International Core Journal of Engineering Volume 8 Issue 1, 2022
ISSN: 2414-1895 DOI: 10.6919/ICJE.202201_8(1).0064

Research on 3D Mapping Technology of Smart Car based on Depth Camera

Ruoyu Zhao[a], Minyong Tong[b]

School of Automobile and Transportation, Tianjin University of Technology and Education, Tianjin 300222, China

[a] 942075447@qq.com, [b] tong13212113033@163.com

Abstract

Aiming at the defects of the sparse landmark map built by the original ORB-SLAM2 algorithm, such as lack of spatial structure information and inability to realize positioning and navigation, the depth information and color information of the RGB-D camera are spliced to obtain a point cloud map. The point cloud map contains rich environmental information, but the map compression performance is poor. Therefore, on the basis of this, a mapping method based on octree is proposed. Finally, after comparative experiments, the storage space occupied by the octree map at different resolutions has dropped by 85.4% on average; in addition, when the algorithm is running, the CPU load rate has dropped by 2.6% on average, and the memory usage has dropped by 64.1% on average. Compared with the original ORB-SLAM2 algorithm, it has better compression performance and real-time performance, which can basically meet the requirements of navigation.

Keywords

Visual SLAM, 3D Mapping, Octree.

1. Preface

With the maturity of 5G, artificial intelligence, cloud computing, blockcha--in and other technologies, the golden age of the smart car industry is also coming. The application environment of smart cars is complex and changeable, from indoor to outdoor, from paved road to non-paved road, from known environ--ment to unknown environment.

A high-precision, updatable, intelligent map that can reflect the surrounding environment of the vehicle is the basis for the realization of intelligent car navigation[1]. The construction of a three-dimensional map with the aid of a vision camera is a prerequisite for the realization of autonomous positioning and navigation of a smart car. Its essence is the result of splicing and fusion of point cloud images constructed multiple times according to the motion trajectory of the smart car and related coordinate system information.

In 2002, Besl et al. used the ICP (Iterative Nearest Point) algorithm to complete 3D point cloud matching, and then realized 3D map construction [2]. The matching effect is greatly affected by the initial estimate and the moving speed, and the matching effect is best at low speeds. But when the input data is noisy, the accuracy of the mapping is greatly reduced. In 2009, Magnusson et al. open sourced the point cloud registration algorithm NDT (Normal Distributions Transform)[3] for the construction of 3D maps, which was improved based on the iterative nearest point algorithm. For registration, a Gaussian distribution probability estimation algorithm is used, which has achieved good results and can obtain multi-view pose transformation data.

 Volume 8 Issue 1, 2022
DOI: 10.6919/ICJE.202201_8(1).0064

At present, due to the huge amount of calculation of machine vision and the limitation of computing power of the computing platform, a good balance cannot be achieved between its work performance and efficiency, and the robustness and real-time aspects of the system need to be strengthened. In this paper, RGB-D camera is used as the visual sensor to study the 3D mapping method of smart car. First, the defects of the sparse road sign map are analyzed, and the 3D point cloud map is constructed using the depth camera. Finally, the constructed 3D map is converted into an octree map that can be used for navigation, which realizes the improvement of the speed and accuracy of the 3D mapping of the smart car.

2. Three-dimensional Mapping System Framework

2.1 The Defect of Sparse Road Sign Map

The sparse landmark map[4] is the main map form of the original feature point VSLAM algorithm represented by ORB-SLAM2. However, this kind of map construction method can only rely on capturing feature information, such as bookshelves, cabinets, benches and other objects with obvious outline features. It can only obtain the edge corner information of the object, and cannot judge the spatial structure of the object through the feature points, so the effect of mapping is not intuitive, as shown in Fig. 1, the positioning function cannot be realized. The dense map can get richer surrounding environment information [5].

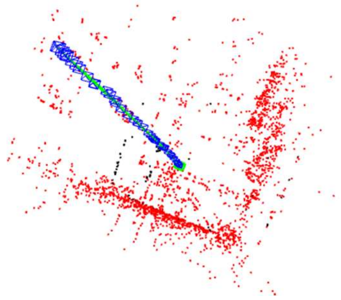


Fig. 1. Sparse Road Sign Map

2.2 Dense Map Construction

2.2.1 Three-Dimensionalpoint Cloud Map based on Rgb-D Camera

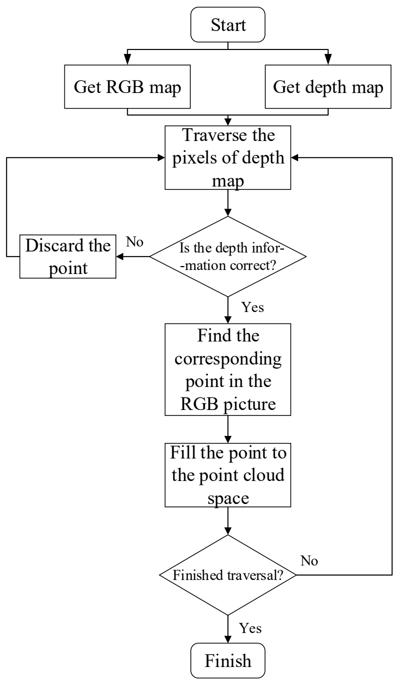


Fig. 2. 3D point cloud map construction process

The three-dimensional point cloud map is the most intuitive and most widely used of dense maps. The point cloud map is obtained by registering and fusing the depth information of the image with the color information, and finally splicing the discrete point clouds through coordinate transformation [6]. The algorithm flow of constructing a 3D point cloud map in this paper is shown in Fig. 2.

Through the theoretical knowledge of visual SLAM, the three-dimensional point cloud coordinates of a single frame image can be obtained as:

$$\begin{cases} z = d / s \\ x = (u - u_0) z / f_x \\ y = (v - v_0) z / f_y \end{cases} \tag{1}$$

Where z is the 16-bit real distance information of the depth map, f_x, f_y, u_0, v_0 ,are the camera internal parameters, s is the camera scale factor, usually s=1000, and the depth measurement unit is converted to meters for easy understanding and calculation. Finally, the pixel RGB information of the color image is filled into the three-dimensional point cloud, and the point cloud data in the format of [x y z r g b] is obtained.

From equation (1), the single-frame 3D point cloud data is $P_i = [x_i y_i z_i]^T, i = 1, 2, 3, \cdots, n$,if the camera pose of the current frame is T, then the the splicing process of the frame and the point cloud of the previous frame can be expressed as:

$$\begin{bmatrix} x_i^{12} \\ x_i^{12} \\ z_i^{12} \\ 1 \end{bmatrix} = \begin{bmatrix} x_i^{1} \\ x_i^{1} \\ z_i^{1} \\ 1 \end{bmatrix} + T \begin{bmatrix} x_i^{2} \\ x_i^{2} \\ z_i^{2} \\ 1 \end{bmatrix} \tag{2}$$

Aiming at the problems of noisy data, irregular and large number of point cloud density, and outliers due to blind spots in the field of view when the depth camera obtains environmental information[7], the generated 3D point cloud map needs to be filtered. Use the PCL point cloud library as a filter, and the version used in this article is 1.7. The main process is as follows:

(1) Smooth the point cloud data: Use Pass Through filter to set the parameters to the depth detection range of the RGB-D camera KinectV2 used in this article (0.5m~4.5m), and eliminate invalid point clouds that exceed the range of the depth sensor;

(2) Removal of outliers: Outliers are points in the point cloud that are partly separated from the plane of the actual object, which is mainly caused by the working error of the sensor. They will cause the calculation of local point cloud features to increase sharply, and then there will be calculation errors, leading to an increase in the probability of point cloud registration failure. At this time, the Radius Outliner Removal filter needs to be used. Its main working principle is to filter according to the number of adjacent points in the radius of the space point, as shown in Fig. 3 below:

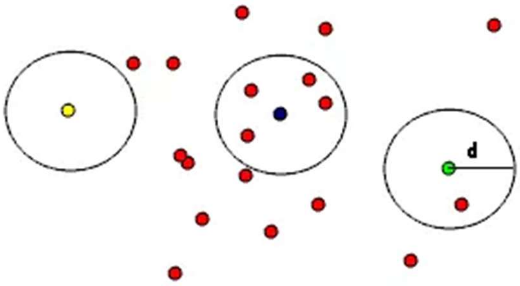


Fig. 3. Radius filtering to remove outliers

(3) Point cloud data down-sampling: Due to the large amount of collection in the process of 3D mapping, storage operations are more difficult. To this end, we use Voxel Grid filter to collect point cloud data from the sensor and create a three-dimensional voxel grid for it one by one, and approximate all points in the grid to a point on the center of gravity of the voxel, so as to correct the point cloud data is compressed, as shown in Fig. 4:

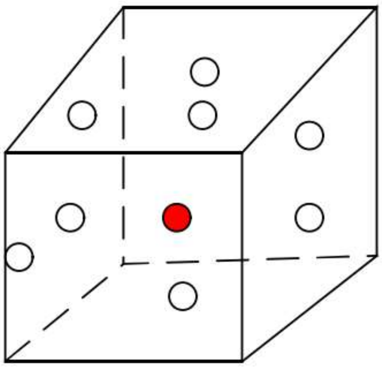


Fig. 4. Schematic diagram of voxel filtering

2.2.2 Existing Problems

Although the point cloud map overcomes the lack of environmental color information and spatial structure of the sparse landmark map, there are still the following problems:

(1) The data storage structure is inefficient and data retrieval is slow;

(2) Unable to complete obstacle detection and two-dimensional map generation, all data in the map are discrete and discontinuous, and cannot represent the space occupation;

(3) When there is noise interference, large errors will occur in the pose calculation, and the point cloud map will have obvious positioning drift. For example, one obstacle may become two, the flat wall may be distorted.

2.3 Octree Map Construction

In view of the problems of dense point cloud maps, this section introduces a three-dimensional mapping method based on octrees on the basis of this section, which has better compression performance, and the map presentation method is concise and clear, combined with the ORB-SLAM2 algorithm, achieve the three-dimensional mapping.

In 1978, Hunter et al. first proposed the concept of Octree [8], which realized the expansion of a flat quadtree to a three-dimensional space. Its essence is a tree-like data model used to express three-dimensional space. The octree divides the objects in the three-dimensional space to form a directional graph with a root node, as shown in Fig. 5:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202201_8(1).0064

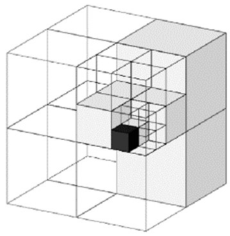


Fig. 5. Schematic diagram of the octree structure

The octree is composed of root nodes, child nodes and leaf nodes[9]. The smallest cube in the figure is the leaf node, the largest is the root node, and the rest are child nodes. Generally, the division starts from the root node, and each node follows the recursive principle of 2×2×2 [10]. This can effectively reduce the storage space occupation and reduce the load real-time computing of the platform, improve the real-time performance of mapping.

3. Comparative Experiment

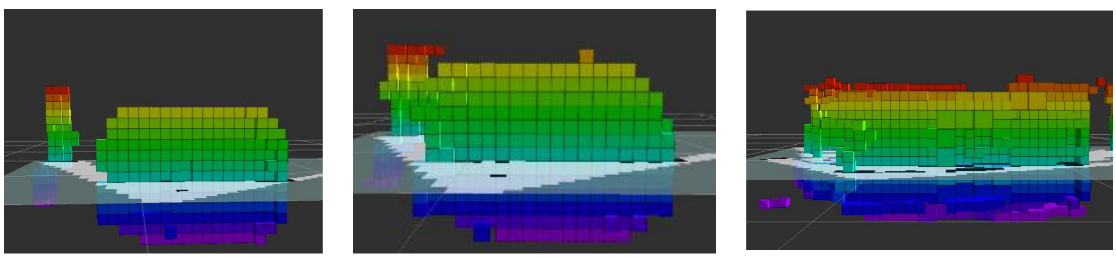
3.1 Map Compression Performance Comparison Experiment

In order to meet the needs of operating conditions such as actual driving, navigation, and obstacle avoidance of smart cars, and to reduce the workload of the car-machine system, the environment map used should occupy as little storage space as possible. Therefore, we hope that the map it uses has better compression performance. For this reason, we use the controlled variable method to compare the map compression performance of the two methods in the same environment, the same hardware equipment, and the same resolution. Use KinectV2 to compare the indoor environment. When building a map, the effect of the 3D point cloud map at different resolutions is shown in Fig. 6:

(a)Resolution 0.05 (b)Resolution 0.03 (c)Resolution 0.001

Fig. 6. Three-dimensional point cloud map at different resolutions

Use the octree map to map the same indoor environment, the effect is shown in Fig 7:



(a)Resolution 0.05 (b)Resolution 0.03 (c)Resolution 0.001

Fig. 7. Octree map at different resolutions

ISSN: 2414-1895 DOI: 10.6919/ICJE.202201_8(1).0064

By comparing the experimental results, it can be seen that the point cloud map describes the scene in detail, so that it includes many detailed information that is not required for autonomous navigation of smart vehicles, such as text and patterns on the wall. At the same time, when the device is moving faster, the point cloud map will have the same cumulative error and drift as the wall on the right in Fig. 7(c), which is not conducive to obtaining accurate environmental spatial information. The storage space occupancy of the above map and the storage space occupancy reduction rate of the octree map relative to the point cloud map are shown in Table 1 below:

Table 1. Storage occupancy

Resolution(m)	0.05	0.03	0.001
Dense point cloud map size(MB)	56.3	127.1	243.5
Octree map size(MB)	12.2	20.7	13.12
Map volume decrease rate(%)	78.3	83.7	94.2

It can be seen from the above table that in the same environment with the same resolution, the compression performance of the 3D dense point cloud map is obviously weaker than that of the octree map, and the lower the resolution, compared to the 3D dense point cloud map. The larger the volume reduction rate of the map, and the reduction of the resolution at the same time, the reduction of the map to the scene increases, and the image is clearer. In the above map, a resolution of 0.01 means that a cube with a side length of 1 cm contains one pixel, and the rest of the values can be deduced by analogy. Therefore, in actual applications, a smaller resolution should be selected according to the needs of the scene to obtain a higher-quality mapping effect. In summary, in terms of map compression performance, under the same experimental conditions, compared to dense point cloud maps, octree maps can greatly save storage space, and the smaller the resolution, the higher the map volume reduction rate. At different resolutions, the volume of the octree map has decreased by 85.4% on average; when the resolution is less than 0.001m, the map volume reduction rate can reach more than 90%.

3.2 Real-Time Comparison Experiment of Mapping

The real-time performance of the mapping algorithm is also an important indicator that affects the mapping effect. In the actual mapping operation, the speed and accuracy of the mapping are largely limited by the processing and computing power of the CPU. Therefore, on the same hardware platform, an algorithm that occupies a smaller CPU load and memory occupancy rate of the host computer can complete the map more smoothly and stably. Therefore, in the above-mentioned control variable experiment, we use system tools to view the system load when the two mapping methods are running. The statistical results of the point cloud map are shown in Table 2:

Table 2. System load when the 3D point cloud map is running

Resolution(m)	0.05	0.03	0.001
CPU load rate(%)	96.8	97.7	100
Memory usage(%)	70.4	79	99.3

The system operating load of the octree map in the same environment, the same hardware equipment, and the same resolution is shown in Table 3:

Table 3. System load when the octree map is running

Resolution(m)	0.05	0.03	0.001
CPU load rate(%)	93.2	95.5	98.1
CPU load drop rate(%)	3.7	2.3	1.9
Memory usage(%)	39.9	60.5	91.9
Memory usage decrease rate(%)	43.3	23.4	7.5

It can be seen from the statistical results of the two tables that with the same input data, as the resolution of the map decreases, the CPU load and memory usage of the three-dimensional point cloud map and the octree map both increase. At the same time, compared with the three-dimensional point cloud map, the CPU load rate and memory usage of the system will decrease when running the octree mapping, but the lower the resolution, the less obvious the decrease. The CPU load rate when the octree map is running at different resolutions dropped by 2.6% on average, and the memory usage dropped by 64.1% on average. In summary, from the perspective of real-time mapping algorithm, the octree map is obviously better than the three-dimensional point cloud map.

4. Conclusion

This article studies the core link of visual SLAM-mapping. Firstly, it introduces the shortcomings of traditional feature point-based sparse landmark maps, that is, the spatial structure of objects cannot be judged by feature points, so the mapping effect is not intuitive, and the positioning function cannot be realized. To solve this problem, the ORB-SLAM2 algorithm is improved and designed. A three-dimensional point cloud map algorithm based on an RGB-D camera is used to register and fuse the depth information and color information of the image, and finally the discrete point clouds are spliced through coordinate transformation to obtain a point cloud map, but the point cloud map cannot be navigated; Subsequently, a three-dimensional mapping method based on an octree is introduced on the basis of the three-dimensional point cloud map. Finally, through comparative experiments, it is proved that compared with the point cloud map and the sparse landmark map, the octree mapping method proposed in this paper has better map compression and real-time map creation, and can basically meet the requirements of navigation.

References

[1] Leslie H, Driverless cars: mapping the trouble ahead [EB/OL]. https://www.ft.com/content/2a8941a4-1625-11e8-9e9c-25c814761640, 2018-2-21.

[2] Besl P J, McKay N D. Method for registration of 3-D shapes[C]//Sensor fusion IV: control paradigms and data structures. International Society for Optics and Photonics, 1992, 1611: 586-606.

[3] Magnusson M. The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection[D]. Örebro universitet, 2009.

[4] Chen Mingyou, Tang Yunchao, Zou Xiangjun, Huang Zhaofeng, Zhou Hao, Chen Siyu. 3D global mapping of large-scale unstructured orchard integrating eye-in-hand stereo vision and SLAM[J]. Computers and Electronics in Agriculture,2021,187.

[5] Yong-bao Ai, Ting Rui, Xiao-qiang Yang, Jia-lin He, Lei Fu, Jian-bin Li, Ming Lu. Visual SLAM in dynamic environments based on object detection[J]. Defence Technology,2020(prepublish).

[6] Schütz Markus, Ohrhallinger Stefan, Wimmer Michael. Fast Out-of-Core Octree Generation for Massive Point Clouds[J]. Computer Graphics Forum,2020,39(7).

[7] Zhao Yipu, Vela Patricio A. Good Feature Matching: Toward Accurate, Robust VO/VSLAM With Low Latency[J]. IEEE Transactions on Robotics,2020.

[8] Donald Meagher. Geometric modeling using octree encoding[J]. Computer Graphics and Image Processing,1982,19(1):129-147.

ISSN: 2414-1895
DOI: 10.6919/ICJE.202201_8(1).0064

[9] Allam Ahmed, Nemra Abdelkrim, Hamerlain Mustapha. Smooth Variable Structure Filter VSLAM[J]. IFAC PapersOnLine,2016,49(15).

[10] Chen Liang, Jin Sheng, Xia Zhoujun. Towards a Robust Visual Place Recognition in Large-Scale vSLAM Scenarios Based on a Deep Distance Learning[J]. Sensors,2021,21(1).