

A LASER-SLAM ALGORITHM FOR INDOOR MOBILE MAPPING

Wenjun Zhang^a, Qiao Zhang^a, Kai Sun^a, Sheng Guo^a

^a Ledor Spatial Information Technology Co., Ltd. Building No.12, Hust Scien&Tech Park, East Lake Hi-Tech Zone, Wuhan, China
- (zhangwenjun, zhangqiao, sunkai, leader3s)@ledor.com.cn

Commission IV, WG IV/7

KEY WORDS: SLAM, Bayes Filter, Graph Optimization, Sub-maps, Virtual Landmark, Indoor Mobile Mapping

ABSTRACT:

A novel Laser-SLAM algorithm is presented for real indoor environment mobile mapping. SLAM algorithm can be divided into two classes, Bayes filter-based and graph optimization-based. The former is often difficult to guarantee consistency and accuracy in large-scale environment mapping because of the accumulative error during incremental mapping. Graph optimization-based SLAM method often assume predetermined landmarks, which is difficult to be got in unknown environment mapping. And there most likely has large difference between the optimize result and the real data, because the constraints are too few. This paper designed a kind of sub-map method, which could map more accurately without predetermined landmarks and avoid the already-drawn map impact on agent's location. The tree structure of sub-map can be indexed quickly and reduce the amount of memory consuming when mapping. The algorithm combined Bayes-based and graph optimization-based SLAM algorithm. It created virtual landmarks automatically by associating data of sub-maps for graph optimization. Then graph optimization guaranteed consistency and accuracy in large-scale environment mapping and improved the reasonability and reliability of the optimize results. Experimental results are presented with a laser sensor (UTM 30LX) in official buildings and shopping centres, which prove that the proposed algorithm can obtain 2D maps within 10cm precision in indoor environment range from several hundreds to 12000 square meter.

1. INTRODUCTION

1.1 Laser-SLAM

In recent years, the technology of simultaneous localization and mapping (SLAM) has been applied in the surveying and mapping domain. In Many places, especially in the indoor environment, there is no GPS signal, SLAM is one of the solution to this problem. Laser scanners are often used in high precision surveying and mapping, Laser range finder is extensively used to acquire 2D information from environment for its high abilities of prevent interference and low influence of light, and many laser-SLAM algorithm have been applied to the robot, but most algorithms are only suitable for small scale and low precision mapping, the new problem with it is that how to draw high precision large-scale map.

About research into SLAM algorithm, the earliest can be traced back to the 80's, Stanford University, Smith [1], who published a seminal article on the issue of SLAM. In this paper, the concept of random map is proposed, which is based on the description of spatial uncertainty and the representation of transformation. By 1991, Leonard proposed the concept of SLAM, Durrant summarized the early research methods of this problem, and discussed the structure and convergence of the SLAM problem. SLAM algorithm is essentially a system state estimation problem. In this way, the solving method can be roughly divided into three categories, which are based on Kalman filter, particle filter and based on graph theory. Kalman filter and particle filter based on the recursive Bayesian state estimation theory, under the condition of the known observation information and control information from the initial state to the current state, estimate the posterior probability of the system. According to the different method of posterior probability representation, a variety of Kalman filtering methods are proposed, like EKF-SLAM algorithm, UKF-SLAM algorithm. Based on the Kalman filter method, it can get a satisfactory result when the Gauss

distribution hypothesis and small non-linearity, and the particle filter does not have such a limit. Rao-Blackwellized particle filters (RBPF) is one of the most representative SLAM algorithm based on particle filter. Base on RBPF, Montemerlo provides FastSLAM algorithm, and propose FastSLAM 2.0 by further optimization.

With the deep study of SLAM, its application has been gradually developed from a small scale to a large scale environment. SLAM algorithm based on graph theory, due to the use of the global optimization of the processing method, can get a better mapping result.

1.2 Particle Filters for Localization and Mapping

In controlling theory, SLAM is a typical dynamic system state estimation problem. The system transfer from a given initial state x_0 and map m_0 to the states $x_{1:t} = x_1, \dots, x_t$ from start to from t moment. In a work by Murphy, Rao-Blackwellized particle filters (RBPF) have been introduced as an effective approach to solve the SLAM problem, this approach is to estimate the posterior $p(x_{1:t}, m | z_{1:t}, u_{1:t-1})$ about the map m and the trajectory $x_{1:t}$. This estimation is performed given the observations $z_{1:t} = z_1, \dots, z_t$ and the odometer measurements $u_{1:t-1} = u_1, \dots, u_{t-1}$. Without the odometer measurements, the Rao-Blackwellized particle filter for SLAM make use of the following factorization

$$p(x_{1:t}, m | z_{1:t}) = p(m | x_{1:t}, z_{1:t}) p(x_{1:t} | z_{1:t}) \quad (1)$$

The SLAM problem is decomposed into two independent posterior probability: The posterior over maps $p(m | x_{1:t}, z_{1:t})$ and the posterior over potential trajectories $p(x_{1:t} | z_{1:t})$, which can be estimated by particle filter. This factorization allows us to first estimate only the trajectory and then to compute the map given that trajectory.

Main process of the RBPF for SLAM is as follows:

1) Sampling: In accordance with Sampling from the proposal distribution $q(x_{1:t}^i | z_{1:t})$, the next generation of particle $\{x_t^i\}_{i=1,\dots,N}$, where N represents the numbers of the particles.

2) Calculating Importance Weighting: Calculating the importance weight $\{w_t^i\}_{i=1,\dots,N}$ of each particle according to the importance sampling principle to remedy the gap between the proposal distribution and the posterior over trajectories.

$$w_t^i = \frac{p(x_{1:t}^i | z_{1:t})}{q(x_{1:t}^i | z_{1:t})} \quad (2)$$

3) Resampling: Particles are drawn with the updated importance weight. This step is based on Monte Carlo Method, only a finite number of particles is used to approximate a continuous distribution.

4) Map Updating: For each particle, the corresponding map estimate $p(m^i | x_{1:t}^i, z_{1:t})$ is computed based on the trajectory $x_{1:t}^i$ of that sample and the history of observation $z_{1:t}$. In this paper, the particle filter is used in the 2.1.1 section, and a map representation and updating method is described in detail.

1.3 Graph Optimization for Localization and Mapping

SLAM problem can also be seen as A problem that inputting the motion model and observation and estimating the maximum posterior probability of the position of robot and landmarks.

$$X^* = \arg \max_x p(X, L | Z) \quad (3)$$

where X = state positions
 L = landmark positions
 Z = observation

The formula (3) can be decomposed into:

$$p(X, L | Z) \propto \underbrace{\prod p(x_{i+1} | x_i, z_i)}_{\text{odometry constraint}} \cdot \underbrace{\prod p(l_j | x_i, z_{ij})}_{\text{landmark constraint}} \quad (4)$$

Under the assumption of Gauss distribution, log on the formula (4) both sides:

$$\begin{aligned} X^* &= \arg \max_x p(X, L | Z) \\ &= \arg \max_x \left\{ \underbrace{\sum \|f(x_i, z_i) - x_{i+1}\|^2}_{\text{odometry constraint}} + \underbrace{\sum \|f(x_i, z_{ij}) - l_j\|^2}_{\text{landmark constraint}} \right\} \end{aligned} \quad (5)$$

Based on the graph theory, the SLAM problem is regarded as the least square optimization problem.

2. RELATED WORK

2.1 New Framework

This section shows a new framework (Sub-maps framework) which combines SLAM Algorithm based on the use of particle filters with graph optimization algorithm. Most SLAM algorithms based on the use of particle filters attempt to maintain a single map with multiple states, and the map is created by the accumulation of observations of the environment and estimates of states (PIC), but the observations are compared against an incomplete and possibly incorrect map, identified with the particle. This approach will inevitably lead to the introduction of cumulative errors. In Sub-maps framework, a single map is divided into many small sub-maps. Each sub-map is just composed of adjacent observations. This can greatly decrease the effect of cumulative errors, and each sub-map has relatively rich topological construction and outstanding features. The relationship among sub-maps can be calculated separately by matching their detail information, than non-consistency of relationships and trajectories of sub-maps make a graph optimization problem, thus the SLAM problem of A single map is translated into several independent SLAM problem of sub-maps and a graph optimization problem (PIC).

2.1.1 Sub-maps: In the sub-maps frameworks, the whole original SLAM task is decomposed to many task of small map, we call such a small map a sub-map. The purpose of this is to avoid the cumulative error generated in a large scale map and improve the parallel efficiency of the algorithm.

Since the laser scanning data for each frame reflects less information, it is difficult to extract and match features in non-adjacent frames with overlapping area, so the traditional SLAM is easy to produce errors, and with the increasing of time, the cumulative error will be increasing. When the map is segmented into sub-maps of reasonable size, there is no cumulative error between sub-maps, and each sub-map has relatively rich topological construction and outstanding feature.

Sub-maps are estimated by using A Particle Filters SLAM like DP-SLAM, we use grid occupied state to describe and update sub-maps. A sub-map is decomposed into grid. In one time scan, the position of the end of the laser scanning line is more likely an obstruction, and the position on the laser scanning line is more likely no obstructions, the rest position is not sure whether it is an obstruction. Assuming that their probabilities are O_o , O_f and O_u .

For any grid point s , o is occupied. The potential trajectories is $x_{1:t} = x_1, \dots, x_t$, and the observations is $z_{1:t} = z_1, \dots, z_t$, After t scans, the probability that the grid s is occupied is:

$$O(o | x_{1:t}, z_{1:t}) = \prod_{i=1}^t O(o | x_i, z_i) \quad (6)$$

Logarithmic:

$$\log O(o | x_{1:t}, z_{1:t}) = \log O_o^c + \log O_o^f \quad (7)$$

Where c_o and c_f represent the number of times the grid is occupied and free.

So, we can make a probabilistic analytic formula of the occupied state of grid s to calculate the posterior over maps $p(m | x_{1:t}, z_{1:t})$ in the equations () like this:

$$p(o | x_{1:t}, z_{1:t}) = \frac{\exp(c_f \log O_f + c_o \log O_o)}{1 + \exp(c_f \log O_f + c_o \log O_o)} \quad (8)$$

When estimating the posterior over potential trajectories $p(x_{1:t} | z_{1:t})$, each observation should be compared with the current incomplete map, so this map have to be kept in memory, and memory consumption will increase with the expansion of the map. In the Sub-maps framework, sub-map calculation is independent of each other, so several sub-maps can be computed simultaneously. We can set the number of sub-maps for parallel computing and the size of sub-map to control the consumption of memory and maintain constant memory consumption.

2.1.2 Virtual landmark: Landmarks are features which can easily be re-observed and distinguished from the environment. Landmarks can be easily set arbitrarily in the simulation of SLAM algorithm. In a real-world environment, we must place a real specific target, but it is not desirable in real applications. DP-SLAM avoid the data association problem by storing multiple detailed maps instead of sparse landmarks, thereby subsuming association with localization, therefore, the authors claim that DP-slam makes no landmark assumptions. This, in turn, can create landmarks through localization of multiple detailed maps, and we call these landmarks Virtual landmarks, because it does not need to set specific landmark in the real scene or extract the features of landmarks. According to the probabilistic analytic formula of the grid occupied map (), a point in the map most likely to be an obstacle is chosen as a virtual landmark, which may not even be distinguishable in visual features.

Create a virtual landmark in order to describe the relationship between the sub-maps, so we only need to be concerned with the relationship between the virtual landmark and the corresponding sub-maps, but not need to care about the absolute position of this virtual landmark itself. The calculation of the virtual landmark, in essence, is to calculate the relationship, which is to calculate the relationship between the virtual landmark and the trajectories of the corresponding sub-map.

The calculation steps of virtual landmark l :

1) Setting the virtual landmark l , l is the most likely to be the grid point of obstacles in the sub-map.

$$l = \arg \max_s p(o | x_{1:t}, z_{1:t}) \quad (9)$$

2) Calculate the relationship z_{ij} between the landmark l_j and the track point x_i , l_j is the landmark of the sub-map j , x_i is the track point in the adjacent sub-maps.

It is clear that the relationship between landmark and track point is independent and does not guarantee consistency, and this problem will be solved by graph theory optimization in section 1.3.

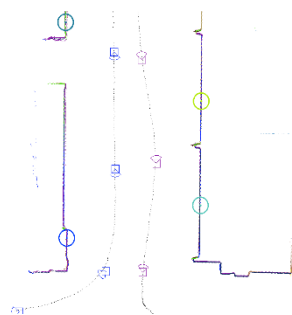


Figure 1. Automatic landmarks. The circles is four automatic landmarks

2.1.3 Optimization strategy: In order to ensure the correctness of the result, we designed two sets of optimization strategies for automatic optimization and manual optimization. Two optimization strategies are used in the same graph theory, but landmark selection is different. The automatic optimization strategy adopts the landmark setting described in section 2.1.2. The manual optimization strategy is to make up the deficiency of the automatic optimization strategy in large scale complex terrain. The manual optimization strategy allows the landmark selection of the corresponding points in several different sub-maps by the way of feature matching or manual selection, then calculate the relations between the points and the locus of map. Manual optimization can be repeated iteratively until the corresponding points can not be found.



Figure 2. Manual landmarks. The circles is a pair manual landmarks

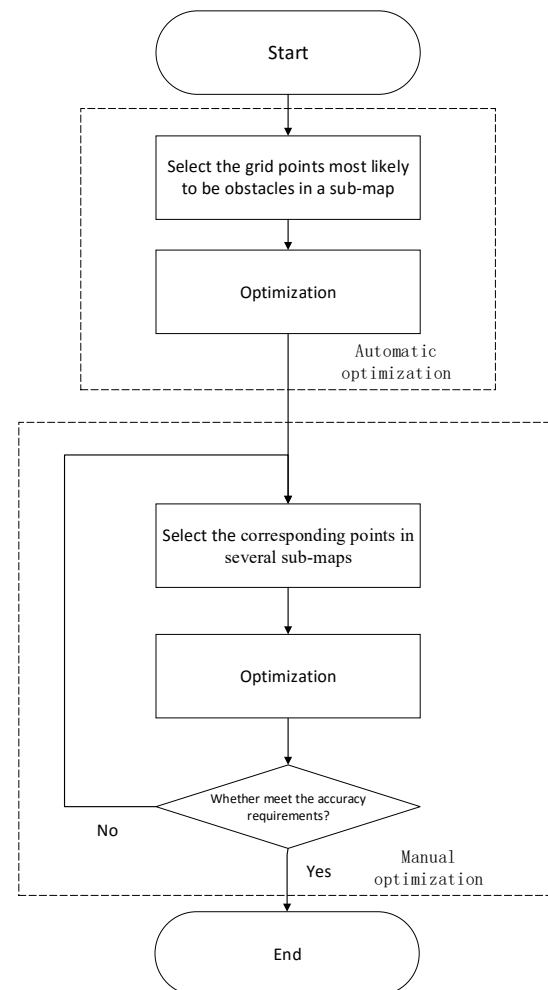


Figure 3. Optimization process

2.2 Dynamic Indexing Map

Another problem of large-scale map SLAM algorithm is the problem of map indexing and memory consumption. In order to quickly retrieve the map grid value, we want to remember the values of each grid in map. But before the implementation of the algorithm, we do not know how large the map may be. In order to ensure that our map structure can be adapted to all the different sizes of map, a dynamic index map is designed. In this map structure, we divide the grid map into larger grids, and each larger grid represents a map block. In the initialization stage, there is only one map block in the map structure. With the implementation of the SLAM algorithm, the map data is beyond the existing map blocks, we create a new map block in the corresponding location of the new map data and record the grid number of the map block. When we need to retrieve a grid value, first calculate the map block of this grid, and then retrieve the grid value in this map block. In this way, we can dynamically allocate memory according to the size of the actual map, and can quickly retrieve the value of the grid.

3. EXPERIMENTAL RESULT

3.1 Platform Description

The platform used is entirely independent research and development of our company. This platform is a set of equipment like a cart. There is a powerful computer engineering, two mutually perpendicular laser sensors and a high resolution panoramic camera, e.g. Figure 4



Figure 4. Indoor mobile measuring vehicle

This equipment is power by the lithium batteries(12 V) of 36 Ah, The fastest speed of 3 m/s implementation. The laser sensors used are two Hokuyo URG-30LX. It is connected to a USB port.

3.2 Result and Discussion

We compared the results of using a single map and the results of our sub-maps framework.

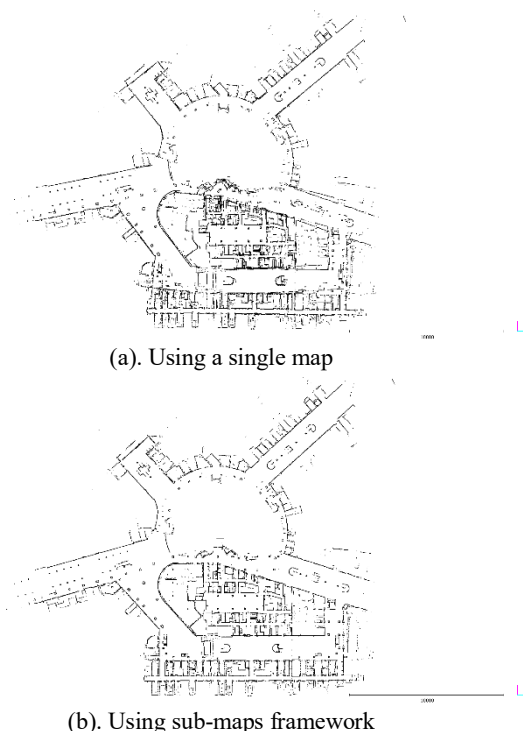


Figure 5. (a). Using a single map (b). Using sub-maps framework.

We tested the office buildings, shopping malls, pedestrian street and parking lots, these scenes represent the different size and complexity environment,

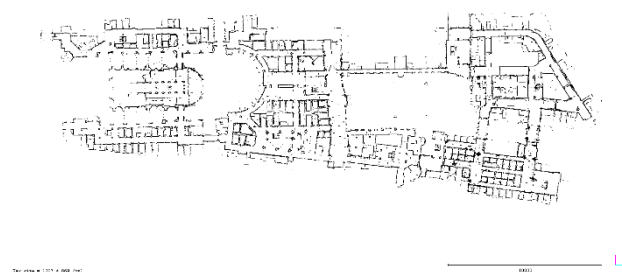


Figure 6. The scale represents 100 meters

4. CONCLUSION

This article, we show a framework of sub-maps. In this framework, we skilfully combine the state estimation method and the graph theory method with the sub-maps and the virtual landmark. We elaborate in detail the calculation method of the sub map, and the selection principle of the landmarks, and how to optimize the map through the relationship between the landmarks and the locus point.

Finally, through the experiment proved that SLAM algorithm based on our sub-maps frameworks in dealing with large scale environment is more effective than ordinary single map particle filter SLAM algorithm, and a dynamic index map structure is designed.to solve the contradiction between the memory and the retrieval speed.

REFERENCES

- Davison A J, Reid I D, Molton N D, et al. MonoSLAM: Real-time single camera SLAM. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on, 2007, 29(6): 1052-1067.
- Bailey T, Nieto J, Guivant J, et al. Consistency of the EKF-SLAM algorithm. *Intelligent Robots and Systems*, 2006 IEEE/RSJ International Conference on. IEEE, 2006: 3562-3568.
- Eliazar A, Parr R. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. *IJCAI*. 2003, 3: 1135-1142.
- Montemerlo M, Thrun S, Koller D, et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*. 2002: 593-598.
- Roller D, Montemerlo M, Thrun S, et al. Fastslam 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *Proceedings of the International Joint Conference on Artificial Intelligence*. 2003.
- Montemerlo M, Thrun S. Simultaneous localization and mapping with unknown data association using FastSLAM. *Robotics and Automation*, 2003. *Proceedings. ICRA'03. IEEE International Conference on. IEEE*, 2003, 2: 1985-1991.
- Grisetti G, Stachniss C, Burgard W. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. *Robotics and Automation*, 2005. *ICRA 2005. Proceedings of the 2005 IEEE International Conference on. IEEE*, 2005: 2432-2437.
- Kim C, Sakthivel R, Chung W K. Unscented FastSLAM: a robust and efficient solution to the SLAM problem. *Robotics*, IEEE Transactions on, 2008, 24(4): 808-820.
- Steux B, Hamzaoui O E. tinySLAM: A SLAM algorithm in less than 200 lines C-language program. *Control Automation Robotics & Vision (ICARCV)*, 2010 11th International Conference on. IEEE, 2010: 1975-1979.
- El Hamzaoui O, Steux B. SLAM algorithm with parallel localization loops: TinySLAM 1.1. *Automation and Logistics (ICAL)*, 2011 IEEE International Conference on. IEEE, 2011: 137-142.
- Grisetti G, Strasdat H, Konolige K, et al. g2o: A general framework for graph optimization. *IEEE International Conference on Robotics and Automation*. 2011.
- Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM. *Computer Vision—ECCV 2014*. Springer International Publishing, 2014: 834-849.
- Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: a versatile and accurate monocular SLAM system. *Robotics*, IEEE Transactions on, 2015, 31(5): 1147-1163.