High-Precision Indoor Robot Dynamic Obstacle Detection with Laser and Camera

Chuanqi Hu*

College of Mechanical Engineering, Shijiazhuang Tiedao University, Shijiazhuang, China

*Corresponding author: huchuanqi@stdu.edu.cn

Abstract. Indoor robot is a very common robot, and its main navigation method is laser SLAM. In the process of detecting dynamic obstacles with laser SLAM, the clustering effect is not obvious, the accuracy is low, and the real-time performance is poor. To solve this problem, a sensor fusion method of lidar and monocular camera is proposed, which improves the accuracy of detecting the position of moving obstacles. When processing lidar data, first preprocess the data, then use the BDSCAN algorithm to cluster the point cloud, and finally use the Kalman filter to predict and calculate the location of the moving obstacle. When processing the data of the monocular camera, the YOLO v3 algorithm is used to quickly detect the target and return the position of the monocular camera. After obtaining the camera and laser data, according to the installation position of the radar and camera, map the position calculated by the radar to the coordinates of the camera. If the coordinate difference after mapping is small, it can be considered that the two sensors have detected the same target. The result is returned to the display interface. Using this method, the efficiency and accuracy of indoor robots in detecting moving obstacles have been greatly improved.

Keywords: SLAM, DBSCAN, YOLO, Indoor robot, Sensor Fusion, ROS, Dynamic Obstacle.

1. Introduction

With the rapid development of SLAM technology and artificial intelligence, the field of indoor robot and outdoor robot navigation has developed rapidly in recent years. Many scholars are actively exploring more efficient and accurate robot navigation methods. The accuracy of dynamic obstacle detection for indoor robots directly affects the accuracy of mapping, and further affects the decision-making accuracy of reachable areas and path planning. If there is sufficient accuracy, you can use the detection of moving obstacles for tracking.

For indoor robots, pedestrians, pets, etc. can be regarded as small moving obstacles. If the detection accuracy of small moving obstacles is improved, the experience of using indoor robots can be greatly improved.

For more than ten years, scholars all over the world have made many breakthroughs in the research of moving obstacle detection. Indoor robots and outdoor robots usually use sensors such as millimeter wave radar, lidar, monocular camera, binocular camera, depth camera, and IMU to detect the
surrounding environment. Among them are mainly divided into visual SLAM and laser SLAM. Due to the large changes in indoor lighting and darkness, shadows may appear, and it is difficult to accurately identify dynamic obstacles simply using a monocular camera. Lidar is less affected by light changes and has high accuracy, and can be used as the main sensor for detecting dynamic obstacles. However, lidar is expensive. When measuring at longer distances, the point cloud is not dense enough, and the sensitivity to obstacles is not high, which is why it cannot be a perfect task for detecting obstacles. Therefore, the method of sensor fusion is a fast and accurate way to detect dynamic obstacles.

Today’s main robot navigation methods are divided into visual SLAM [1] and laser SLAM [2]. In the field of laser SLAM, Azim [3] used an octree-based grid method to detect moving objects, and then clustered them with the DBSCAN algorithm which is computationally intensive and complex. Zhang J [4] uses the minimum number of clusters, and the search radius cannot be adjusted at any time. With this method of processing data, the clustering effect will gradually become worse as the distance is farther. Joseph Redmon created YOLO, an algorithm with extremely high speed and accuracy [5]. In the field of sensor fusion, N. Dalal and B. Triggs [6] proposed the HOG algorithm, which can be used to detect human images and is also used in robot visual navigation with good results. Tan Y [7] used HOG features and SVM classifiers to identify vehicle targets, and used millimeter wave radar to determine the depth information of the targets. The KITTI data set provides six hours of traffic scenes, and provides data formats and procedures for the recording platform [8].

In this paper, the method of detecting dynamic obstacles for indoor robots is divided into three parts: laser detection, visual detection and sensor information fusion. The main improvements include the addition of background point removal to the traditional DBSCAN algorithm to reduce the amount of calculation, the use of Kalman filtering to calculate the position of dynamic obstacles more accurately, and the addition of sensor fusion to increase the accuracy of detection. To a certain extent, the speed and accuracy of detecting dynamic obstacles are improved.

2. Detect dynamic obstacles with laser SLAM

This article uses the RPLIDAR A1 lidar to obtain environmental information. The PLIDAR A1 lidar has a measurement frequency of 10 Hz. It can measure 1000 points each time and can output the location information of the surrounding environment with itself as the origin of the coordinate. This information includes location information and echo strength.

![Detecting obstacles with laser SLAM](image)

**Fig. 1** The Process of Detecting Obstacles with Lidar

2.1. Preprocess Data

Let \((d, s)\) be the radar area of interest, \((m, n)\) be the resolution of the raster map. The size of each grid is set to 10cmx10cm. The original point cloud data of 3D lidar is on \(O_{xy}\) coordinates with lidar as the origin of the coordinates. The grid coordinates are \(O_{uv}\). The relationship between radar coordinates and grid coordinates is Eq. (1)
In order to effectively improve the accuracy of obstacle clustering and reduce the amount of data and calculation, the maximum and minimum height map method is used to remove the background point cloud data.

Let the returned point cloud data be Eq. (2)

\[ P_i = \{x_p, y_p, z_p, I_p\} \]  

That is, x coordinates, y coordinates, z coordinates, echo intensity. The data of each raster is Eq. (3)

\[ G_j = \{x^l_{G_j}, y^l_{G_j}, x^h_{G_j}, y^h_{G_j}, n_{G_j}, g_{G_j}\} \]  

That is, the coordinates of the vertex of the lower left corner of the grid \( x^l_{G_j}, y^l_{G_j} \), the coordinates of the vertex of the upper right corner of the grid \( x^h_{G_j}, y^h_{G_j} \), grid number, Grid occupation status. If

\[
\begin{align*}
    &x^l_{G_j} < x_p < x^h_{G_j} \\
    &y^l_{G_j} < y_p < y^h_{G_j}
\end{align*}
\]  

It can be judged that \( P_i \) is in the corresponding grid.

The expressions for the maximum and minimum heights of all point cloud data projected to the same grid are \( h_{j_{\text{max}}} \) and \( h_{j_{\text{min}}} \). If the difference between the maximum value and the minimum value is less than the threshold \( \xi \), it can be considered that the data in this grid is highly consistent, so the point cloud data in this grid can be eliminated. Dynamic obstacle data such as humans and pets can be retained. Through this method, the point cloud data volume is reduced without affecting the characteristics of dynamic obstacles.

2.2. Obstacle Clustering

The point cloud clustering of lidar generally uses the clustering method based on the DBSCAN (Density-Based Spatial Clustering of Applications with Noise). This clustering method can divide regions with sufficiently high density into clusters, which is very suitable for lidar to search for dynamic obstacles. DBSCAN is less affected by noise and can find clusters of any shape.

The DBSCAN algorithm divides the point cloud data into core points, boundary points, and noise points. The main parameters are the neighborhood \( \xi \) and the density threshold \( \rho \). The core idea is to start from a selected core point and continuously expand to the area where the density is reachable, so as to obtain a maximum area containing the core point and the boundary point, and any two points in the area are connected by density.

Then we process the point cloud data collection \( D \):

- Use the neighborhood \( \xi \) of the point \( a_i \) in the point cloud to search. If the neighborhood of this point contains more points than the density threshold \( \rho \) then a new cluster with the point \( a_i \) as the core object can be created. All point cloud data located in the neighborhood of the \( a_i \) point is regarded as objects with reachable density for the \( a_i \) point.
• If the density of a point $a_i$ to point $a_0$ is reachable, and there is an object chain $a_i, a_2, a_3, ..., a_n$, if

$$\begin{align*}
    a_i & \subset D \\
    1 < i < n
\end{align*}$$

(5)

$a_{i+1}$ is directly arrived density up to $a_i$ about the neighborhood $\xi$ and the density threshold $\rho$, then the object $a_n$ is directly arrived density up to the object $a_i$ about the neighborhood $\xi$ and the density threshold $\rho$.

• Density reachability is the transitive closure of direct density reachability. This relationship is asymmetrical, and only the core objects can reach each other in density. Density connection is a symmetrical relationship.

• Repeat the previous steps until each point cloud data has its own cluster.

• Considering that the laser radar has different densities for the same object at different distances, the returned point cloud data has different densities, so the density threshold $\rho$ should increase with the increase of the distance. Here we use the empirical formula

$$\rho = C_\theta \theta^d$$

(6)

2.3. Dynamic Obstacle Calculation

If the detected obstacles are the same obstacle, we should associate them. This article uses the position change $\delta_x$, the aspect ratio changes $\delta_r$, and the point cloud density change $\delta_\rho$ to make the association of obstacles.

In $p_i^j$ is the existence state of each point in the $i$th area at time $t$. If $p_{i+1}^j = 1$ exists, $p_{i+1}^j = 0$ does not exist. $\sum p_i^j$ is the number of point clouds in the area.

$$\begin{align*}
    \delta_x &= \sqrt{(x_i^{j+1} - x_i^j)^2 + (y_i^{j+1} - y_i^j)^2} \\
    \delta_r &= \frac{r_i^{j+1} - r_i^j}{r_i^j} \\
    \delta_\rho &= \frac{\sum p_{i+1}^j}{\sum p_i^j}
\end{align*}$$

(7)

Use criterion

$$D(i, j) = c_1 \delta_x + c_2 \delta_r + c_3 \delta_\rho$$

(8)

to determine whether it is the same obstacle.

Use Kalman filter to predict the location of obstacles, the prediction equation of Kalman filter is

$$X_{i+1} = AX_i + \omega_i$$

(9)

$A$ is the state transition matrix, the time interval is 0.1s, and $\omega_i$ is noise and satisfies the Gaussian distribution.
\[
X_t = \begin{bmatrix}
  x_t \\
  y_t \\
  v_x^t \\
  v_y^t
\end{bmatrix}, \quad A = \begin{bmatrix}
  1 & 0 & T & 0 \\
  0 & 1 & 0 & T \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\]

(10)

So, the predicted value and predicted covariance equation is

\[
\begin{align*}
X_{t+1|t} &= AX_{t|t} \\
P_{t+1|t} &= AP_{t|t}A^T + Q
\end{align*}
\]

(11)

Since the speed information of the indoor robot can be directly obtained from the data returned by the IMU, the speed information of the dynamic obstacle can be obtained indirectly by the position information at the moments \( t-1 \) and \( t \).

In order to improve the accuracy of dynamic obstacle speed estimation, the least square method of multi-frame position difference can be used to determine the speed at this time. When a dynamic obstacle appears for the first time, the speed is considered to be 0; when the obstacle appears 2, 3, 4, or 5 times, a function fitting is performed. When the obstacle appears more than 5 times, that is, 5 frames ahead in the above case, the previous 5 frames of data are used for secondary fitting to obtain speed data.

3. Detect Dynamic Obstacles with Monocular Camera

For dynamic obstacles, there are requirements for multi-scale detection, multi-target detection and fast calculation speed. This paper uses YOLOv3 to detect dynamic obstacles.

**Fig. 2 YOLO v3 Structure**

3.1. Input Matrix and Output Matrix

Use the camera to capture an image of 416×416, and regard the coordinates of the upper left corner of the image as the origin of coordinates (0,0).

In the structure of YOLO v3, there is no pooling layer and fully connected layer, and the change of the tensor size is achieved by changing the step size of the convolution kernel. In YOLO v3, a picture is reduced five times. The input of the image is 416×416, and the output is 13×13.
3.2. Multi-scale Prediction
According to the returned image matrix, you can obtain 3 feature maps with a depth of 255, 3 feature maps with different scales are $y_1$ of $13 \times 13 \times 255$, and $y_2$ of $26 \times 26 \times 255$. $Y_3$ of $52 \times 52 \times 255$. Because each box has five basic parameters of abscissa x, ordinate y, length w, width h, confidence, probability of 80 categories, and three colors of RGB, the depth is $3 \times (5 + 80) = 255$.

In order to connect two feature maps with different scales, upsampling of $2 \times 2$ is used to ensure that the scales of the stitched tensors are the same. The two stitches are $26 \times 26$ scale stitching and $52 \times 52$ scale stitching respectively.

With feature maps of different scales, the distance of the object in the image does not affect the prediction result. The finer grid cell can detect finer objects.

3.3. Bounding Box Prediction
YOLOv3 uses k-means clustering to obtain the initial bounding box size on the image data set. A total of 9 lengths and widths are selected as follows:

\[(10,13),(16,30),(33,23),
(30,61),(62,45),(59,119),
(116,90),(156,198),(373,326)\]  

(12)

Use multiple Logistic classifiers to classify the target on these 9 lengths and widths. This is a two-class classifier. Logistic regression is used to score the part surrounded by the anchor, that is, how likely the location is to be the target. For every Logistic regression, a $(x, y, w, h, c)$ will be output, that is, the $Ox$ coordinates, length, width, and confidence of a certain position.

4. Fusion Sensor Information
Sensor fusion is the matching process of time and space information.

In the process of time synchronization, the indoor robot moves at a low speed, and the scanning rate of the radar is relatively stable, so the scanning time of the radar is used as the reference. The camera acquisition is triggered every scan to ensure the consistency of the lidar and camera data. The point cloud data is located in the lidar coordinate system $O_XYZ_l$ centered on the lidar while the visual image information is located in the camera coordinate system $O_XYZ_c$ centered on the camera. Due to the different installation angles and positions, the coordinates need to be calibrated and converted

\[
\begin{pmatrix}
    x_v \\
    y_v \\
    z_v
\end{pmatrix} = R \begin{pmatrix}
    x_c \\
    y_c \\
    z_c
\end{pmatrix} + T,
\]

(13)

$R$ is the rotation matrix, and $T$ is the displacement matrix.

Due to the previous removal of the background points of the point cloud, the number of point clouds is small. After DBSCAN clustering, the position and coordinates of dynamic obstacles in the lidar coordinate system can be obtained. Then through coordinate calibration, the point cloud data of the lidar is projected on the image, and the border of the obstacle is
\[ F_i = \begin{pmatrix} u_i \\ v_i \\ l_i \\ w_i \end{pmatrix} \] (14)

Among them, \((u_i, v_i)\) is the coordinates of the upper left corner of the frame, and \((l_i, w_i)\) is the size of the frame. Through the judgment formula

\[ \left| \frac{u_i - u_c}{u_c} \right| + \left| \frac{v_i - v_c}{v_c} \right| + \left| \frac{l_i - l_e}{l_e} \right| + \left| \frac{w_i - w_e}{w_e} \right| < \sigma \] (15)

If the judgment formula is established, it is considered that the pair of frame information is matched successfully, and the laser and the camera successfully recognize the same target.

5. Experimental Results

5.1. Experiment Environment

This article uses the Raspberry Pi 3b platform as the ROS host, STM32 as the indoor robot main control board, and Ubuntu 16.04 as the ROS slave as the display experiment results.

In view of the slow-moving speed of indoor robots and the small detection range, the Silan A1 radar is used. The radar has a measurement radius of 12m, a scanning range of 360°, and a measurement frequency of 8000Hz.

The camera uses an ordinary monocular camera.

The coordinates of the camera and the lidar are relatively fixed, and a rigid connection is adopted. The position and angle of the robot are fixed, and the ROS function is used for calibration. The algorithm in this paper is implemented under the Raspberry Pi system. ROS is the abbreviation of Robot Operating System. It is a combination of communication mechanism, library code and agreed protocol. It aims to simplify the complexity and difficulty of creating complex robots across robot platforms. The communication logic of each module under ROS is shown in Fig.3

![Fig. 3 The Communication Logic of ROS](image)

5.2. Experiment Process

This article first uses an indoor robot in a real scene to detect the position of a fixed obstacle. During the movement of the robot, the position of the fixed obstacle is continuously calibrated and the detection result is recorded. At the same time, the IMU is used to calculate the position of the obstacle, and the measurement result of the IMU is compared with the result of the sensor fusion.

After measuring the position of the fixed obstacle, start moving the obstacle. Let the robot detect dynamic obstacles while moving, and record the detection results.
5.3. Detect the Location of Fixed Obstacles

In the scene shown in the figure below, six locations are set to detect obstacles. The robot travels randomly at a steady speed.

In order to comprehensively display the information changes and information of obstacles, the observation results of obstacles in 6 different positions of the robot are selected.

The following figure shows the detection results of the robot in different positions. Fig.4 are for short distance and long-distance facing obstacles respectively. Fig.5 are for short distance front right and left front facing obstacles. Fig.6 show the detection results of obstacles on the side at different angles in the camera and radar respectively.

![Fig. 4 The First Observation Position](image)

![Fig. 5 The Second Observation Position](image)
5.4. Detect the Location of Fixed Obstacles
This paper not only realized the display of static obstacles during the experiment, but also recorded the position of moving obstacles, and compared them with the detection results of IMU.

Some of the experimental results are shown in the Fig.7 and Fig.8.

5.5. Analyze Experimental Results
It can be seen from the figure that in this experiment, the maximum estimated value of the measurement error in the x direction is 58cm, and the maximum estimated error in the y direction is 63cm. The algorithm is for the target in the real scene the recognition and tracking accuracy is high.

In the process of constant changes in the external environment, the algorithm can still detect the locations of moving obstacles and fixed obstacles more accurately, and return accurate values. The
target can be detected quickly even when the position of the moving obstacle changes a large distance, which has certain robustness.

However, there are some obvious disadvantages when using this method. As the detection time becomes longer and the moving distance of the indoor robot increases, there will be significant accumulation of errors in both the x-axis direction and the y-axis direction. As time changes, larger errors will appear. This is because when processing the feature point data provided by the laser and camera, the accumulation of errors is not considered, and the reliability of the long-term estimation results will be reduced.

6. Conclusions

In view of the low moving speed of indoor robots and low accuracy in judging dynamic obstacles, this paper designs an information fusion method based on lidar and monocular camera to ensure a fast and accurate method of identifying dynamic obstacles. This method also has the ability to predict the moving position of dynamic obstacles. It can effectively track dynamic obstacles in an indoor environment, and has good real-time and robustness. In the next step of the research, we should focus on the processing of accumulated errors before and after sensor fusion to obtain more accurate and reliable obstacle position information.

References

[1] T. Gandhi and M. M. Trivedi, "Pedestrian Protection Systems: Issues, Survey, and Challenges," in IEEE Transactions on Intelligent Transportation Systems, vol. 8, no. 3, pp. 413-430, Sept. 2007, doi: 10.1109/TITS.2007.903444.

[2] Trung-Dung Vu, Julien Burlet, Olivier Aycard. Grid-based localization and local mapping with moving object detection and tracking [J]. Information Fusion,2010,12.

[3] Azim, Asma & Aycard, Olivier. (2014). Layer-based supervised classification of moving objects in outdoor dynamic environment using 3D laser scanner. IEEE Intelligent Vehicles Symposium, Proceedings. 1408-1414. 10.1109/IVS.2014.6856558.

[4] J. Zhang and J. Kerekes, "An Adaptive Density-Based Model for Extracting Surface Returns From Photon-Counting Laser Altimeter Data," in IEEE Geoscience and Remote Sensing Letters, vol. 12, no. 4, pp. 726-730, April 2015, doi: 10.1109/LGRS.2014.2360367.

[5] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.

[7] Y. Tan, F. Han and F. Ibrahim, "A Radar Guided Vision System for Vehicle Validation and Vehicle Motion Characterization," 2007 IEEE Intelligent Transportation Systems Conference, Bellevue, WA, USA, 2007, pp. 1059-1066, doi: 10.1109/ITSC.2007.4357754.

[8] A Geiger, P Lenz, C Stiller, R Urtasun. Vision meets robotics: The KITTI dataset [J]. The International Journal of Robotics Research, 2013, 32.