TraMap: SLAM-Based trajectory matching and map generation for emergency scenarios

Yuqing Sun[1], Lei Wang[1,2,3], Sunhaoran Jin[1], Xinyang Liu[1], Bingxian Lu[1], and Zhenquan Qin[1]

[1] School of Software, Dalian University of Technology, Dalian 116600, China
[2] Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116600, China
[3] Peng Cheng Laboratory, Shenzhen 518000, China

Abstract. At present, in emergency research and rescue scenarios, it is often necessary to deploy base stations or nodes in advance, which will greatly increase the workload. This paper proposes a Simultaneous Localization and Mapping (SLAM) method called "TraMap", which is no need to deploy any nodes and additional power supply, and the rescuer only needs to carry an intelligent device to locate, and greatly reduces the deployment time and cost consumption.

Keywords: Indoor localization · Location based services · SLAM · Trajectory processing · Feature point correction

1 Introduction

The increasingly complex building structure adds a lot of difficulties to the research and rescue work in the emergency scenarios[1, 2]. In the complex environment, the commander cannot judge the location of the rescuer, which will lead to missing the best rescue opportunity. Therefore, it is of great significance to design a convenient and fast rescuer positioning system to improve rescue efficiency and ensure the safety of the rescuer.

At present, the main indoor positioning technologies, such as Infrared technology[3], Ultra-WideBand (UWB) technology[4], Wi-Fi fingerprint positioning technology[5, 6], Computer Vision positioning technology[7], etc., need to collect environmental information in advance, or deploy nodes in advance and need continuous power supply, which has high requirements for the external environment and is difficult to maintain.

We aim at the commander does not need to deploy additional nodes and completely power off to get the route[8] and current location of the area where the rescuers are searching and when the commander is unable to receive the position of the rescuer, the current position can be inferred according to the historical trajectory.

Based on SLAM technology[9, 10], we propose a method of trajectory matching and mapping, which is called "TraMap". It will process the trajectory information of the rescuer, build the trajectory map and emergency map for matching, calibrate and optimize the walls and corners as the common features of the

trajectory map and emergency map, and effectively determine the location of the rescuer.

Compared with single point positioning, the advantage of trajectory matching map is that even if part of the trajectory is disconnected, we can complete it through the trend, and we can easily know the historical path, so that we can predict when the signal is disconnected.

The contributions of this paper are as follows:

- Propose a new trajectory segmentation strategy, which can segment trajectories effectively and carry out simultaneous localization and mapping.
- Extract the feature of points and optimize the calibration.
- Demonstrate how to match the TraMap to the emergency map.

In Section 2, we introduce how to construct a TraMap, segment it and extract feature points. In Section 3, we match the TraMap with the emergency map and correct the feature point. The method is tested and validated in Section 4. Section 5 will introduce the related work in detail.

2 Generate a TraMap

When a rescuer enters, his action track is obtained by a series of processing such as acquisition and segmentation, and feature points are marked.

2.1 Obtain rough motion trajectory

With the rapid development of intelligent devices, there are more and more kinds of sensors on mobile phones. Since Android 1.5, the system has built-in 8 kinds of sensors[11], such as acceleration sensor and gyroscope and etc. Based on this background, inertial navigation system has also been established and developed rapidly. Pedestrian Dead Reckoning (PDR) is mostly used in indoor positioning Inertial Navigation System (INS)[12]. It uses mobile phone sensors to obtain pedestrian movement data as data set to calculate the user trajectory, so as to construct the user trajectory. The PDR positioning adopts the station center coordinate system. When the pedestrian walks in the i-th step, the position under the station center coordinate system satisfies equation (1):

$$\begin{cases} E_k = E_0 + \sum_{n=1}^{k} d_n sin(\varphi_n) \\ N_k = N_0 + \sum_{n=1}^{k} d_n cos(\varphi_n) \end{cases} \tag{1}$$

We take the entrance as the initial point $S_0(E_0, N_0)$, and generate a rough trajectory through the above algorithm. Figure 1. shows the trajectory construction diagram.

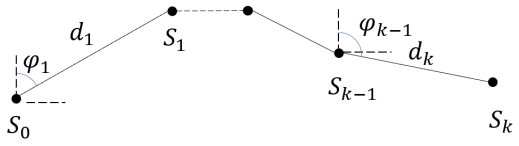


Fig. 1. Pedestrian trajectory diagram

2.2 Segment motion trajectory

After obtaining the rough trajectory, we need to obtain the key points of the trajectory, that is, feature points, which are used to match with the marker points of the emergency map. In this paper, we propose a new method to segment the trajectory into line segments, the start point and the end point of each line segment are the feature point.

The segment steps are as follows:

1) Determine the initial range: Select the farthest two points p_a and p_b from the multi-point trajectory data, and the distance between the two points is d;
2) Set double thresholds: Set two thresholds as $t_a = d/n, t_b = (n-1)d/n$, the range of n is $1.5 \sim 10$;
3) The first search: Select any point of p_a and p_b as the starting point and the other point as the end point, calculate the distance from the starting point point by point, and take the first point whose distance from the starting point is greater than t_b as the first middle point p_c;
4) The second search: Select p_c as the current starting point, take the starting point in step 3 as the current end point, calculate the distance from the current starting point to the current end point point by point, and take the first point whose distance from the current end point is less than t_a as p_d;
5) Determine the trajectory point: In the path between p_c and p_d, take the point farthest from the starting point in step 3 as p_e;
6) Obtain the splitting trajectory: Take p_e obtained in step 5 as the path point of the splitting trajectory, and then take p_e as p_a or p_b in step 1, and repeat the above steps until point p_e in step 5 coincides with point p_c or p_d.

The distance between the two points is the Euclidean distance of the trajector vector. In step 2, a status value is also set, and the initial value of status is 0; When step 3 is completed, the status is set to 1; When step 4 is completed, status is set to 0.

The pseudo code of the algorithm is shown in Algorithm 1.

Algorithm 1 Trajectory Segmentation

Input: The set of point, $P = \{p_1, p_2, p_3, \dots\}$; The thresholds t_a, t_b.
Output: The set of line segment $L = \{L_1, L_2, L_3, \dots\}$;the endpoint of line segment.

1: $p_{a,b}$ = SelectTwoPoints(P);
2: Set starting point and end point
3: **for** each $p \in P$ **do**
4: Calculate distance from starting point
5: Set the first point whose distance from the starting point is greater than t_b as P_c
6: **end for**
7: Set P_c as the starting point and set end point as the starting point in Line 2
8: **for** each $p \in P$ **do**
9: Calculate distance from starting point to end point
10: Set the first point whose distance from the starting point is less than t_a as P_d
11: **end for**
12: **for all** *line* such that $line \in C$ *to* D **do**
13: Set P_e as the point farthest from the starting point in Line 4
14: **end for**;
15: **while** $E = C$ *or* D **do**
16: Set P_e as P_a or P_b in Line 1
17: **end while**

3 Match TraMap with emergency map

Generally, the emergency map provides a small amount of information, such as fire exits, fire hydrants, toilets, even the color. Because these information will affect the matching, we need to preprocess the emergency map.

3.1 Emergency map preprocession

In order to maintain the contour invariance, the physical structure diagram shown in Figure 2. (a) is taken as an example to binarize the region boundary and connected region in the image to obtain the binary image shown in Figure 2. (b). Then, color reversal is performed on Figure 2. (b) to obtain Figure 2. (c), which represents the area image where the user can walk, and Figure 2. (d) shows the image with feature points marked after removing the interference information.

3.2 Matching based on *SAD* algorithm

This paper uses Sum of Absolute Differences (SAD) algorithm for map matching[13], SAD algorithm is a common method in pattern recognition. It traverses the whole search graph, find the most similar subgraph as the final matching result. The idea of the algorithm is simple, with high matching accuracy, widely used in image matching.

The similarity measure formula of SAD algorithm is as follows:

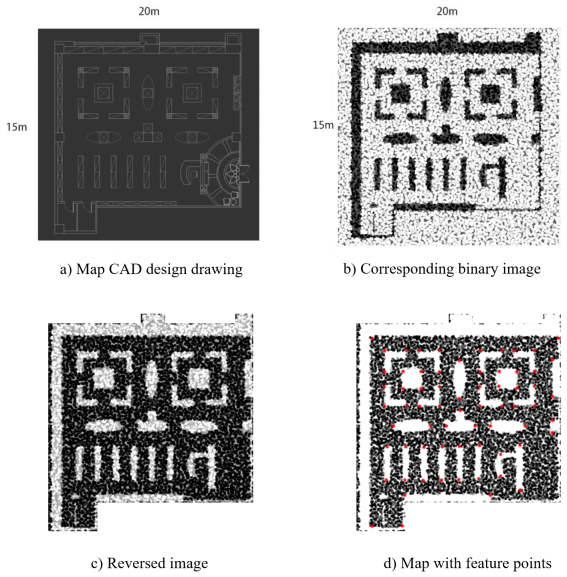


Fig. 2. Emergency map with prepocession

$$D(i,j) = \sum_{s=1}^{M}\sum_{t=1}^{N} |S(i+s-1, j+t-1) - T(s,t)| \tag{2}$$

where $1 \le i \le m-M+1, 1 \le j \le n-N+1$. M and N represents the length and width of the TraMap; m and n represents the length and width of the emergency map; (i,j) is the search coordinate; and $D(i,j)$ is the average absolute difference. The smaller its value is, the more similar it is. Therefore, we only need to find the smallest $D(i,j)$ to determine the location of the matching subgraph. Algorithm 2 shows the pseudo code of SAD algorithm.

3.3 Feature points correction

All points extracted from the emergency map (called marker nodes) are feature points in TraMap. The observed measurement values between the marker nodes and the feature points where they are detected are the observed edges. In order to match the emergency map on the TraMap, the TraMap must be rigid. Therefore, the standard deviation of the covariance of the observed edge is fixed at $\sqrt{0.5}$ m on the x and y axes, and only a small movement of the marker angle around its respective posture node is allowed. Use the known equivalent points in the two maps to obtain the conversion between the two maps, which is used to initialize the position of the marker node. The initial length of the marker edge is calculated based on the distance between the marker nodes.

Algorithm 2 SAD Algorithm

Input: Trajectory map;Emergency map.
Output: The area of emergency map where trajectory lies in.

```
[R, B, G] = Size(Emergency map);
if G == 3 then
    EmergencyMap=rgb2gray(Emergency map);
end if
[r, b, g] = Size(Trajectory map);
if g == 3 then
    TraMap=rgb2gray(Trajectory map);
end if
N = b;
M = R;
dst=zeros(M − N, M − N);
for i = 1 : M − N do
    for j = 1 : M − N do
        temp=src(i : i + N − 1, j : j + N − 1);
        dst(i, j)=dst(i, j)+sum(sum(abs(temp-mask)));
    end for
end for
```

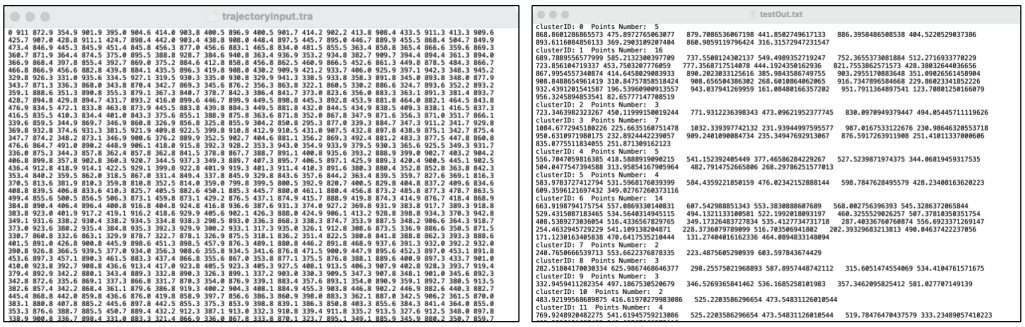
In order to associate each marker point extracted from the emergency graph with the potential corresponding relationship in the prior graph, the edge between each marker node and each prior node is added to TraMap, if the distance between them is below a certain threshold. These link edges represent zero transformation, and the standard deviation is set to$\sqrt{0.5}$ m on the x and y axes to account for possible noise. In fact, the corners of the links are allowed to move evenly with each other, but they cannot move away from each other without increasing cost.

4 Experiments and results

We chose the indoor environment of 20m × 15m as shown in Figure 2. (a) for the sample collection. There are many obstructions in the room, which can better simulate most of the indoor environment.

We collected several trajectory points through the PDR algorithm, connected them to form a trajectory, and then divided them through Algorithm 1 to get the trajectory cluster as shown in Figure 3. The segmented trajectory map is drawn through 15 times of MATLAB simulation. Figure 4. shows that we use SAD algorithm to match TraMap and emergency map. We can see that PDR algorithm can only roughly match emergency map because of the influence of sensor accuracy and system complexity.

Then we add feature points correction, Figure 5. shows the positioning results after feature correction, and the performance is better than the simple trajectory template matching.



a) Trajectory fingerprint points collected b) Clustering point set

c) Trajectory formed by the set of generated points

Fig. 3. Trajectory cluser results

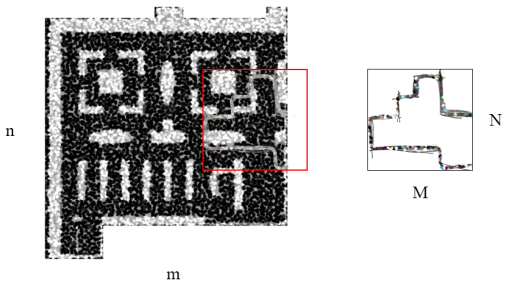


Fig. 4. Matching result with SAD algorithm

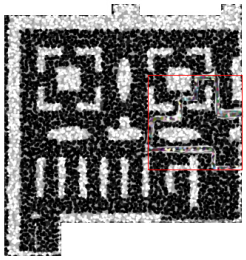


Fig. 5. Matching with feature points correction

We show the time cost of pixel template matching and the relationship between target positioning accuracy and pixel distance in Figure 6. It can be seen from the figure that when the pixel distance is less than 0.3 m, the time cost increases exponentially with the decrease of the pixel distance. In addition, when the pixel distance is less than 0.5 m, the average positioning error tends to be stable and close to the minimum error. Therefore, in order to ensure lower time cost and higher positioning accuracy, the binary image with pixel number = 120×100 and pixel distance = 0.5 m is selected for pixel template matching.

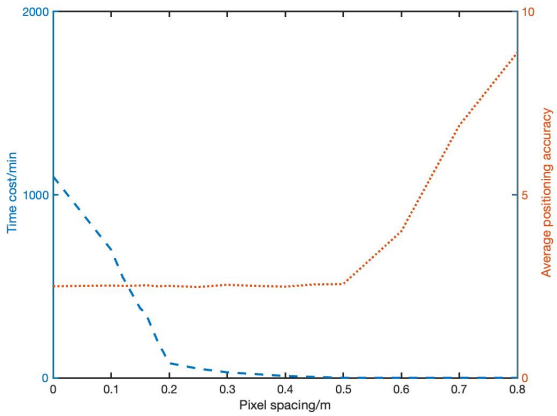


Fig. 6. Relationship of pixel, time cost and accuracy

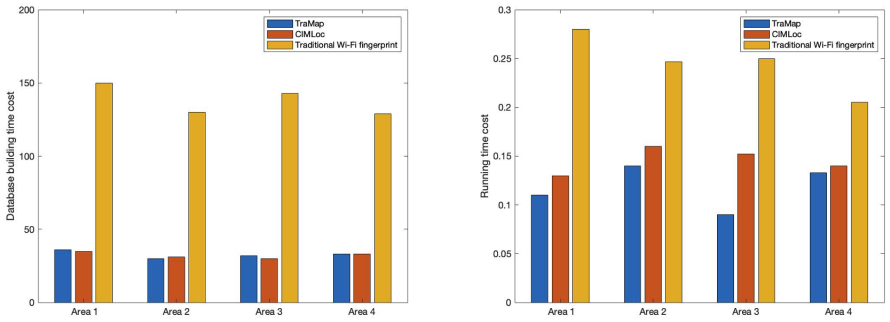
As shown in Figure 7., we tested TraMap, classical Wi-Fi fingerprint localization algorithm and CIMLoc[14] in the same indoor environment (as shown in Figure 4.). The test results show that there are a lot of drift in the location results of fingerprint location algorithm, and the location points can not be connected into trajectorys; both TraMap and CIMLoc algorithms can provide more

accurate positioning data, but TraMap's processing of feature points is more accurate than CIMLoc.

Fig. 7. Test of three methods in the same environment

In order to compare the test results better, we divide the test area into four parts in the way as shown in Figure 8. and Figure 9. shows the time complexity of three algorithms in different areas, which is reflected in the relationship between database building time and runnning time. It can be seen that TraMap has low time complexity and can be used as a positioning method in emergency scenarois.

Fig. 8. Map after division



a) Database building time

b) Running time

Fig. 9. Comparison of three methods

5 Related work

This section relates our approach to the state of the art and further details how we extend on it.Trajectories play an important role in many fields. In dealing with trajectories, many teams have done a lot of work, including trajectory segmentation and clustering. Alexandros and Nikos[15] inferred the turning limit of OpenStreetMap data by mining the matching trajectory of historical map from the existing fleet management services. Sotiris brakatsoulas[16] provided an algorithm to match the trajectory with the map, effectively exchanging the accuracy with the calculation speed. Hong Hu and XueMei Li[17] proposed an indoor pedestrian dead reckoning algorithm based on RSSI ranging and positioning, and uses extended Kalman filter to realize the fusion of positioning information. Most of the current trajectory clustering algorithms are based on the complete trajectory, which leads to low efficiency. To solve this problem, a trajectory clustering algorithm based on structural similarity is proposed by G Yuan and SX Xia[18]. Their research methods provide a great train of thought for this paper.

With the increasing demand of indoor positioning, the research of SLAM is gradually enriched[19]. HF Durrantwhyte[20] proposed a probabilistic method is proposed to combine trajectory with slam, but it needs to be based on continuous appearance.

In addition, Image Matching is also developing rapidly. In order to match TraMap, we have done a lot of research and investigation[21–23]. Filev and Hadjiiski[24] compared these methods and give us the advantage and disadvantage of them. Finally, we choose SAD algorithm, because the idea is simple, easy to understand, simple operation process, high matching accuracy.[25].

6 Conclusion

We developed a SLAM-based trajectory processing and mapping method named "TraMap" to obtain information from a rough trajectory. We also propose to correct it by matching corner points in emergency map with feature points in trajectory. The performance of the trajectory map after correction is better than before and other methods, which can better match the emergency map, and the commander can follow the trajectory of the rescuer. "TraMap" provides a better solution for emergency rescue.

References

1. S. Zorn, R. Rose, A. Goetz, and R. Weigel. A novel technique for mobile phone localization for search and rescue applications. In *International Conference on Indoor Positioning & Indoor Navigation*, 2010.
2. L. Wang, D. Zhao, T. Ni, and S. Liu. Extraction of preview elevation information based on terrain mapping and trajectory prediction in real-time. *IEEE Access*, PP(99):1–1, 2020.
3. D. Hauschildt and N. Kirchhof. Advances in thermal infrared localization: Challenges and solutions. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pages 1–8, 2010.
4. L. Taponecco, A. A. D'Amico, and U. Mengali. Joint toa and aoa estimation for uwb localization applications. *IEEE Transactions on Wireless Communications*, 10(7):2207–2217, 2011.
5. L. Hong, L. Sun, H. Zhu, L. Xiang, and X. Cheng. Achieving privacy preservation in wifi fingerprint-based localization. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 2337–2345, 2014.
6. Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. Avoiding multipath to revive inbuilding wifi localization. In *Proceeding of the International Conference on Mobile Systems*, page 249–262, 2013.
7. V. S. Kalogeiton, K. Ioannidis, G. Ch. Sirakoulis, and E. B. Kosmatopoulos. Real-time active slam and obstacle avoidance for an autonomous robot based on stereo vision. *Cybernetics and Systems*, 50(1-4):239–260, 2019.
8. S. P. Rana, M. Dey, H. U. Siddiqui, G. Tiberi, and S. Dudley. Uwb localization employing supervised learning method. In *IEEE International Conference on Ubiquitous Wireless Broadband ICUWB'2017*, pages 1–5, 2017.
9. Mwmg Dissanayake, P. Newman, S. Clark, H. F. Durrantwhyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Trans Ra*, 17(3):229–241, 2013.
10. G. Grisetti, R. Ku?Mmerle, C. Stachniss, and W. Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2011.
11. M. Shoaib, S. Bosch, O. Incel, H. Scholten, and P. Havinga. Fusion of smartphone motion sensors for physical activity recognition. *Sensors*, 14(6):10146–10176, 2014.
12. L. Huang, X. Gan, B. Yu, H. Zhang, S. Li, J. Cheng, X. Liang, and B. Wang. An innovative fingerprint location algorithm for indoor positioning based on array pseudolite. *Sensors (Basel, Switzerland)*, 19(20), 2019.
13. Nourain Nadir, Samir Brahim Belhaouari, Josefina Samir, and Janier. Fast template matching method based on optimized metrics for face localization. pages 30–34, 09 2021.

14. X. Zhang, Y. Jin, H. X. Tan, and W. S. Soh. Cimloc: A crowdsourcing indoor digital map construction system for localization. In *IEEE Ninth International Conference on Intelligent Sensors*, 2014.
15. Vassiliou, Yannis, Grivas, Nikos, Efentakis, Alexandros, Pfoser, and Dieter. Crowdsourcing turning-restrictions from map-matched trajectories. *Information systems*, 64(Mar.):221–236, 2017.
16. Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas, and Carola Wenk. On map-matching vehicle tracking data. In *Proceedings of the 31st International Conference on Very Large Data Bases*, page 853–864, 2005.
17. H. Hong, X. M. Li, Kang D., and Lu C. J. Indoor pdr algorithm based on rssi ranging positioning. *Computer and Modernization*, 2016.
18. Guang-ya, ZHANG, Jing, and ZHANG. Trajectory clustering based on trajectory structure and longest common subsequence. 2018.
19. H. F. Durrantwhyte and T. Bailey. Simultaneous localization and mapping (slam): part ii. *IEEE Robotics & Amp Amp Automation Magazine*, 13(2):99 – 110, 2006.
20. W. P. Maddern, M. Milford, and G. Wyeth. Continuous appearance-based trajectory slam. In *IEEE International Conference on Robotics & Automation*, 2011.
21. H. K. Yan, C. S. Wan, Y. L. Chan, W. L. Hui, W. H. Wong, K. M. Cheng, and Y. Huo. Method and apparatus for coding mode selection, 2013.
22. Method and apparatus for estimating distances in a region, 2002.
23. Richard Ian Kitney and N. Smith. Image segmentation method.
24. Comparison of similarity measures for the task of template matching of masses on serial mammograms. *Medical Physics*, 32(2), 2005.
25. F. Alsaade and Y. M. Fou Da. Template matching based on sad and pyramid. *International Journal of Computer Science & Information Security*, 10(4):17–20, 2012.