Motion Trajectory Sequence-Based Map Matching Assisted Indoor Autonomous Mobile Robot Positioning

Jianzhong Zhang[2] (✉), Wenping Yu[1], Jingdong Xu[2], and Yuwei Xu[2]

College of Computer and Control Engineering, Nankai University, Tianjin, China
[1]`{yuwenping}@mail.nankai.edu.cn`
[2]`{zhangjz,xujd,xuyw}@nankai.edu.cn`

Abstract. Position information is one of basic elements for context awareness of autonomous mobile robots. This paper studies the positioning algorithm of autonomous mobile robots suitable for search and rescue in dark building corridors and underground mine tunnels when an emergency occurs, and proposes a novel map matching aided positioning algorithm based on Hidden Markov Model. This algorithm does not rely on a camera, and only uses the inertial sensors installed in mobile robot and the indoor map to realize the fusion of dead reckoning and map matching. Firstly, it recognizes the position-related motion pattern during the motion of wheeled mobile robot, and then the motion trajectory is divided into a sub-trajectory sequence. By matching the sub-trajectory sequence with the indoor map, the proposed algorithm achieves accurate tracking and positioning of mobile robots. In order to verify the effectiveness of our proposed algorithm, this paper adopts four-wheel differentially driven robot to conduct experimental analysis in an actual indoor scenario. The experimental results show that compared with the traditional dead reckoning technology, this algorithm can distinctly reduce the average positioning error of mobile robot, and it is robust to heading angle and acceleration noises within a certain error range.

Keywords: Mobile Robot · Indoor Positioning · Hidden Markov Model · Motion Pattern Detection.

1 Introduction

With the advancement of artificial intelligence technology, network technology and sensor technology, the research and application of autonomous mobile robots have made remarkable progress in recent years. Indoor autonomous mobile robots are increasingly integrated into people's daily lives [1]. Autonomous mobile robots can be used not only in large-scale applications in modern intelligent warehouses, home services and many other aspects, but also in detection and search and rescue in corridors of complex buildings that are outages due to fire accidents, subway passages, and underground mines. . Therefore, the research of

indoor autonomous mobile robot technology has gradually become a hot topic, and many domestic research institutes such as Tsinghua University, Harbin Institute of Technology, Nankai University, South China University of Technology are committed to the research and development of indoor autonomous mobile robots [2-6]. The autonomous positioning of the mobile robot is a process in which the robot autonomously determines its position in the working environment, and is one of the most basic problems in improving the autonomous capabilities of the mobile robot.

In terms of outdoor positioning, the Global Positioning System (GPS) has become a widely used positioning technology for mobile robots. In terms of indoor positioning, due to the blocking and interference of GPS signals on the external walls of buildings and indoor complex electromagnetic environment, indoor mobile robots There is no universal solution to the positioning problem [7-8]. Currently, researchers have proposed a variety of positioning methods for indoor autonomous mobile robots, including navigation beacon-based positioning [9], computer vision positioning [10-11], dead reckoning positioning [12], map matching positioning [13-14] and synchronous positioning and map creation (SLAM) [15-16] and so on. Positioning based on navigation beacons relies on deployed feature signals in a range of environments to provide stable and accurate location information, but requires high deployment and maintenance costs. Dead reckoning positioning uses inertial sensors and encoders to provide relatively accurate phase positions over short distances, but there is cumulative error that gradually increases as the distance travels, and the robot's initial pose needs to be known in advance. Map matching positioning uses known indoor maps to construct topological maps, feature maps, and other abstract maps, and then the position of the mobile robot is obtained by matching the robot motion trajectory with the indoor path. From the map matching positioning realization principle, the real-time performance is relatively poor. Shortcomings. The SLAM technology has unique advantages in the face of unknown environments and can provide indoor planes or 3D maps while providing positioning [17]. However, this method requires mobile robots equipped with more complex sensor devices, such as infrared, ultrasonic radar, and RGBD vision systems. Therefore, it has higher implementation cost.

Complex corridors, subway station passages, or underground mines often have multiple passageways, similar to "mazes". In the event of a fire or other unexpected accident, the power supply is damaged, the communication infrastructure cannot be used, smoke, etc. It poses challenges for the positioning of autonomous mobile robots. Due to limitations in work environment or deployment conditions, it is difficult to establish visual or wireless navigation beacons in advance. Therefore, positioning technology based on navigation beacons is not suitable; the effects of high temperature and smoke on the indoor environment, cameras and other difficult to provide image information, computer The failure of visual positioning technology; the timeliness of SLAM technology can not meet the urgent need for time factors in the above scenarios. In response to these problems, the paper introduces a hidden Markov model (HMM) that does

not rely on a camera. It only uses inertial sensors (accelerometers, gyroscopes, and magnetometers) installed in autonomous mobile robots and known indoor maps to achieve dead reckoning. The combination of mapping technology and map matching technology.

The remainder of this paper is organized as follows: Section II reviews the related work in the literature. Section III gives an overview of the AiFiMatch system and its preprocessing components. Then, we detail the proposed HMM-based map matching algorithm in Section IV. Section V shows the experimental results and analysis of the AiFiMatch system. Finally, Section VI concludes this paper.

2 Robot Motion Model and Positioning Method

In the field of indoor autonomous mobile robot positioning, dead reckoning technology and map matching technology have a good complementarity. This paper proposes a map matching-assisted positioning method based on moving trajectory sequences of mobile robots to realize the fusion of the above two technologies. The positioning method uses stairs, corridor corners, etc. in the indoor environment as virtual landmarks. When the mobile robot passes through these landmark locations, the inertial sensor data will show a specific pattern. Therefore, in the text, the above landmark location is called a posture correlation. Position, when the robot's movement distance is short, the dead reckoning technology can give the real-time position of the robot. In the robot's movement distance is long, the robot's motion trajectory can be divided into multiple trajectory fragments according to the landmark position, continuous trajectory fragment The track sequence is composed of hidden Markov models. The above trajectory sequence can be matched to the corresponding path in the indoor map, and then the position estimation of the mobile robot is given. Further, when the robot's motion trajectory is long enough, the robot's When the initial position is unknown, the absolute position of the mobile robot can still be estimated.

This section first introduces the relevant modules and simplified motion models of the four-wheel differential-driven autonomous mobile robot used in subsequent experiments, and then derives the dead reckoning technology based on the motion model. Then the overall architecture of the fusion positioning method proposed in this paper is introduced. And the indoor map abstraction module, the final description of the mobile robot motion gesture recognition and use of the four-wheel differential drive robot designed in this paper to conduct a related pre-experiment. The trajectory matching technology based on Hidden Markov Models is the core module of the positioning method, which will be discussed in detail in the next section.

2.1 Robot Motion Model and Its Dead Reackoning Principle

In this paper, a four-wheel differentially driven wheeled mobile robot is used to study the positioning problem of autonomous mobile robots in indoor environment. The driving motor is a DC motor. The two driving motors on one side are

connected in reverse parallel and use the L298N motor driving module. Control the DC motor, and the mobile robot adopts the Raspberry Pi second-generation B-board (B.V1.2) as the main control chip. According to the driving mode of the mobile robot, the movement models of the two wheels on each side of the wheeled robot are the same. Therefore, the movement model of the mobile robot can be simplified to a left and right two-wheel differential driving mode. Figure 1 (a) shows the mobile robot. The simplified model, where (x, y) is the position coordinate of the mobile robot in the global coordinate system, Θ is the included angle between the mobile robot and the true north direction, ie the heading angle.
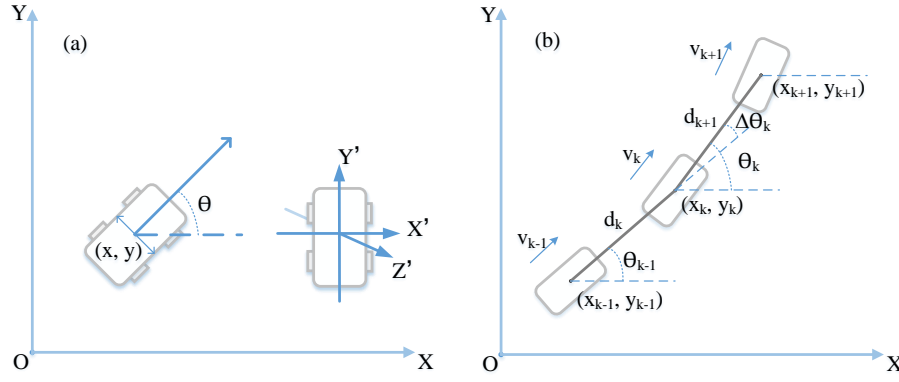

Fig. 1. Simplified motion model and its dead reckoning principle for four-wheel differentially driven robot. (a) Motion model and self coordinate system. (b) Dead reckoning in the global coordinate system.

Dead reckoning technology is widely used in the positioning of indoor autonomous mobile robots, and is particularly suitable for short-range positioning. The autonomous mobile robot used in this article has a built-in digital compass, three-axis accelerometer and gyroscope. The digital compass gives the initial attitude of the mobile robot. The accelerometer and the gyroscope can measure the movement acceleration and rotation angular velocity of the mobile robot, and then pass it. Integral operation can calculate the distance and heading change of the mobile robot, and then analyze the latest position and attitude of the mobile robot.

In order to determine the position and attitude of the mobile robot in the plane and establish the global right angle coordinate system OXY, it may be assumed that the initial position (x_0, y_0) is the origin of the coordinates and the initial attitude is the positive direction of the X-axis. Then the position and posture of the K time mobile robot can be expressed by vector. In the current moment, the instantaneous velocity of the mobile robot is represented by the angle between the moving direction of the mobile robot and the positive

direction of the X-axis, and the transverse ordinates of the mobile robot in the global coordinate system, as shown in Figure **??** (b). When the update cycle of sensor data is very small, if the update cycle of the sensor is about 5 milliseconds in this paper, in one cycle, the trajectory of the mobile robot can be approximated to a straight line, then the position of the K time mobile robot can be recursively obtained by the Equation 1.

$$\begin{pmatrix} v_k \\ \theta_k \\ x_k \\ y_k \end{pmatrix} = \begin{pmatrix} v_{k-1} \\ \theta_{k-1} \\ x_{k-1} \\ y_{k-1} \end{pmatrix} + \begin{pmatrix} 0.5(a_{k-1} + a_k)\Delta t \\ 0.5(\omega_{k-1} + \omega_k)\Delta t \\ d_k \cos \theta_{k-1} \\ d_k \sin \theta_{k-1} \end{pmatrix}, k \geq 1 \tag{1}$$

where the time interval from the $k-1$ time to the k moment, if the sensor fixed the data update period, also represents the update period, indicating the instantaneous acceleration in the direction of the moving robot at the time of k, which can be measured by the Y-axis component of the accelerometer, indicating the angular velocity of the heading angle at the k time of the moving robot. It can be measured by the Z-axis of the gyroscope, and indicates the movement distance of the mobile robot from $k-1$ to k. It can be drawn from the Equation 2:

$$d_k = \frac{v_{k-1} + v_k}{2}\Delta t \tag{2}$$
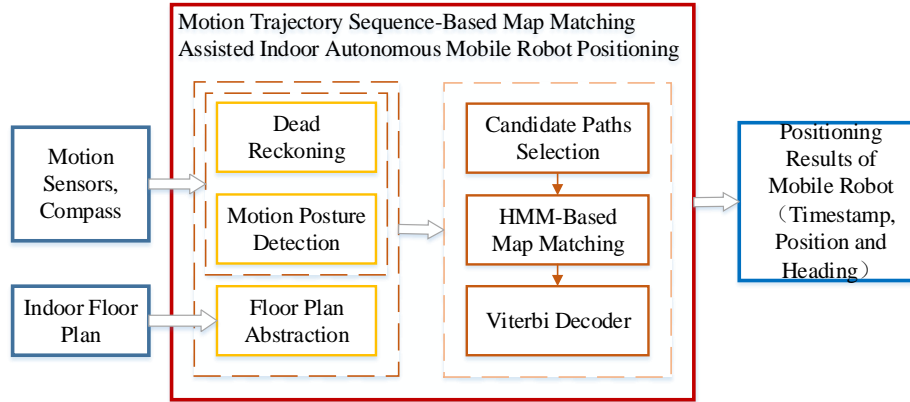
2.2 Architecture Overview

Fig. 2. System architecture of mobile robot positioning method.

The overall architecture of mobile robot localization method presented in this paper is shown in Figure 2. The sensor data and the indoor map are used as

the input of the positioning method, in which the sensor data is collected by the inertial sensors and magnetometers mounted on the mobile robot, and the indoor map is obtained by manually input or the indoor electronic map construction algorithm such as SLAM technology. The map Abstract module abstracts the indoor plane electronic map into the directed graph, the position estimation module and the motion attitude recognition module use the sensor data to give the relative position and position related action of the mobile robot respectively. The goal of map matching module is to match the motion trajectory of the mobile robot with the sequence of nodes in the directed graph, and then estimate the real time position of the mobile robot. First, the path is selected according to the direction angle estimation and the connection of the indoor path. Secondly, according to the latest alternative path and phase. The position change updates the related parameters in the hidden Markov model, and finally estimates the possibility of all the alternative paths through the online Viterbi decoder. When the matching algorithm is in the convergence stage, the most possible alternative path is the optimal estimation. The final output of the localization method proposed in this paper is the real-time location and heading of mobile robots.

2.3 Indoor Floor Plan Abstraction

The posture-relative positions divide the indoor path into path segments. With the path segment as a node, the action type that moves from one path segment to another path segment is a directed edge. Then the interior floor plan can be abstracted as a directed graph. Figure 3 shows an example of an interior plan and its corresponding directed graph. In this paper, a node is represented by a seven-tuple $(id, x_1, y_1, x_2, y_2, \varphi_1, \varphi_2)$, where $x_i, y_i, i = 1, 2$ represent the horizontal and vertical coordinates of the two endpoints of the path segment represent the heading angle when the mobile robot moves on the path segment and reaches the corresponding endpoint. A tuple represents a directed edge between nodes, which represents the identity of the starting node, represents the identity of the terminated node,,, and represents the path from the path fragment represented by the starting node to the termination node, respectively. The horizontal, vertical coordinates, and attitude types of the relevant positions of the motion pose to which the path segment passes.

2.4 Motion Posture Detection

This section presents a decision tree model for the motion gesture recognition of indoor autonomous mobile robots. This paper focuses on the positioning of mobile robots in indoor plane maps. Therefore, this decision tree model considers only the relevant gesture recognition of the movement of a mobile robot on a plane. Including static, straight, left/right turns and U-turns. The horizontal movement attitude of the indoor autonomous mobile robot is distinguished by different modes of the horizontal component of the accelerometer and the vertical component of the gyroscope built into the mobile robot, wherein the horizontal component of the accelerometer is the X-axis and Y in the local coordinate
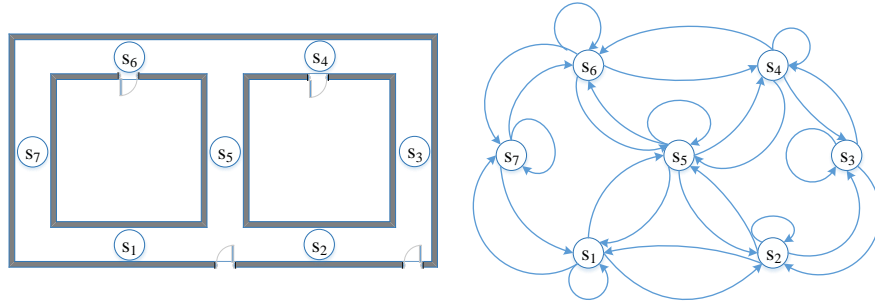
Fig. 3. Indoor floor plan abstraction example.

system of the robot. - The synthesis of the axis, and the vertical component of the gyroscope is the data of the Z-axis in the robot's local coordinate system, and the attitude of the mobile robot is determined by extracting the vertical component of the accelerometer. The method can be easily extended to three-dimensional indoor positioning scenes.
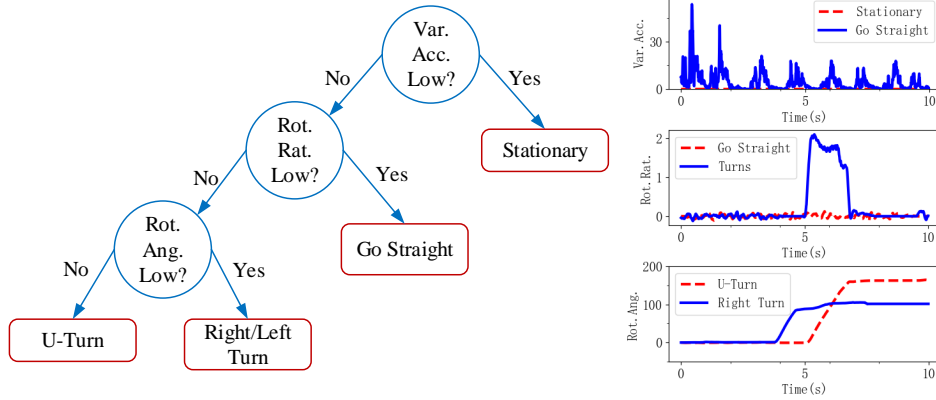

Fig. 4. Decision tree for motion posture detection.

Figure 4 shows a decision tree for the motion gesture recognition of an indoor autonomous mobile robot. The decision tree uses the built-in acceleration of the mobile robot and the signal characteristics of the gyroscope data to identify different motion gestures of the mobile robot. Considering that the linear velocity of the mobile robot is obtained from the integral of the acceleration level component in time, the instantaneous velocity has an accumulated error at a certain moment. Therefore, the uppermost layer of the decision tree uses the variance of the acceleration level component to judge the mobile robot's Motion or motion, not instantaneous speed; the second level of the decision tree uses

the magnitude of the rotational angular velocity measured on the Z-axis of the gyroscope to determine the straight line or turn of the mobile robot; finally, the third level of the decision tree passes through the turn process. The angle of rotation is integrated over time to obtain the rotation angle, and the rotation angle is used to distinguish whether the mobile robot is turning left/right or turning.

Table 1. Confusion Matrix of Motion Posture Detection

Motion Posture	Left Turn	Right Turn	U-turn	Go Straight
Left Turn	58	0	2	0
Right Turn	0	58	1	1
U-turn	0	0	60	0
Go Straight	0	0	0	60

In the real experimental environment, the four-wheeled differential-driving robot described in the previous paragraph was used to perform left-turning, right-turning, left-right turning, and straight-motion testing of different movement postures. Each sample was sampled for 3 seconds and each movement gesture The effective number of samples is 60. The classification results obtained by the above decision tree model are shown in Table 1.

3 HMM based Map Matching Algorithm

Based on the motion posture detection based on inertial sensor data, the motion trajectory of the mobile robot can be divided into trajectory segments by left and right turn and U-turn, and these trajectory segments form a trajectory sequence in the time dimension. This section gives a detailed description of Hidden Markov Models for trajectory sequence matching to indoor abstract maps.

3.1 Hidden Markov Model

A Hidden Markov Model is a time-series probability model that describes the state of a process using discrete random variables. A basic HMM can be represented as $\lambda = (S, V, A, B, \pi)$, where:

1) $S = \{s_1, s_2, s_3, \ldots, s_N\}$ is the set of possible states and $N = |S|$. In our case, each state represents an indoor road segment, that is, a node of the directed graph. Therefore, a state s is represented by the ordered tuple in the form of $(id, x_1, y_1, \varphi_1, x_2, y_2, \varphi_2)$, where id is the identification of road segment, x_1, y_1, φ_1, x_2, y_2 and φ_2 are different attributes of node of the directed graph, respectively. It should be noted that if the mobile robot can reach another path segment by going straight from one path segment, these two path segments can

be merged into a new path segment, ie a new hidden state, as shown in Fig. 3. The path segments s_4, s_6 can be combined into new path segments.

2) $V = \{v_1, v_2, v_3, \ldots, v_M\}$ is the set of observations from the model and $M = |V|$. In our case, an observable state represents the relative movement distance and heading measured by an inertial sensor mounted on a mobile robot and expressed in terms of $(dist, \varphi)$.

3) $A = \{a_{ij}\}$ is the state transition probability distribution, where $a_{ij} = p\{q_{t+1} = s_j | q_t = s_i\}, i, j \leq N$, where q_t denotes the state at time t. In other words, a_{ij} indicates the possibility of moving from one road segment to adjacent road segments.

4) $B = \{b_i(k)\}$ is the observation probability distribution in state i, where $b_i(k) = p\{z_t = v_k | q_t = s_i\}, 1 \leq i \leq N, 1 \leq k \leq M$ and z_t, q_t are the observation and state at time t, respectively. In other words, $b_i(k)$ indicates the possibility of a certain distance and direction measured by the inertial sensor after the mobile robot has passed a road segment.

5) $\pi = \{\pi_i\}$ is the initial state distribution, where $\pi_i = p\{q_1 = S_i\}, 1 \leq i \leq N$.

3.2 Transition Probability Distribution (A)

The transition probability distribution refers to the possibility of moving from a hidden state to the next hidden state. In this paper, the possibility of moving from one path segment to the adjacent path segment is indicated. The adjacent path segments are divided by posture-related positions. Each posture-related position has a corresponding motion posture. The higher the matching degree of the mobile robot's motion posture and position-related posture, the mobile robot passes through a path segment. The higher the probability that the gesture-related position moves to another path segment, and vice versa. Therefore, we use the degree of match between the motion pose and position-related poses of the mobile robot to represent the transition probabilities between adjacent path segments. Let the edge of the directed graph corresponding to the route segment move to the path segment be defined by section 2.3. The position-related pose corresponding to the edge is the motion pose of the mobile robot that has been identified at the given time t. The probability is shown in Equation 3, where it can be obtained from the motion pose confusion matrix in Section 2.4.

$$p(s_{j,t}|s_{i,t-1}) = p(s_j|s_i, Rob_{MA}(t)) = p(Rob_{MA}(t)|e^{ij}.MA) \qquad (3)$$

3.3 Observation Probability Distribution (B)

In this paper, an observable state consists of the relative displacement of the mobile robot and the heading angle, and the two are independent of each other. Therefore, the observable probability distribution can be defined as:

$$P(v_{k,t}|s_{j,t}) = P(\varphi(t)|s_{j,t}) \cdot P(dist(t)|s_{j,t}) \qquad (4)$$

where, $P(\varphi(t)|s_{j,t})$ represents the observable probability of the mobile robot's heading at time t, the condition of an implicit state, and $P(dist(t)|s_{j,t})$ denotes the observable probability determined by the relative displacement of the mobile robot.

The higher the degree of match between the heading of the mobile robot and the path segment, the greater the possibility that the mobile robot is located in the path segment. In the indoor environment, the error of the heading angle of the mobile robot not only comes from the accumulated error in the dead reckoning process, but also comes from the interference of various metal materials in the room with the magnetometer. In general, the error of the heading angle is relatively large. It is difficult to accurately model this error. Therefore, in this paper, Equation 5 is used to model the heading information in the observable state.

$$
\begin{aligned}
&P(\varphi(t)|s_{j,t}) \\
&= P\{\varphi(t)|s_j.\varphi_i, i = 1, 2\} = \begin{cases} 1, if |\varphi(t) - s_j.\varphi_i| < H_{TH}, i = 1, 2 \\ 0, others \end{cases}
\end{aligned}
\tag{5}
$$

where, H_{TH} is a constant threshold used to determine whether the heading angle of the mobile robot matches the direction of the path segment. In order to avoid that the correct path segment is excluded due to the large error of the heading angle, H_{TH} is set to 59 in this paper.

The relative displacement error of the mobile robot mainly comes from the accumulative error caused by the acceleration error in the dead reckoning process. Here we assume that the relative displacement of the mobile robot obeys the Gaussian distribution. On the one hand, intuitively speaking, the closer the relative displacement of the mobile robot is to the length of the path segment, the more likely the mobile robot is located in the path segment; on the other hand, the length of the path segment is much larger than the relative displacement of the mobile robot. All path fragments should have similar possibilities. Combining the above two conditions, this paper uses Equation 6 to model the relative displacement information in the observable state.

$$
\begin{aligned}
&P(dist(t)|s_{j,t}) \\
&= P\{dist(t)|s_j.dist\} = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma_d} e^{-4.5}, dist(t) + 3\sigma_d \le s_j.dist \\ \frac{1}{\sqrt{2\pi}\sigma_d} e^{-\frac{(dist(t) - s_j.dist)^2}{2\sigma_d^2}}, others \end{cases}
\end{aligned}
\tag{6}
$$

Where $s_j.dist$ is the length of the path segment, which can be derived from the two endpoints of the path segment, which is the standard deviation of the relative movement distance of the mobile robot at time t. In order to estimate the value, this paper first tests the change of the accelerometer's value when the mobile robot is at rest, and estimates the standard deviation of the acceleration based on the absolute median error (MAD) of the test data. The principle of the dead reckoning described in Section 2.1 is used. It can be inferred that there is a secondary relationship with [18].

$$\sigma_a = 1.4826 \times median(|\Delta a|) \tag{7}$$

3.4 Initial State Distribution

If the starting point of the mobile robot is known, then the path segment where the starting point is located is the initial state, and the probability is set to 1; if the starting point of the moving robot is unknown, all can be selected by formula 5 based on the initial heading information of the moving robot. For possible alternative path segments, the initial probability distribution is a uniform distribution over the candidate path segments. Correspondingly, the starting point of the mobile robot is assumed to be the average of the starting points of all candidate path segments. In this paper, the initial heading angle of a mobile robot can be obtained from a digital compass or set manually.

3.5 Optimal Motion Trajectory Estimation

Based on the above-defined Hidden Markov Model, this paper uses Viterbi algorithm to determine the optimal estimation of the moving trajectory of a mobile robot. For a given observable state sequence $(z_1, z_2, ..., z_k)$, the goal of the Viterbi algorithm is to find the most hidden State sequence $(q_1, q_2, ..., q_k)$. Fig. 5 briefly illustrates the decoding process of the Viterbi algorithm.
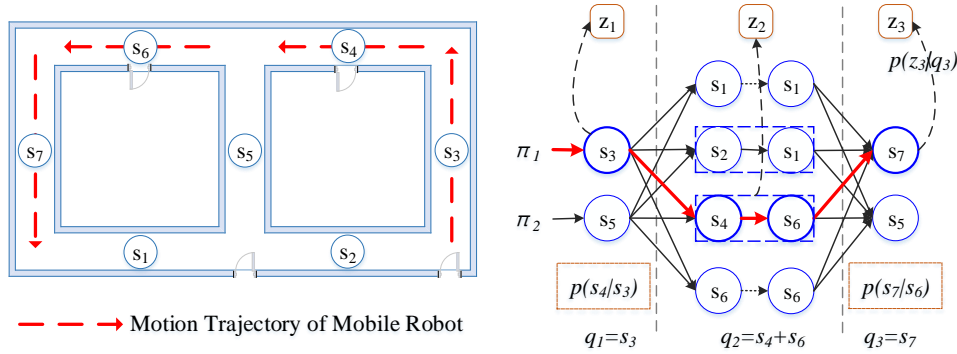

Fig. 5. Illustration of the proposed HMM model Viterbi decoding.

The Viterbi decoder is implemented by the dynamic programming method. First, a Viterbi variable is defined to represent the maximum probability that the Hidden Markov Model will reach the state along a path at time t and output:

$$\delta_t(i) = \max P\{q_1, q_2, \cdots, q_t = s_i, z_1, z_2, \cdots, z_t | \lambda\} \tag{8}$$

At time $t + 1$, the maximum probability reaching the hidden state s_j can be recursively derived from the Viterbi variable at time t by the following equation.

$$\delta_{t+1}(j) = [\max_i(\delta_t(i) \cdot P\{q_{t+1} = s_i | q_t = s_j\})] \cdot P\{z_{t+1} | q_{t+1}\}, 1 \leq t \leq k \qquad (9)$$

By recording the back-pointer, at time k, the most likely hidden state sequence, that is, the optimal estimation of the moving trajectory of the mobile robot can be obtained by the path backtracking method. Algorithm 1 summarizes the processing flow of the trajectory sequence map matching method presented in this paper.

AiFiMatch algorithm is summarized with the pseudocode in Algorithm 1.

Algorithm 1 AiFiMatch online map matching algorithm

Input: Sensor data up to current time t: $data_{1:t}$
Input: Abstract Indoor Floor Plan: fp
Input: Hidden State Sequence and Its Length up to last time $t - 1$: $S_k(t-1)$, k
Input: Array of Viterbi variable and backward pointer: $VtbArr_k(t-1)$
Output: Pedestrian's position estimation at current time t: $Ped_{loc}(t)$
Output: Optimal state sequence estimation at current time t: $Q(t)$
 1: $Ped_{AT}(t), at \leftarrow activity\_detect(data_{1:t})$
 2: **if** $Ped_{AT}(t) \mathrel{!=} None$ **then**
 3:     $st \leftarrow at$
 4:     // Calculate the transition probability
 5:     $S(t) \leftarrow [\,]$
 6:     **for** $s_i \in S_k(t-1)$ **do**
 7:         // Select next segments according to fp
 8:         $S^i(t) \leftarrow next(fp, s_i)$
 9:         // Extract segments according to Wi-Fi data
10:         $S^{i,wifi}(t) \leftarrow extract(data_{st:t}^{wifi}, S^i(t))$

---

4   Evaluation

We uses the wheel mobile robot introduced in Section 2 to complete the experimental analysis. The experimental environment is part of the fifth floor of the experimental building of the author. The experimental area is divided into east and west parts, and the east part is approximately 84.85 * 66.8 square meters. The west part is approximately 68.7 * 106.75 square meters, the connecting corridor length is 46.25 meters and the width is 2.4 meters. The overall layout is shown in Figure 6. In order to record the real position of the robot during the movement, this article divides the experimental area into squares of 0.8 * 0.8 square meters and marks. In the experiment process, another intelligent vehicle with camera function is used to move in parallel with the robot. Real-time recording of the location. In the experimental area, the robot moves along the two planned paths in Figure 6 and is denoted as T_1 and T_2, respectively. The length of T_1 is 181.9 meters, including 3 position related positions. The whole

```
11:      for s_j ∈ S^{i,wifi}(t) do
12:          tr[i][j] ← p(s_j|s_i, Ped_{AT}(t))
13:      end for
14:      S(t).extend(S^{i,wifi}(t))
15:   end for
16:   // Calculate the observation probability
17:   z_t ← dead_reckon(data_{st:t})
18:   for s_i ∈ S(t) do
19:      ob[i] ← p(z_t|s_i)
20:   end for
21:   VtbArr_{k+1}(t) ← viterbi(VtbArr_k(t−1), ob[ ], tr[ ][ ])
22:   k ← k + 1
23:   S_k(t) ← S(t)
24: else// Not a location-related activity
25:   ob ← [ ], S_k(t) ← S_k(t−1)
26:   // Update the observation probability
27:   z_t = dead_reckon(data_{st:t})
28:   for s_i ∈ S_k(t) do
29:      ob[i] ← p(z_t|s_i)
30:   end for
31:   VtbArr_k(t) ← viterbi(VtbArr_k(t−1), ob[ ])
32: end if
33: // Optimal State Sequence
34: Q(t) ← None
35: if check_status(VtbArr^k(t)) = Convergence then
36:   Q(t) ← viterbi_decode(VtbArr^k(t))
37: end if
38: // Update the location estimation according to Q(t)
39: Ped_{loc}(t) ← update_location(data_{st:t}, Q(t))
```

trajectory is divided into 3 trajectory-related trajectories divided into 4 trajectory sub-tracks; T_2 length is 180.5 meters, including 2 posture related positions, the whole trajectory is 2 The position-related poses are divided into three trajectories consisting of sub-tracks, and the robot is repeated 9 times for each path.

4.1 Influence of Heading Direction Errors

When the starting point is unknown, the convergence performance of the algorithm is closely related to the recognition results of position-related gestures. In general, after at least two consecutive position-recognition gestures need to be correctly recognized, the map matching algorithm based on the gesture recognition is likely to converge. , And correspondingly, in the process of position-related gesture recognition (such as left and right turning) facing indoor plane maps, the error of the heading angle has a greater impact on the attitude recognition results. Therefore, this section first analyzes the heading angle error for
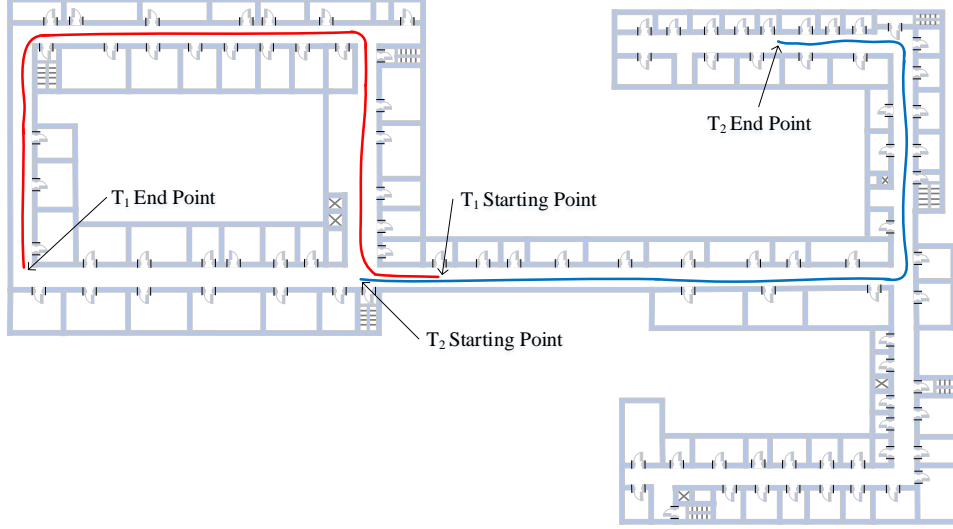
Fig. 6. Indoor floor plan of experimental environment and mobile robot trajectories.

the proposed text. The influence of the convergence performance of map matching algorithm. Here, we first define the correct rate of position-related gesture recognition by Equation 10.

$$a = b + c \tag{10}$$

Assume that the distribution of the heading angle estimation error is Gaussian and the average is 0. Based on the original data of the heading angle, a Gaussian distribution-based random number is added to simulate different degrees of error. Figure 7 shows the errors at different heading angles. The variation of the correct rate of position-dependent gesture recognition under the condition of the standard deviation of the distribution. It can be seen from Fig. 7 that the accuracy of position-dependent gesture recognition is stable under certain heading angle error conditions, but when the standard deviation of the error distribution reaches a certain level (T1 is 40 degrees and T2 is 30 degrees), The correct rate dropped rapidly.

4.2 Convergence Speed Analysis without Knowing the Starting Point

Under the condition that the starting point is unknown, after the mobile robot moves a certain distance, the algorithm can still converge and finally estimate the real-time position of the mobile robot. The distance before the convergence of the algorithm shows the convergence performance of the algorithm. In order to evaluate the convergence performance of the proposed map matching algorithm,
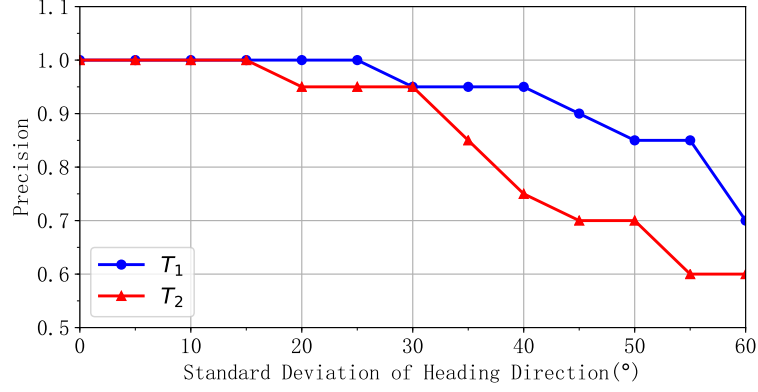
Fig. 7. Heading errors on the precision of mobile robot posture detection.

we compare the proposed map matching algorithm with the semimatch algorithm proposed in [18]. The semMatch algorithm has certain similarities with the algorithm presented in this paper. The hidden Markov model is also used to implement map matching operations. However, the details of the model are slightly different.
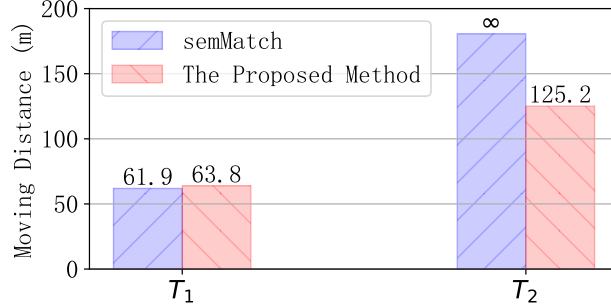

Fig. 8. Distance traveled before convergence for each trajectory.

Using the same decision tree model to identify the pose types in the trajectory of a moving robot, Figure 8 shows the convergence of the two algorithms. On the one hand, for the T1 path, both algorithms reach the convergence state after passing through two pose-related positions. However, the algorithm proposed in this paper needs to observe the subsequent path segment after identifying the corresponding pose type, so the convergence performance is improved. Slightly worse, when it reaches the convergence state, the mobile robot moves 1.9 meters more. On the other hand, for the T2 path, the algorithm proposed in this paper reaches the convergence state shortly after the recognition of the first position-

related pose, and the corresponding seamatch algorithm does not converge due to the symmetry of the interior structure. The infinity is denoted by the infinity large size ∞. The main reason is the difference in the model definition of the two map matching algorithms. The implicit state of the map matching algorithm proposed in this paper is the straight line segment in the indoor road network. In the T2 path, the mobile robot Firstly, a path segment with a long enough length is combined with the subsequent recognition of the first position-related pose to achieve the convergence of the algorithm.

4.3 Online Positioning Performance with Knowing the Starting Point

Under the condition that the starting point is known, the localization algorithm proposed in this paper does not need to pass the motion trajectory matching stage to converge. After convergence, the algorithm can track the moving trajectory of the mobile robot in real time. We use the Euler distance between the real physical position of the autonomous mobile robot and the position estimate given by the algorithm to analyze the real-time positioning performance. Figure 9 shows the positioning of autonomous mobile robots with increasing distances on both T1 and T2 paths. Error variation.
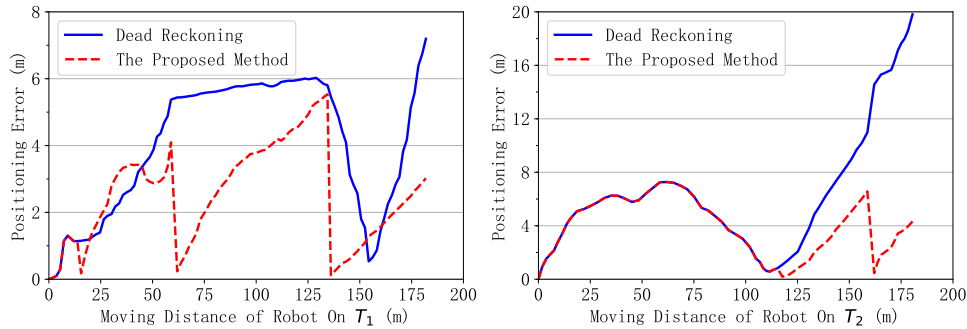

Fig. 9. Online positioning errors for each trajectory.

When the motion trajectory does not include posture related position, the map matching assisted positioning technology proposed in this paper is equivalent to the traditional dead reckoning technology, but after identifying the relevant position of the pose, the coordinates of the known pose related position can be used to move the coordinates. The real-time position of the robot is corrected. The real-time positioning results of the mobile robot on the two paths T1 and T2 shown in Fig. 9 are verified by the positioning results on both paths. For the T1 path, the average positioning error is 4.0. The meter falls to 2.49 meters, while for the T2 path, the average positioning error decreases from 6.58 meters to 3.39 meters. From the above trend, it can be deduced that in

the actual environment, the greater the density of posture-related locations, the more obvious the improvement of the traditional dead reckoning technology is.

4.4 Influence of Acceleration Errors

From the principle of dead reckoning technology described in section 2.1 of this paper, we know that the acceleration error has a quadratic relationship with the error of the moving distance of the mobile robot. Therefore, the acceleration error is one of the main sources of accumulated error in dead reckoning technology. This section simulates different acceleration error scenarios by adding Gaussian random generation numbers of different standard deviations to the original acceleration data, and then analyzes the influence of the acceleration error on the positioning algorithm proposed in this paper.
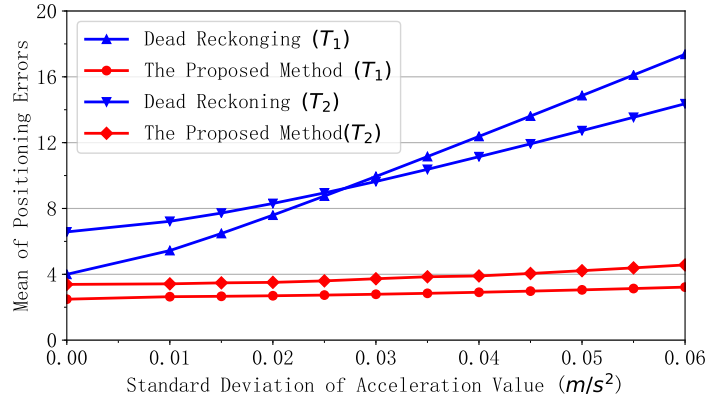

Fig. 10. Acceleration value errors on the mean of positioning errors for each trajectory.

Figure 10 shows the average positioning error of the mobile robots in the two paths T1 and T2 under different standard deviation values. It can be seen from the figure that, on the one hand, based on a slight increase in the acceleration error, the dead reckoning is based on dead reckoning. The positioning method of the technique rapidly increases the average positioning error, which means that the positioning performance of dead reckoning technology is very sensitive to the acceleration error; on the other hand, compared with the traditional dead reckoning technology, the same acceleration error changes. The positioning error of the positioning algorithm proposed in this paper changes more smoothly, which shows that the positioning algorithm has a certain degree of robustness to the acceleration error.

5   Conclusion

In order to solve the difficult problem of positioning of autonomous mobile robots in dark complex building corridors, subway station corridors, or underground mines after a sudden accident such as a fire, this paper proposes an indoor autonomous mobile robot tracking based on hidden Markov model for map matching assistance. And positioning methods. In the structured indoor environment, this method uses the recognition of position-related gestures to match the motion trajectory of the mobile robot to the indoor map. Compared with the traditional dead reckoning technology, the method can significantly reduce the influence of cumulative error on the positioning accuracy, and for certain The degree of heading angle and acceleration errors are robust. This method does not rely on cameras. It uses only accelerometers, gyroscopes, magnetometers, and known indoor maps installed in autonomous mobile robots to achieve fusion positioning of dead reckoning and map matching techniques, even when the starting point is unknown. In addition, the effective tracking and positioning of mobile robots can also be achieved, with features such as simple deployment, low manufacturing cost, and easy operation.

References

1. Woodman, O., Harle, R.: Pedestrian localisation for indoor environments. In: Proceedings of the 10th international conference on Ubiquitous computing, ACM (2008) 114–123
2. Bao, H., Wong, W.C.: An indoor dead-reckoning algorithm with map matching. In: Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International, IEEE (2013) 1534–1539
3. Rai, A., Chintalapudi, K.K., Padmanabhan, V.N., Sen, R.: Zee: Zero-effort crowd-sourcing for indoor localization. In: Proceedings of the 18th annual international conference on Mobile computing and networking, ACM (2012) 293–304
4. Xiao, Z., Wen, H., Markham, A., Trigoni, N.: Lightweight map matching for indoor localisation using conditional random fields. In: Proceedings of the 13th international symposium on Information processing in sensor networks, IEEE Press (2014) 131–142
5. Wang, H., Sen, S., Elgohary, A., Farid, M., Youssef, M., Choudhury, R.R.: No need to war-drive: Unsupervised indoor localization. In: Proceedings of the 10th international conference on Mobile systems, applications, and services, ACM (2012) 197–210
6. Zhou, B., Li, Q., Mao, Q., Tu, W., Zhang, X.: Activity sequence-based indoor pedestrian localization using smartphones. IEEE Transactions on Human-Machine Systems **45**(5) (2015) 562–574
7. Aly, H., Youssef, M.: semmatch: Road semantics-based accurate map matching for challenging positioning data. In: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM (2015) 5
8. Bloit, J., Rodet, X.: Short-time viterbi for online hmm decoding: Evaluation on a real-time phone recognition task. In: ICASSP, IEEE (2008) 2121–2124