the algorithm proposed, for each matched pair of feature points if the semantic distance is far than a threshold (set to 110 in our experiment), they are considered as an outlier that is excluded from participating in the following steps. Table 6 shows the result. Since fewer pairs of feature points are mismatched, our algorithm with distance calculation is slightly better.

4.3. Efficiency analysis

We evaluate the time complexity of the dynamic object detection and removal algorithm proposed in this paper. The programs are written with C++ language and tested on a computer with Intel i7-8750H CPU, GTX1060 6G GPU, 16GB memory, and Ubuntu16.04 OS. The videos used in our experiments include cases with and without dynamic objects. Videos consist of RGB and depth components. The frame size is 640 by 480. The number of frames varies in the range of 600 to 1500.

We compare our proposed method with the state-of-the-art methods [4,40]. The experiments were repeated ten times and the average time is reported in Table 7. The visual SLAM system consists of several components. To study the time used for dynamic object detection and removal, the results reported excludes the time consumed by the inference of the neural network, which can be trained offline with a much powerful computer. In the evaluation of the stationary cases, the time reported includes the training of the network for our method to be consistent with ORB-SLAM2 that involves no training process.

The average time spent by our method is 17.21 milliseconds for detecting dynamic objects and removal, which is less than half of the time spent by the second-best method [40]. It is evident that the proposed method is more efficient in handling dynamic objects. In our evaluation for the stationary cases, ORB-SLAM2 spent the least amount of time. Our proposed method took 1,021.37 milliseconds, which involves the training of the network.

5. Conclusion

This paper presents a novel visual SLAM algorithm, applied in a hospital scene serving the treatment of COVID-19. It is implemented on the ORB-SLAM2 framework, handling complex dynamic environments. This algorithm uses the semantic segmentation results of MASK R-CNN to construct a semantic descriptor and creatively combines the knowledge graph to obtain high-level semantic information. We effectively remove moving objects, and, hence, reduce the impact of feature points mismatching and finally improve the accuracy of pose estimation. We conducted experiments in real hospital scenes and successfully established semantic maps. To verify the effectiveness of the algorithm, we conducted extensive experiments on the TUM dataset comparing with other algorithms, and the results show great improvement in highly dynamic scenarios. This algorithm not only can be applied to RGB-D cameras but also can be extended to monocular and stereo cameras, having good application prospects.

In the future, we plan to improve the real-time performance of the system. We will investigate other semantic segmentation models considering both accuracy and efficiency. We also find that the results of MASK R-CNN are not that accurate around the contours of objects. This can be further studied to help refine the semantic information around the contours. Also, it will be beneficial to take into account the pixel-wise accuracy of semantic segmentation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the special fund for Basic Scientific Research in Central Colleges and Universities (Grant No. ACAIM190102), the Project of Collaborative Innovation in Anhui Colleges and Universities (Grant No. GXXT-2019-003), the Open Fund of Key Laboratory of Flight Techniques and Flight Safety (Grant No.2018KF06), Scientific Research Project of Civil Aviation Flight University of China (Grant No.J2020-125) and Open Fund of Key Laboratory of Flight Techniques and Flight Safety, CAAC (Grant No. FZ2020KF02). We would like to express our gratitude to all those who helped us during the writing of this paper. We acknowledge the help of Professor H. Wang, who has offered suggestions in academic studies. We awe a special debt of gratitude to the Hospital of the Hefei University of Technology, from where we can get the data of hospital environment data. We thank doctor Gao, who is vice director of the Hospital of the Hefei University of Technology, and chief nurse, Ms. Zhang for their reasonable and wise advice in the healthcare field. Finally, thanks to all the partners of the lab for their company and support.

References

[1] I. Alonso, L. Riazuelo, A. Murillo, Enhancing V-SLAM keyframe selection with an efficient ConvNet for semantic analysis, in: Proceedings of IEEE International Conference on Robotics and Automation, Piscataway, NJ, 2019, pp. 4717–4723.
[2] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: a deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach.Intell. 39 (12) (2017) 2481–2495.

[3] H. Bay, T. Tuytelaars, L.J.V. Gool, SURF: speeded up robust features, in: Proceedings of European Conference on Computer Vision, Springer-Verlag, Berlin Heidelberg, 2006, pp. 404–417.
[4] B. Berta, J.M. Facil, C. Javier, J. Neira, DynaSLAM: tracking, mapping and inpainting in dynamic scenes, IEEE Rob. Autom. Lett. 3 (4) (2018) 4076–4083.
[5] M. Calonder, V. Lepetit, P. Fua, BRIEF: binary robust independent elementary features, in: Proceedings of European Conference on Computer Vision, Springer-Verlag, Berlin Heidelberg, 2010, pp. 778–792.
[6] L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, IEEE Trans. Pattern Anal. Mach.Intell. 40 (4) (2018) 834–848.
[7] G.L. David, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[8] Y. Ding, X. Zhang, J. Tang, A noisy sparse convolution neural network based on stacked auto-encoders, in: Proceeding of IEEE International Conference on System, Man, Cybernetics, 2017, pp. 3457–3461.
[9] J. Engel, T. Schps, D. Cremers, LSD-SLAM: large-scale direct monocular SLAM, in: Proceedings of European Conference on Computer Vision, Springer, Cham, 2014, pp. 834–849.
[10] B. Fang, J. Ding, Z. Wang, Autonomous robotic exploration based on frontier point optimization and multistep path planning, IEEE Access 7 (2019) 46104–46113.
[11] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.
[12] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, in: Proceedings of IEEE International Conference on Computer Vision, Piscataway, NJ, 2017, pp. 2980–2988.
[13] Z. Hu, J. Tang, P. Zhang, J. Jiang, Deep learning for the identification of bruised apples by fusing 3D deep features for apple grading systems, Mech. Syst. Signal Process. 145 (2020) 106922.
[14] Z. Hu, J. Tang, Z. Wang, K. Zhang, L. Zhang, Q. Sun, Deep learning for image-based cancer detection and diagnosis – a survey, Pattern Recognit. 83 (2018) 134–149.
[15] M. Kaneko, K. Iwami, T. Ogawa, T. Yamasaki, K. Aizawa, Mask-SLAM: robust feature-based monocular slam by masking using semantic segmentation, in: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Piscataway, NJ, 2018, pp. 3710–3718.
[16] D.H. Kim, J.H. Kim, Effective background model-based RGB-d dense visual odometry in a dynamic environment, IEEE Trans. Rob. 32 (6) (2017) 1565–1573.
[17] J. Klappstein, T. Vaudrey, C. Rabe, A. Wedel, R. Klette, Moving object segmentation using optical flow and depth information, in: Proceedings of Pacific-Rim Symposium on Image and Video Technology, Springer-Verlag, Berlin Heidelberg, 2009, pp. 611–623.
[18] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, in: Proceedings of 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Piscataway, NJ, 2007, pp. 225–234.
[19] S. Li, D. Lee, RGB-D SLAM in dynamic environments using static point weighting, IEEE Rob. Autom. Lett. 2 (4) (2017) 2263–2270.
[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, C.L. Zitnick, Microsoft COCO: common objects in context, in: Proceedings of European Conference on Computer Vision, Springer, Switzerland, 2014, pp. 740–755.
[21] Matterport, Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow, 2018/2020. https://github.com/matterport/Mask_RCNN.
[22] J. Mccormac, A. Handa, A. Davison, S. Leutenegger, SemanticFusion: dense 3D semantic mapping with convolutional neural networks, in: Proceedings of IEEE International Conference on Robotics and Automation, Piscataway, NJ, 2017, pp. 4628–4635.
[23] R. Munoz-Salinas, R. Medina-Carnicer, UcoSLAM: simultaneous localization and mapping by fusion of keypoints and squared planar markers, Pattern Recognit. 101 (2020) 107193.
[24] R. Murartal, J.D. Tardos, ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras, IEEE Trans. Rob. 33 (5) (2017) 1255–1262.
[25] R. Munoz-Salinas, M.J. Marin-Jimenez, R. Medina-Carnicer, SPM-SLAM: simultaneous localization and mapping with squared planar markers, Pattern Recognit. 86 (2019) 156–171.
[26] L.R. Paul, Measuring corner properties, Comput. Vis. Image Underst. 73 (3) (1999) 291–307.
[27] N.D. Reddy, P. Singhal, V. Chari, K.M. Krishna, Dynamic body VSLAM with semantic constraints, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Piscataway, NJ, 2015, pp. 1897–1904.
[28] L. Riazuelo, L. Montano, J.M.M. Montiel, Semantic visual SLAM in populated environments, in: Proceedings of European Conference on Mobile Robots, Piscataway, NJ, 2017, pp. 1–7.
[29] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: an efficient alternative to SIFT or SURF, in: Proc. of the International Conference on Computer Vision, IEEE, 2011, pp. 2564–2571.
[30] E. Rosten, Machine learning for high-speed corner detection, in: Proceedings of European Conference on Computer Vision, Springer-Verlag, Berlin Heidelberg, 2006, pp. 430–443.
[31] R.F. Salasmoreno, R.A. Newcombe, H. Strasdat, P.H.J. Kelly, A.J. Davison, SLAM++: simultaneous localisation and mapping at the level of objects, in: Proceedings of IEEE Computer Vision and Pattern Recognition, Piscataway, NJ, 2013, pp. 1352–1359.
[32] J. Sturm, N. Engelhard, F. Endres, W. Burgard, D. Cremers, A benchmark for the evaluation of RGB-D SLAM systems, in: Proc. of the International Conference on Intelligent Robot Systems, 2012, pp. 573–580.
[33] Y. Sun, M. Liu, Q.H. Meng, Improving RGB-D SLAM in dynamic environments: a motion removal approach, Rob. Auton. Syst. 89 (2017) 110–122.
[34] W. Tan, H. Liu, Z. Dong, G. Zhang, H. Bao, Robust monocular SLAM in dynamic environments, in: Proceedings of IEEE International Symposium on Mixed and Augmented Reality, Piscataway, NJ, 2013, pp. 209–218.
[35] K. Wang, Y. Lin, L. Wang, L. Han, M. Hua, X. Wang, S. Lian, B. Huang, A unified framework for mutual improvement of SLAM and semantic segmentation, in: Proceedings of IEEE International Conference on Robotics and Automation, Piscataway, NJ, 2019, pp. 5224–5230.
[36] R. Wang, W. Wan, Y. Wang, K. Di, A new RGB-D SLAM method with moving object detection for dynamic indoor scenes, Remote Sens. 11 (10) (2019) 1143.
[37] T. Whelan, R.F. Salas-moreno, B. Glocker, A.J. Davison, S. Leutenegger, ElasticFusion, Int. J. Rob. Res. 35 (14) (2016) 1697–1716.
[38] G. Yang, H. Lv, Z. Zhang, L. Yang, J. Deng, S. You, J. Du, H. Yang, Keep healthcare workers safe: application of teleoperated robot in isolation ward for COVID-19 prevention and control, Chin. J. Mech. Eng. 33 (1) (2020).
[39] R. Ye, X. Zhou, F. Shao, L. Xiong, J. Hong, H. Huang, W. Tong, J. Wang, S. Chen, A. Cui, C. Peng, Y. Zhao, L. Chen, Feasibility of a 5G-based robot-assisted remote ultrasound system for cardiopulmonary assessment of patients with COVID-19, Chest 159 (1) (2021) 270–281.
[40] C. Yu, Z. Liu, X.J. Liu, F. Xie, Y. Yang, Q. Wei, Q. Fei, DS-SLAM: a semantic visual SLAM towards dynamic environments, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Piscataway, NJ, 2018, pp. 1168–1174.
[41] X. Yuan, L. Kong, D. Feng, Z. Wei, Automatic feature point detection and tracking of human action in time-of-flight videos, IEEE/CAA J. Autom. Sin. 4 (4) (2017) 677–685.
[42] C. Zhao, L. Sun, P. Purkait, T. Duckett, R. Stolkin, Dense RGB-D semantic mapping with Pixel-Voxel neural network, Sensors 18 (9) (2018) 3099.
[43] C. Zhuang, X. Yuan, W. Wang, Boundary enhanced network for improved semantic segmentation, in: International Conference on Urban Intelligence and Applications, 2020, pp. 172–184. Taiyuan, China, Aug. 14-16

Dr. Xiaohui Yuan received a B.S. degree in electrical engineering from the Hefei University of Technology, Hefei, China in 1996 and a Ph.D. degree in computer science from Tulane University, New Orleans, LA, USA in 2004. He is currently an Associate Professor at the Department of Computer Science and Engineering in the University of North Texas. His research interests include computer vision, data mining, machine learning, and artificial intelligence. His research findings have been reported in over one hundred peer-reviewed papers. Dr. Yuan is a recipient of Ralph E. Powe Junior Faculty Enhancement award in 2008 and the Air Force Summer Faculty Fellowship in 2011, 2012, and 2013.