A Robust Localization Method in Indoor Dynamic Environment

Shan Huang¹, Hong-Zhong HUANG¹ and Qi Zeng²
¹Center for System Reliability and Security Research, University of Electronic Science and Technology of China, chengdu 611731, china
²EvenTec, chengdu 610097, china

Abstract: Localization is one of the core technologies for mobile robots to achieve full autonomous movement, and is a prerequisite for other autonomous tasks. The robot working environment is dynamic in most cases, so the localization algorithm must overcome the effects of dynamic changes in the environment. The paper proposed a localization algorithm that allows the robot to perform robust and lifelong localization in dynamic environment. The algorithm filters out high-dynamic objects and updates semi-static objects on the map at the same time, it can also use the information provided in semi-static objects to improve localization performance. In this paper, the processing of dynamic objects is divided into two parts: filtering of high-dynamic objects and updating of semi-static objects. For high dynamic object filtering, a dynamic object detection method combining a delay comparison method and a tracking method is proposed by observed the characteristics of localization system; For the update of semi-static objects, this paper uses the pose graph optimization and occupancy map to implement the dynamic update of the map. The combination of the two methods allows the robot to achieve long-term stable localization in a dynamic environment. The experimental results demonstrate that the proposed method allows the robot achieve long-term localization, overcome the effects of high-dynamic objects and keeping the map always consistent with the environment.

Keywords: robotics; localization algorithm; dynamic environment; map update; localization reliability

1 Introduction

With the development of mobile robot technology, fully autonomous mobile robots have been widely used in scenarios such as warehousing logistics, field rescue, park delivery and mall services. Accurate positioning of mobile robots is a prerequisite for mobile robots to perform fully autonomous movements. Localization technology is regarded as one of the core
technologies of mobile robot technology. Therefore, the study of mobile robots localization has great significance for realize fully autonomous movement of mobile robots.

There are a large number of researches on mobile robot localization technology. In general, mobile robot localization methods can be divided into two categories: wireless-based and map-based[1].

Wireless-based methods include GPS, UWB methods, Bluetooth methods, wifi methods, and RFID methods[2]. The advantage of these positioning methods is that if there is a signal, the absolute position can be obtained without accumulated errors. However, the wireless-based method is susceptible to occlusion, and the base station needs to be set in advance. This severely limits the application of wireless-based method in the field of mobile robots.

The map-based method needs to build an environment map, and then use the map to match with the sensor data to get the robot’s position. Because the map-based method does not need to modify the environment, it has been widely used in the field of mobile robots.

According to the sensor type, the robot localization method can be divided into Lidar-based method and camera-based method[3]. Because lidar has the advantages of high accuracy and strong anti-interference ability, this paper uses lidar for robot localization. The localization method mentioned below in this paper refer to lidar-based method.

F. Dellaert[4] proposed Monte Carlo localization (MCL) method. This method uses particles to represent robot's pose, and uses the odometer to propagate the particles, and uses the lidar observation data to weight particles. The position of the robot is obtained by the weighted sum of all particle. But MCL assumes that the environment is static and performs poorly in a dynamic environment. D. Fox[5] proposed a method that identifying whether the lidar beam come from dynamic objects by comparing the actual range and the expected range of the lidar beam. D. Schulz et al[6] through tracking dynamic object to distinguish whether the lidar data comes from dynamic objects. Although the above method can handle highly dynamic objects better, it cannot handle semi-static objects. However, there are a large number of semi-static objects in the actual environment, such as the change of the store location in the mall, the decoration of the store, the movement of the furniture position, etc. Therefore, in addition to the processing of highly dynamic objects, the processing of semi-static objects is also very important for the long-term localization.

For the processing of semi-static objects, the basic idea is to update the static objects to the map. Chen et al[7] and Brechtel et al[8] extended the original grid map to include dynamic objects in the grid. Meyer-Delius et al[9] proposed dynamic grid map, which uses a hidden Markov model to represent the probability of dynamic objects. Tipaldi et al[10] use RBPF (Rao-Blackwellized particle filter) to simultaneously estimate the pose of the robot and the state of the environment. The above methods all use RBPF to estimate the robot pose and update the map. Therefore, they cannot meet the real-time requirements. or it consume a huge amount of memory, and it not works under limited resource conditions.

Instead of use particle filter, Maria et al[11] and Boniardi et al[12] using pose graph to process dynamic objects. This paper is also inspired by their work. The main difference is that this paper use occupancy map while they use point cloud map. The point cloud map are very sensitive to the sensor noise and are easily affected by dynamic objects. And occupancy map are naturally insensitive to these noises, so it is more robust than point cloud map.

Since the operating environments of service robots are full of high dynamic objects and
semi-static objects, the long-term stable application of robots requires a localization system that can handle high dynamic objects and semi-static objects at the same time. At the same time, due to cost reasons, the localization system of service robot cannot consume too much computing resources and memory resources. In order to overcome the shortcomings of the above methods and achieve long-term stable operation of the robot, this paper proposes a positioning method that can simultaneously handle highly dynamic objects and semi-static objects. At the same time, this method can run on the Pentium processor in real time. This method identifies highly dynamic object by tracking and detecting high dynamic objects, and removes these data as noise, which greatly improves the localization stability in a highly dynamic environment. At the same time, because semi-static objects can provide rich information for localization, this paper integrating semi-static object information into the map by map update, and proposed a new map update method which can update maps very quickly. The proposed method has been tested in the mall for one year. The test results show that this method can process highly dynamic objects and semi-static objects at the same time, and can achieve long-term stable localization in this environment.

We summarize our contributions as follows:

(1) A practical robot localization system, which can handle highly dynamic objects and semi-static objects, and achieve stable localization in a dynamic environment.

(2) A new dynamic object detection method, which uses the characteristics of the localization system to achieve stable detection of dynamic obstacles by delaying decision-making.

(3) A map updating method, which fusion the pose-graph optimization and the grid map to realize the dynamic update of the map.

(4) Integration the proposed method with planning module, mapping module and control module in a mobile robot platform, which has been tested for one year in the mall environment.

The rest of this paper is presented as follow. In Section 2, the system overview is introduced. The details of the implementation of the proposed method are presented in Section 3 and Section 4. Section 5 provides the experimental results. Finally the conclusion and future works are summarize in Section 6.

2 System Overview
The overall system framework is shown in Figure -1

![Figure 1. system framework](image)

As can be seen from Figure -1, the localization algorithm is divided into two modules: highly
dynamic object processing module and map updating module (semi-static object processing module). The sensors used include lidar and wheel odometer. The highly dynamic object processing module receives lidar data and wheel odometer data, and uses the odometer data as the ground truth to detect and track dynamic objects. This module detects the laser data caused by high dynamic objects, and outputs a filtered laser data without laser beams caused by high dynamic object; The map update module receives filtered laser scan and odometer data, outputs updated map and robot position.

3 High Dynamic Object Process Module
The purpose of high dynamic object process module is to filter out laser beams caused by moving objects, such as pedestrians, carts, etc. Because these moving objects can not only provide localization information, but also reduce the signal-to-noise ratio of observation data and reduce the success rate of localization. This paper combines the comparison method and the tracking method to process dynamic objects. Proposed a delay comparison method that overcome the shortcomings of the comparison method by considering the characteristics of the localization system. The delay comparison method greatly improve the performance of dynamic object detection.

3.1 Delayed Comparison Method
The comparison method refers to the comparison between the current laser scan z(t) and the previous laser scan z(t-1) or the last few laser scans z(t-k), if the data in the current laser scan falls in the field of view of the last laser scans, the data was caused by dynamic object. If it falls within the field of view of the last laser scan, it means that the object was not observed at the last scan, but the object was observed at the current scan, so the object must be a dynamic object. The comparison method is shown in Figure 2:

![Figure 2. comparison method](image)

The cross symbol indicates the current position of the robot, the yellow dot indicates the previous laser scan, the magenta polygon indicates the field of view corresponding to the previous laser scan, the blue dot indicates the current laser scan, and the red dot indicates that the current laser point that detected as dynamic objects, which is a moving people. The red dot is within the field of view of the previous laser scan, so it is considered as dynamic object.
For robots, there are two types of dynamic objects: approaching the robot and moving away from the robot. It can be seen from the principle of the comparison method that it can only handle dynamic objects approaching the robot, because if the dynamic object is moving away from the robot, the dynamic object is not in the field of view of the previous laser scan, so the comparison method cannot detect the dynamic object moving away from the robot.

In order to overcome the problem of the comparison method, this paper proposes the delay comparison method. The odometer data can be considered high precision in short time, so the localization system only needs to correct the drift of the odometer on a regular period, and does not need to be corrected in real time. That is to say, it is not necessary to use the latest laser data z(t) for localization, but the last laser data z(t-1) can be used, such as the laser data a few seconds ago. If the past data z(t-1) is used for localization, the data z(t-1) can not only be compared with the previous data z(t-2) to identify dynamic objects, but also can be compared with the future data z(t) to identify dynamic objects. The above is the delayed comparison method. The delayed comparison method is shown in Figure 3:

![Figure 3. delayed comparison method](image)

The magenta polygon indicates the field of view of z(t-1), the green polygon indicates the field of view of z(t+1), and the red dot indicates a person who is moving away from the robot. Obviously the red point is outside the magenta polygon, so the comparison method cannot recognize that it is a dynamic object, However, the red dot is inside the green polygon, so the delayed comparison method can identify the pedestrian as a dynamic object.

Obviously, comparing with past data can identify dynamic objects approaching the robot, while comparing with future data can identify objects moving away from the robot, so the delayed comparison method can identify two types of dynamic objects. Compared with the comparison method, it can greatly improve the robot's ability of handling of dynamic objects and enhance the stability of the robot's localization in a highly dynamic environment.

### 3.2 Tracking Method

Unlike the comparison method, the tracking method[11] does not use the field of view of the previous laser scan, but only uses the laser points. The tracking method first clusters laser points a according to Euclidean distance, so as to decompose laser points into different classes. These different classes are matched to the classes of the previous laser points determine whether it is the same object. If it is determined that it is the same object, it can be considered that the tracking is successful; if there is no match, it can be considered as a newly appearing object, and a new ID is assigned for tracking. To determine whether the two classes are the same object, it can be use the distance and similarity of the two classes as metric. Every time an new object
appears, a Kalman filter is assigned to calculate the position and speed of the object. If the 
object is tracked continuously for several frames, it is judged whether it is a dynamic object 
based on the object's speed. If it’s speed is larger than a threshold, it is considered a dynamic 
object.

4 Semi-Static Object Process Module

Unlike high dynamic objects, semi-static objects can provide rich localization information\textsuperscript{[19]}. Therefore, it cannot be directly regarded as noise, but it is necessary to make full use of the 
information provided by semi-static objects to achieve long-term stable localization. When 
processing semi-static objects, it is assumed that the high dynamic objects in the laser data have 
been filtered out, and the laser data only contains static objects (objects matching the map) and 
semi-static objects (objects not matching the map). The paper uses the pose graph 
optimization\textsuperscript{[12]} to update the map.

4.1 Pose-Graph Optimization

Pose-Graph is formulation of robot state estimation problem, which uses nodes to represent the 
pose of the robot and edges to represent the spatial constraints between the two nodes. Pose-
Graph is shown in Figure 4:

![Figure 4. Pose-Graph](image)

$X_i$ is the robot pose, $X_i = (x_i, y_i, \theta_i)$. $Z_{ij}$ is the spatial constraints between nodes. $Z_{ij}$ can 
be obtained by odometer or laser scan-matching. The black edge represents the inter-frame 
constraint, and the red edge represents the loop-closure constraint. We use the CSM\textsuperscript{[13]} 
algorithm for coarse search and use the NICP\textsuperscript{[14]} algorithm for fine search to calculate the 
relative pose of two nodes. Given the inter-frame constraints and loop-closure constraints, the 
maximum likelihood estimation of the robot trajectory can be obtained through back-end 
optimization. If the pose of the entire trajectory of the robot is known, the corresponding grid 
map can be generated by the occupied grid mapping algorithm\textsuperscript{[15]}.

Both the odometer measurement and the inter-frame scan-matching have errors, so as the 
robot walks, the errors will be accumulated. Only when the robot enters the known area and 
forms a loop detection can the error be eliminated. The red edge in Figure 4 represents the loop-
closure constraint, $X_4$ and $X_5$ measurement the same feature in the environment. Therefore, we
can use the scan-matching algorithm to calculate the relative pose \( Z_{ij} \), and build the loop-closure constraint. After the construction of the loop constraint is completed, the maximum likelihood estimation of the robot trajectory can be solved by optimization.

The observations of relative poses between nodes are denoted by \( Z_{ij} \) and \( \Lambda_{ij} \). \( Z_{ij} \) means the relative pose between node i and node j, \( \Lambda_{ij} \) is the corresponding information matrix. The predicted value of the relative pose between the nodes is the current relative pose between the two nodes, the predicted value is given by:

\[
\begin{align*}
h(X_i, X_j) &= \left[ R^T(\theta) \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \right] \\
&= \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix}
\end{align*}
\]

Obviously the error function is the difference between the predicted value and the observed value:

\[
e_{ij}(X) = e(X_i, X_j) = h(X_i, X_j) - Z_{ij}
\]

Solving the maximum likelihood estimation of the pose of the robot is equivalent to solving the objective function:

\[
X^* = \arg\min_X \sum e_{ij}^T(X)\Lambda_{ij}e_{ij}(X)
\]

Since the prediction function is a nonlinear function, the problem of robot pose estimation is a nonlinear least square problem. For the solution of equation (3), it can be performed by local linearization and then iterative update. Because the non-linearity comes from the error function, we can take the first-order approximation to achieve the linearization of the entire objective function:

\[
e_{ij}(X + \Delta X) \approx e_{ij}(X) + J_{ij}\Delta X
\]

\( J_{ij} \) is the Jacobian matrix of the error function relative to \( X \). According to equation (1) and equation (2), \( e_{ij}(X) \) is only related to \( X_i \) and \( X_j \), and is not related other terms. There \( J_{ij} \) has only two terms, \( X_i \) and \( X_j \), is non-zero, all other terms is zero:

\[
J_{ij} = \begin{bmatrix} 0 & \cdots & \frac{\partial e_{ij}(X)}{\partial X_i} & \cdots & \frac{\partial e_{ij}(X)}{\partial X_j} & \cdots & 0 \end{bmatrix}
\]

The partial derivative of \( e_{ij}(X) \) to \( X_i \) is shown in equation (6):

\[
\frac{\partial e_{ij}(X)}{X_i} = J_i = \begin{bmatrix} -R^T(\theta) \frac{\partial R^T(\theta)}{\partial \theta} \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \\ 0 \end{bmatrix} = \begin{bmatrix} -R^T(\theta) \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \\ 0 \end{bmatrix}
\]

\( J_{ij} \) is the Jacobian matrix of the error function relative to \( X \). According to equation (1) and
The partial derivative of \( e_y(X) \) to \( X_j \) is shown in equation (7):

\[
\frac{\partial e_y(X)}{X_j} = J_j = \begin{bmatrix} R^T(\theta_j) & 0 \\ 0 & 1 \end{bmatrix}
\] (7)

Combined equation (5), equation (6) and equation (7), we can get the exact expression of the first-order approximate of the error function. Substituting equation (4) into equation (3), the objective function becomes:

\[
F(X + \Delta X) = \sum (e_y(X) + J_y \Delta X) \Lambda_y (e_y(X) + J_y \Delta X)
\]

\[
= \sum (e_y^T \Lambda_y e_y + 2e_y^T \Lambda_y J_y \Delta X + \Delta X^T J_y^T \Lambda_y J_y \Delta X)
\]

\[
= \sum c_y + 2b_y^T \Delta X + \Delta X^T H_y \Delta X
\]

\[
= c + 2b^T \Delta X + \Delta X^T H \Delta X
\] (8)

after the error function is linearized, the objective function becomes a quadratic function. Therefore, the minimum value of the objective function can be solved by making the derivative equal to 0:

\[
\frac{\partial F(X + \Delta X)}{\Delta X} = 2H \Delta X + 2b = 0
\] (9)

Obviously equation (9) is a linear equations that can be easily solved. The solution obtained here is the minimum value after the objective function is linearized, and it only holds in the neighborhood of the current solution, so iterative iteration is required until the solution converges. For simplicity, We use g2o\(^{[20]}\) to solve the nonlinear least squares problem.

4.2 Map Update

When localizing robot, it is assumed that there is a static map of the environment. if there is no semi-static object in the environment, the direct matching of the map and laser data can achieve stable localization. Obviously, if the environment changes, the scan-matching will be failed. Therefore, this paper uses scan-matching result as an indicator to determine whether the environment has changed. If the scan-matching failed, it means that the environment has changed, so we use the scan-matching failure as a condition to trigger the map update. The map update process is shown in Figure 5:

\[\text{Figure 5. map update flow-chart}\]
When the environment has not changed, the robot uses the current data to match the static map to obtain the robot pose. When the environment changes, the map matching will be failed, and the robot pose can only rely on the inter-frame scan-matching, and cannot be corrected by the map matching. The failure of the map matching indicates that the environment has changed, and the static map can no longer reflect the real situation of the environment. At this time, the current observation data needs to be used to update the map. Therefore, the positioning system will cache the lidar data when the matching fails, the cache data forms a pose graph. When the robot re-enters the known area, the loop detection is successful, forming a global loop-closure constraint. Then we can use the pose-graph optimization to correct the robot trajectory and update map.

In the pose graph, all poses have cumulative errors, except for the first node and the last node. Pose-graph optimization is shown in Figure 6:

![Figure 6. Pose-Graph Optimization](image)

As can be seen from Figure 6, the robot is running in an office environment, and a part of the map of the environment is erased. Therefore, when the robot walks in the environment corresponding to the erased part, the map matching will fail. The blue trajectory is the trajectory of the map matching failure. The pose error of the blue trajectory is accumulated, and the accumulated error of the last node is the largest. When the robot enters a known area, the loop detection module will find that the last node matches the map successfully, therefore the last pose is corrected and a loop-closure is formed, the black edge in Figure 6 is the loop-closure constraint. After pose-graph optimization, the error of the entire trajectory has been corrected, and the green trajectory is the corrected trajectory.

Since the first node and the last node are matched with the static map, the optimized pose graph coordinate is aligned with the static map. Here, we can consider the pose graph as a submap composed of semi-static objects, and the static map represents the global map. The process of updating the map is the process of fusing the local submap with the global map. In order to minimize the ambiguity in the process of map fusion, we divide the fusion into two module: clearing module and adding module, that is, first clearing the objects that do not exist in the map, and then adding new objects.
First, using the pose-graph and laser data, we can generate a local grid map and the map identifies the free areas and occupied areas. For the occupied area in the global map, if it is falls in a free area in the local submap, it means that the object exists in the global map, but it does not exist in the local submap. It’s means that the global map object was remove. Through this step, we can clear those objects that no longer exist in the global map. After that, we add the semi-object into global map. In order to prevent ambiguity in the fusion map due to positioning errors, if the object in the local sub-map is less than a threshold from the object in the global map, it is considered that the object is caused by positioning errors and is not a real change, so this Objects are directly ignored and not added to the global map.

The proposed map updating method first uses the pose-graph optimization to align the two coordinate, and generates an occupied grid map, then fusion the two occupied grid map. The complexity of pose graph optimization is proportional to the number of nodes. The actual environment changes slowly, Therefore, the number of nodes in the pose graph is generally less than 100. For this order of nodes, the pose-graph optimization is less than 10ms on the Pentium processor. The generation of occupied maps with known robot pose is basically less than 5ms. Because the coordinate of the local submap and the global map are aligned, the submap fusion only needs to traverse the local submap once. the submap fusion costs 10ms ~20ms with 5cm resolution. Therefore, the total time of the proposed map update method is 25ms~35ms, and it can be run in real-time. Because it only store the occupied map instead of the original laser data, the fusion map’s memory consumption is not exceeding 10M. Therefore, this algorithm is very suitable for commercial robots with relatively limited computing resources and storage resources.

5 Experiment
In order to verify the effectiveness of the proposed localization algorithm, we integrates the localization module, mapping module, planning module and control module to form a complete navigation system. The performance of the proposed method has been fully tested in a dynamic environment. The test platform of this paper is the prototype of the automatic sales robot FANBOT, as shown in Figure 7:
The platform is a differentially driven wheeled robot, which is equipped with a variety of sensors, including RGBD sensors, binocular cameras, 2D lidar, etc. This paper only uses 2D lidar data. The processor of this prototype is a Pentium quad-core processor.

5.1 High Dynamic Object Test

High dynamic object test scenario is that the robot running in the environment, and there are pedestrians walking back and forth in front of the robot, both moving away from the robot and approaching to the robot. Without filtering dynamic objects, the point cloud map generated by the robot is shown in Figure -8:
As can be seen from Figure 8, pedestrians have caused very serious interference to the point cloud map generated by the robot. The point cloud map contains data caused by a large number of dynamic objects, which makes the point cloud map more messy and disorderly, which will seriously affect the localization performance. The point cloud map filtered by the delayed comparison algorithm is shown in Figure 9:

**Figure 8.** Point-cloud map without filtered

**Figure 9.** Point-cloud map filtered by delayed comparison

As can be seen from Figure 9, the proposed method basically filters out dynamic objects, which greatly improves the quality of the generated point cloud map. It is worth noting that, in
addition to filtering out pedestrians, some laser points that are not pedestrians are also filtered out. This is because FANBOT uses low-cost lidar. The data quality is poor, and normal data will also contain a lot of data noise. Therefore, it is filtered as a dynamic object by the algorithm. From the comparison between Figure 8 and Figure 9, we can see that the proposed method can successfully filter out highly dynamic objects and greatly improve the quality of point clouds.

5.2 Map Update Test

The semi-static object test is conducted in an office environment. First, a static map of the office scene is constructed, and then the static map is artificially edited, and a part of the map is deleted, which is equivalent to constructing a map of only half of the scene. The test static map is shown in Figure -10:

![Figure 10. test static map](image)

The test method for map update is to let the robot walk in the environment and see if its localization system can construct the other half of the object in the office. We let the robot walk 5 times in the office, and the resulting map is shown in Figure 11:
It can be seen from Figure 11 that the proposed algorithm successfully constructed half of the deleted objects, proving that the algorithm can adaptively update the map after the environment changes, so that the map is always consistent with the current environment, while allows the robot to achieve long-term stable localization.

5.3 Long-term Localization Test
FANBOT’s main operating environment is shopping malls, which are full of various highly dynamic objects and semi-static objects. FANBOT is running in shopping malls during one year, except for artificially forcing the robot to cause robot position loss, the robot has never lost its position due to the dynamic change of the environment, and its localization system can adaptively update the environment map. The result of FANBOT’s updated map in a shopping mall in Luzhou is shown in Figure 12:

Figure 12. (a) original map  
Figure 12. (b) updated map

Figure 12(a) shows a submap near the elevator of a shopping mall. After the robot has been running for a period of time, the staff of the shopping mall placed several doll grab machines
at that location. Figure 12(b) shows the updated map after the machine is placed. It proved that FANBOT’s localization system automatically updates the semi-static object into the environment map. The operation results of FANBOT show that the proposed algorithm is adaptable to the dynamic environment and can realize stable long-term localization in the dynamic environment.

5.4 Localization Reliability Analysis
After a long period of operation in a shopping mall environment, we found that the three factors that have the greatest impact on robot positioning reliability are the following: tire wear, kidnap, and the limitation of lidar sensor perception.

The diameter of the robot tire will be calibrated at the factory. As increase of the running time, the wear of the tire becomes more serious, which resulting in changes in the radius of the tires. The difference between the real diameter and the calibrated diameter is getting bigger and bigger, which leads to the worsening accuracy of the robot's odometer. The accuracy of the odometer is a very important sensor for the robot localization system, which not only affects the localization accuracy when the environment changes, but also affects the accuracy of the initial solution of pose graph optimization. Therefore, preventing the wear of the robot tires, or automatically calibrating the tire diameter is very important for the reliability of robot localization.

Kidnap is another important factor that affects the reliability of robot localization in actual operation. Because pedestrians are curious about robots, there are often people who push the robot from one place to another, which will cause the robot localization failed. And, the amount of information of the lidar data is not rich enough, so that the robot cannot quickly perform global localization, so it’s requires human intervention to recovery robot position. Therefore, the fusion of lidar data and visual sensor data to solve the problem of kidnap is the future research direction.

Lidar is an optical sensor, so it cannot see transparent glass. However, there is a large amount of transparent glass in the shopping mall environment, so there will always be situations in which no objects can be detected by lidar. At this time, the robot pose is purely dependent on the odometer, which causes the positioning of the robot to drift. Therefore, the fusion of multiple sensor data, so that at least one sensor can observe the data, has very positive impact on the stability of robot localization.

6 Conclusion
This paper proposed a localization algorithm, which can process highly dynamic objects and semi-static objects in the environment at the same time, and achieve stable long-term localization in the dynamic environment. We use the feature that the localization system does not need to corrected in real-time, and propose a delay comparison method that can handle two dynamic objects moving away from the robot and approaching to the robot. Besides, we propose a method that combines pose graph optimization and occupied mapping to update the map. Experiments show that this method can handle semi-static objects in the environment, and keep the robot map consistent with the current environment. And through a series of experiments, we proved that the proposed algorithm can make the robot achieve stable long-term localization in the actual dynamic environment. Finally, we summarize the three major factors that affect the localization reliability during the actual operation. The future work is mainly to further optimize these three factors.
References

[1] Gao Yun-feng, Zhou Lun, Lü Ming-rui and Liu Wen-tao 2013 Research overview of indoor location methods for autonomous mobile robots. *J. Transducer and Microsystem Technologies*. 32 1-5

[2] Wang Z., Liu M and Zhang Y 2019 Mobile localization in complex indoor environment based on ZigBee wireless network. *J. Physi Conference Series*. 1314 1221-4

[3] Kuutti S, Fallah S and Katsaros K 2018 A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet of Things Journal*. 11 1

[4] Dellaert F, Fox D and Burgard W 1999 Monte carlo localization for mobile robots. *IEEE Int Conf on Robotics and Automation* (Detroit: Institute of Electrical and Electronics Engineers) pp 1322-28

[5] Fox D, Wolfram B and Sebastian T 1999 Markov localization for mobile robots in dynamic environments. J. Artificial Intelligence Research. 11 391-427

[6] Hhnel D, Triebel R and Burgard W 2003 Map building with mobile robots in dynamic environments. *IEEE Int Conf on Robotics and Automation* (Taipei: Institute of Electrical and Electronics Engineers) pp 1557-63

[7] Chen C, Tay C and Mekhnacha K 2006 Dynamic environment modeling with gridmap: a multiple-object tracking application. *IEEE Int Conf on Control, Automation, Robotics and Vision* (Singapore: Institute of Electrical and Electronics Engineers) pp 1-6

[8] Brechtel S, Gindele T, and Dillmann R 2010 Recursive importance sampling for efficient grid-based occupancy filtering in dynamic environments. *IEEE Int Conf on Robotics and Automation* (Anchorage: Institute of Electrical and Electronics Engineers) pp 3932-38

[9] Meyer-Delius D, Hess J M, Grisetti G and Burgard W 2010 Temporary maps for robust localization in semi-static environments. *IEEE Intl Conf on Intelligent Robots and Systems* (Taipei: Institute of Electrical and Electronics Engineers) pp 5750-55

[10] Tipaldi G D, Meyer-Delius D and Burgard W 2013 Lifelong localization in changing environments. The International Journal of Robotics Research 34 1622-78

[11] Sualeh, M and Kim G. Dynamic 2019 Multi-LiDAR based multi-object detection and tracking. *Sensors* 19 1474

[12] Grisetti G, KuMnerle R, Stachniss C and Burgard W 2010 A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine* 2 31-43.

[13] Olson E B 2009 Real-time correlative scan matching. *IEEE Int Conf on Robotics and Automation* (Kobe: Institute of Electrical and Electronics Engineers) pp 4387-93

[14] Serafin J and Grisetti G 2017 Using extended measurements and scene merging for efficient and robust point cloud registration. *Robotics and Autonomous Systems* 92 91-106.

[15] Collins T, Collins J and Ryan C 2007 Occupancy grid mapping: An empirical evaluation. *IEEE Conf on Control and Automation* (Athens: Institute of Electrical and Electronics Engineers) pp 1-6

[16] Merfels C and Stachniss C 2017 Sensor Fusion for Self-Localisation of Automated Vehicles. *Journal of Photogrammetry Remote Sensing and Geoinformation Science* 85 113-126
[17] Lazaro Maria T, Capobianco R and Grisetti G 2018 Efficient Long-term Mapping in Dynamic Environments. *IEEE International Conference on Intelligent Robots and Systems* (Madrid: Institute of Electrical and Electronics Engineers) pp 153-60

[18] Boniardi F, Tim C and Kümmerle R 2019 A pose graph-based localization system for long-term navigation in CAD floor plans. *Robotics and Autonomous Systems* 112 84-97.

[19] Valencia R, Saarinen J and Andreasson H 2014 Localization in highly dynamic environments using dual-timescale NDT-MCL. *IEEE Int Conf on Robotics and Automation* (Hong Kong: Institute of Electrical and Electronics Engineers) pp 3956-62

[20] Kummerle R, Grisetti G and Strasdat H. 2011 G2o a general framework for graph optimization. *IEEE Int Conf on Robotics and Automation* (Shanghai: Institute of Electrical and Electronics Engineers) pp 3607-13