2013 IEEE Intelligent Vehicles Symposium (IV)
June 23-26, 2013, Gold Coast, Australia

Lidar Scan Matching EKF-SLAM Using the Differential Model of Vehicle Motion

Daobin Wang, Huawei Liang, Tao Mei, Hui Zhu, Jing Fu, Xiang Tao

***Abstract*— Simultaneous localization and mapping is a mobile robot positioning themselves and creating the map of the environment at the same time, which is the core problem of the vehicle achieve the authentic intelligent. EKF-SLAM is a widely used SLAM algorithm based on the extended Kalman filter. The EKF-SLAM proposed in this paper based on the differential model of vehicle motion, which consider the vehicle trajectory as many small straight line segments. The algorithm effectively reduce the positioning error compared with the dead reckoning and has more simplified and generic model compared with the EKF-SLAM algorithm based on vehicle kinematics model. Meanwhile, it has a lower requirements on the hardware acquisition system. The algorithm is more robust than the traditional EKF-SLAM. So the algorithm will have a certain reference value on the SLAM research and provide a new way on the SLAM research based on the differential model of vehicle motion.**

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is mobile robots positioning themselves by the environment information and at the same time create the environmental map according to their position. It is characterized by the integration of environmental information and dead reckoning and can improve the performance of dead reckoning.

SLAM was first proposed by Smith, Self and Cheeseman. It's the core issue for the mobile robot to achieve the truly independent[1]. SLAM problem involves unknown and uncertain environment description and sensor noise, therefore it needs to use probabilistic methods to describe. The theory of uncertainty representation and probabilistic map provide a theoretical basis for in-depth research on the SLAM problem. SLAM algorithms include EKF-SLAM[2,3], CEKF-SLAM[4], UKF-SLAM[5], Rao-Blackwellized filter SLAM[6,7], Fast-SLAM[8], DP-SLAM[9,10], EIF-SLAM[11], TJTF-SLAM[12], Scan-matching SLAM[13] and so on. One of the most classic and widely used algorithm is EKF-SLAM. The algorithm fuses environmental information and dead reckoning through an extend Kalman filter.

However, these algorithms are based on the kinematic model of vehicle, which consider the vehicle trajectory as a continuous structure composed of many tiny arc segment. The arc curvature is determined by the rotation angle of the front wheels, and the arc length is determined by the vehicle speed and the time interval[2-4,14]. The kinematic model need to obtain the instantaneous rotation angle of the front wheels and the vehicle speed. In fact the measuring of these variables is difficult and is susceptible to the noise influence. In addition to this, the kinematic model of vehicle is not applicable in the biped robot and the crawler robot. Besides, these algorithms need to unify the kinematic model coordinate system and the observation coordinate system.

In response to these problems, we proposed a new EKF-SLAM algorithm in this paper, which is based on the differential model of vehicle motion, and just need to obtain the traveled distance in the sampling interval and the instantaneous facing angle of the vehicle. We have done some experiments to verify this algorithm in a wide range outdoor environment. The experimental results showed that the algorithm proposed in this paper can calculate the result faster and runs stably.

II. THE ANALYSIS OF THE ALGORITHM

A. The differential model of vehicle motion

The vehicle motion model involved in this algorithm is a differential model, which consider the vehicle travel trajectory as a continuous structure composed of many tiny straight segment, shown in Figure 1.

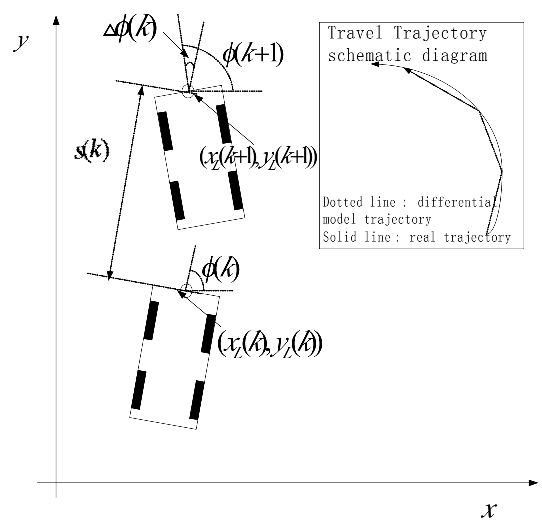

Fig.1 The differential model of vehicle motion

Select the laser radar as the reference point, the location of the vehicle at the k moment is $(x_L(k), y_L(k))$ and the facing angle of the vehicle is $\phi(k)$. After the vehicle traveling a distance of $s(k)$, the location of the vehicle

Daobin Wang is with the Department of Automation, University of Science and Technology of China (phone: 086-0551-5393191; e-mail: wdaobin@ mail.ustc.edu.cn).

Huawei Liang is with the Institute of Advanced Manufacturing Technology, Chinese Academy of Sciences.

Tao Mei is with the Institute of Advanced Manufacturing Technology, Chinese Academy of Sciences.

become to $(x_L(k+1), y_L(k+1))$, and the facing angle become to $\phi(k+1)$. The delt of the vehicle facing angle is $\triangle\phi(k)$. So there is:

$$\begin{cases} x_L(k+1) = x_L(k) + s(k)\cdot\cos\phi(k) \\ y_L(k+1) = y_L(k) + s(k)\cdot\sin\phi(k) \\ \phi(k+1) = \phi(k) + \Delta\phi(k) \end{cases} \tag{1}$$

The model used in this paper just needs to obtain the real-time traveling distance and facing angle of the vehicle, and didn't need to change the steering mechanism of the vehicle, so the measurement of the required variables is more easy and the accuracy is higher than the kinematic model. Besides, the reference point of the model can be set to arbitrary position of the vehicle, thus it avoid the conversion between the different coordinate systems. Meanwhile, for the non-rear-wheel drive car and the crawler robot, the model is still applicable. So the model used in this paper has a certain advantages compared with the vehicle kinematic model.

B. The observation model of environment features

The algorithm proposed in this paper selected the trees, the rods and some columnar things in the environment as the characteristics. Feature extraction method as shown in Fig.2. Assumption the i th feature in the environment is the circle showed in the Fig.2. And the red point in this figure represent the data acquired from lidar. The r_1 and r_n are the two edge rays, and the angle between the two rays is $\triangle\beta$.

$$D_i = \Delta\beta \cdot \frac{1}{n}\sum_{j=1}^{n} r_j \tag{2}$$

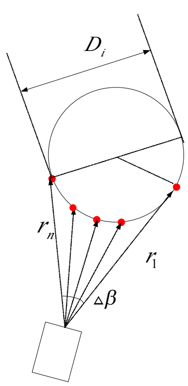

Fig.2 Feature extraction

According to the equation (2) we can calculate the relative position of the feature to the vehicle and the feature radius. Then we match these extracted features of different time, so that the feature points of two consecutive lidar data can establish an association.

The observation model of environment features as shown in Fig.3. The position of the i th road marker in the environment, which is the environment feature, is $B_i(i=1,2,\cdots k)$, whose coordinate is $X_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T$. The observe vector of the i th road marker get from lidar is $Z^i = (z_r^i, z_\beta^i)^T$, in which the z_r^i is the distance between the feature and the lidar, and the z_β^i is the observe angle of the feature from the lidar.

$$Z^i = \begin{bmatrix} z_r^i \\ z_\beta^i \end{bmatrix} = h(X_L, X_i) = \begin{bmatrix} \sqrt{(x_i - x_L)^2 + (y_i - y_L)^2} \\ a\tan(\dfrac{y_i - y_L}{x_i - x_L}) - \phi + \dfrac{\pi}{2} \end{bmatrix} \tag{3}$$

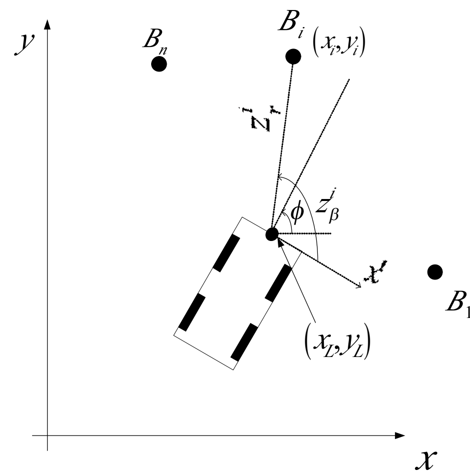

Fig.3 The observation model of lidar

C. The EKF-SLAM algorithm based on the differential model of vehicle motion

The system state vector is $X = \begin{bmatrix} x_L & y_L & \phi & \cdots & x_i & y_i & \cdots \end{bmatrix}^T$. The input vector is $U = \begin{bmatrix} s & \triangle\phi \end{bmatrix}^T$. The observe vector is $Z = \begin{bmatrix} \cdots & Z^i & \cdots \end{bmatrix}^T$. The system predict equation and the system observe equation can be written as (4):

$$\begin{cases} X(k+1) = F(X(k), U(k)) \\ Z(k) = H(X(k)) \end{cases} \tag{4}$$

In order to record conveniently, we denote:

$$X_L = \begin{bmatrix} x_L & y_L & \phi \end{bmatrix}^T \tag{5}$$

$$X_b = \begin{bmatrix} \cdots & x_i & y_i & \cdots \end{bmatrix}^T \tag{6}$$

So the system state vector can be written as:

$$X = \begin{bmatrix} X_L^T & X_b^T \end{bmatrix}^T \tag{7}$$

Thus, the prediction equation can be written as:

$$\begin{bmatrix} X_L(k+1) \\ X_b(k+1) \end{bmatrix} = \begin{bmatrix} f(X_L(k), U(k)) \\ X_b(k) \end{bmatrix} \tag{8}$$

The observe equation can be written as:

$$\begin{bmatrix} \vdots \\ Z^i \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ h(X_L(k), X_i(k)) \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ h_i(X(k)) \\ \vdots \end{bmatrix} \tag{9}$$

$J(k)$, $B(k)$ and $H(k)$ are all Jacobian matrices, as shown in (10), (11) and (12):

$$J(k) = \frac{\partial F}{\partial X} = \begin{bmatrix} \frac{\partial f}{\partial X_L} & \frac{\partial f}{\partial X_b} \\ \frac{\partial X_b}{\partial X_L} & \frac{\partial X_b}{\partial X_b} \end{bmatrix} = \begin{bmatrix} J_1(k) & 0 \\ 0 & I(k) \end{bmatrix} \tag{10}$$

$$B(k) = \frac{\partial F}{\partial U} = \begin{bmatrix} \frac{\partial f}{\partial U} \\ \frac{\partial X_b}{\partial U} \end{bmatrix} = \begin{bmatrix} B_1(k) \\ 0 \end{bmatrix} \tag{11}$$

$$H(k) = \frac{\partial H}{\partial X} = \begin{bmatrix} \vdots & \vdots \\ \frac{\partial h_i}{\partial X_L} & \frac{\partial h_i}{\partial X_b} \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots \\ H_{a,i}(k) & H_{b,i}(k) \\ \vdots & \vdots \end{bmatrix} \tag{12}$$

In which, $H(k)$ can be simply written as follow:

$$H(k) = \begin{bmatrix} H_a(k) & H_b(k) \end{bmatrix} \tag{13}$$

The prediction of the system state is $\bar{X}$, and the estimation of the system state is $\hat{X}$. The covariance matrix of the system state vector $P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$. V is the covariance matrix of U, and $Q_V = B_1 \cdot V \cdot B_1^T$.

So we can derive the prediction of the system state, which is shown in (14).

$$\begin{cases} \bar{X}_{k+1|k} = F(\hat{X}_{k|k}, U(k)) \\ \bar{P}_{k+1|k} = \begin{bmatrix} J_1(k) \cdot P_{11} \cdot J_1^T(k) + Q_V(k) & J_1(k) \cdot P_{12} \\ P_{21} \cdot J_1^T(k) & P_{22} \end{bmatrix} \end{cases} \tag{14}$$

And the updated state of the system is:

$$\begin{cases} \hat{X}_{k+1|k+1} = \bar{X}_{k+1|k} + K_k \cdot (Z_k - H(\bar{X}_{k+1|k})) \\ \hat{P}_{k+1|k+1} = [I - K_k \cdot H(k)] \cdot \bar{P}_{k+1|k} \end{cases} \tag{15}$$

The K_k in (15) can be calculated as follow:

$$\begin{aligned} K_k = &\begin{bmatrix} \left(J_1(k) \cdot P_{11} \cdot J_1^T(k) + Q_V(k)\right) \cdot H_a^T(k) + J_1(k) \cdot P_{12} \cdot H_b^T(k) \\ P_{21} \cdot J_1^T(k) \cdot H_a^T(k) + P_{22} \cdot H_b^T(k) \end{bmatrix} \\ &\cdot \Big(\left[H_a(k) \cdot \left(J_1(k) \cdot P_{11} \cdot J_1^T(k) + Q_V(k)\right) + H_b(k) \cdot \left(P_{21} \cdot J_1^T(k)\right)\right] \\ &\cdot H_a^T(k) + \left[H_a(k) \cdot \left(J_1(k) \cdot P_{12}\right) + H_b(k) \cdot P_{22}\right] \cdot H_b^T(k) + R(k)\Big)^{-1} \end{aligned} \tag{16}$$

III. THE VEHICLE TRAJECTORY TRACKING EXPERIMENT IN WIDE RANGE OUTDOOR ENVIRONMENT

In order to verify the EKF-SLAM algorithm based on the differential model of vehicle motion, we did related trajectory tracking experiment in a wide range outdoor environment. The vehicle we used in this experiment is showed in Fig.4. The parameters of the sensors equipment on this vehicle are showed in (17).

Fig.4 Intelligent Pioneer

$$\begin{aligned} \sigma_s &= 0.07 \\ \sigma_{\Delta\phi} &= 7\pi / 180 \\ \sigma_r &= 0.12 \\ \sigma_\beta &= \pi / 180 \end{aligned} \tag{17}$$

The experiment environment is a section road around a lake, which has 5,000 meters long, as shown in Fig.5.

Fig.5 The selected path of our experiment

The initial value of the covariance matric of system state error, P, is very essential. Through many repeated experiments we set the initial value:

$$P_{0|0} = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.001 \end{bmatrix} \tag{18}$$

$$P_0 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{19}$$

Where, the equation (18) is the initial covariance matric of the vehicle state error, and the equation (19) is the initial covariance matric of the road markers state error.

The experiment result is showed in Fig.6, Fig.7, Fig.8, Fig.9 Fig.10 and Fig.11. Fig.6 shows the trajectory tracking result, in which, the blue circles are the features in the environment around the road, the black line is the GPS data, the blue line is the result of dead reckoning and the red line is the result of the algorithm proposed in this paper. Fig.7 is the errors of the two algorithms. According to the experiment result, it's easy to know that the EKF-SLAM algorithm proposed in this paper is better than the dead reckoning. At the same time, we show the matching frequency distribution of the extracted feature points in Fig.8.

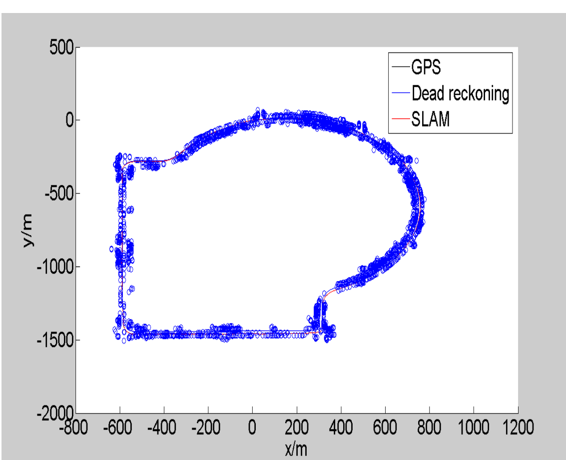

Fig.6 The trajectory tracking experiment

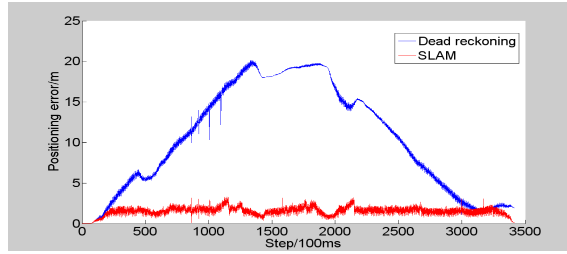

Fig.7 The error of trajectory tracking

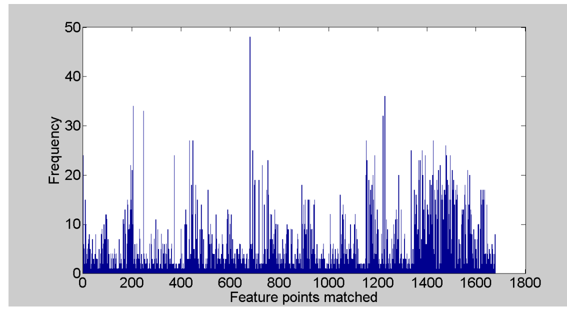

Fig.8 the matching frequency distribution

Meanwhile, in order to know more information on the algorithm, we respectively draw the standard deviation changes of the x_L, y_L, ϕ in Fig.9, Fig.10 and Fig.11.

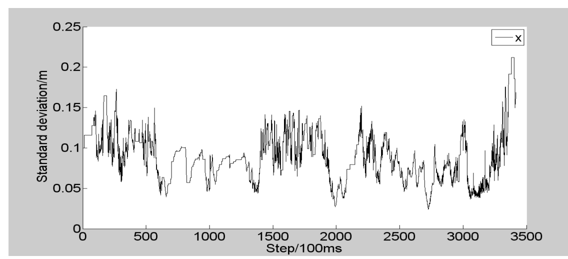

Fig.9 The standard deviation change of x_L

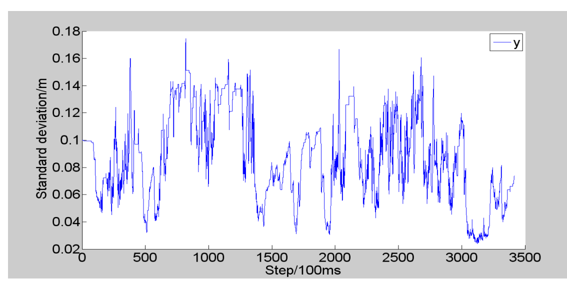

Fig.10 The standard deviation change of y_L

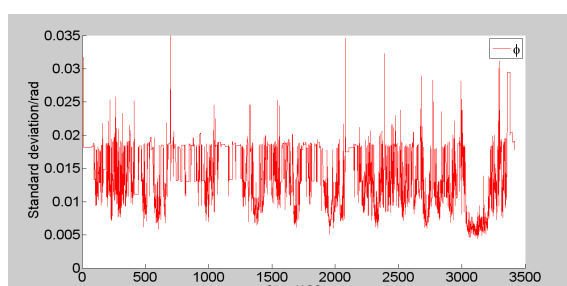

Fig.11 The standard deviation change of ϕ

SLAM algorithm is characterized by using the environmental information to eliminate the random noise and the cumulative error of the system. Especially in the case of the closed-loop path, the positioning accuracy of repeated sections would be relatively higher. For the unfamiliar environment SLAM also can maintain a relatively good positioning accuracy. EKF-SLAM is the first SLAM algorithm to be proposed. Its computational complexity is increasing along with the increase of the feature point number, and its model versatility is very poor. The algorithm proposed in this paper is an improvement of it.

According to the experiment result we can know that the positioning accuracy of the algorithm proposed in this paper is high. The max error is 3.28m, and keeps the error at about 1.5m in the most of time. On the contrary, due to the impact of the cumulative error, the max error of dead reckoning is up to 20.15m.

Meanwhile we can know that when the vehicle returns to the starting point, the error of the algorithm dropped to 0.15m quickly. This phenomenon indicates that the accuracy of EKF-SLAM is higher in a familiar environment.

From the change of the state variable standard deviation we can know the convergence of the algorithm. Due to the number of feature points is tremendous and the feature points are not always existed during the whole experiment process, so we only show the changes of the standard deviation of vehicle state variables in this paper. From the result we can see that the change of standard deviation is relatively stable and the value are all very small, so the algorithm convergence is good.

IV. CONCLUSION

The EKF-SLAM based on the differential model of vehicle motion can not only effectively inhibit the random noise and cumulative error, but also is more concise and more generic compared with the traditional EKF-SLAM. The algorithm proposed in this paper need to real-time obtain the facing angle and the travel distance of the vehicle. Compared with the EKF-SLAM based on the vehicle kinematic model, the facing angle and the travel distance can be measured more easily and more accurately. Meanwhile, because it just need a minor improvement on the bases of dead reckoning and has a significant improvement on the positioning accuracy, so the algorithm proposed in this paper has a certain practical research value.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant Nos. 91120307 and the Natural Science Foundation of Jiangsu Province under Grant Nos.BK2012151.

REFERENCES

[1] Smith, R, M. Self, and P. Chesseman. Estimating uncertain spatial relationships in robotics[C]//In Proceedings of Conference on Uncertainty in Artificial Intelligence. 1988.

[2] José E. Guivant, Eduardo Mario Nebot, Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation[J]. IEEE Transactions on Robotics and Automation, 2001.17(3):242-257.

[3] J.A. Castellanos, R. Martinez-Cantin, J.D. Tardós , J. Neira, Robocentric map joining: Improving the consistency of EKF-SLAM[J]. Robotics and Autonomous Systems, 2007. 55(1): 21–29

[4] Control, Autom Dept, SLAM for robot navigation[J]. IEEE Aerospace and Electronic Systems Magazine, 2008. 23(12): 16-19

[5] Hwang A , Seong W, Lee PM, Experiment on Simultaneous Localization and Mapping Based on Unscented Kalman Filter for Unmanned Underwater Vehicles[J]. International Journal Of Offshore And Polar Engineering, 2012. 22(1): 63-68

[6] Carlone L, Ng MK, Du JJ, Bona B, Indri M, Simultaneous Localization and Mapping Using Rao-Blackwellized Particle Filters in Multi Robot Systems[J]. Journal Of Intelligent & Robotic Systems, 2011. 63(2): 283-307

[7] Dong J.F., W.S. Wijesoma, A.P. Shacklock, An efficient rao-blackwellized genetic algorithmic filter for SLAM[C]//In Proceedings - IEEE International Conference on Robotics and Automation. 2007.

[8] Kim, C., R. Sakthivel, and W.K. Chung. Unscented FastSLAM: A robust algorithm for the simultaneous localization and mapping problem[C]//In Proceedings - IEEE International Conference on Robotics and Automation. 2007.

[9] Wang, Z., S. Huang, and G. Dissanayake, D-SLAM: A decoupled solution to simultaneous localization and mapping[J]. International Journal of Robotics Research, 2007. 26(2): 187-204.

[10] Eliazar AI, Parr R, DP-SLAM 2.0[C]// IEEE International Conference On Robotics And Automation. 2004

[11] Cheein FA, Steiner G, Paina GP, Carelli R, Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection[J]. Computers And Electronics In Agriculture, 2011. 78(2): 195-207

[12] Paskin, M., Thin junction tree filters for simultaneous localization and mapping[D], University of California, Berkeley, 2002

[13] Checchin P, Gerossier F, Blanc C, Chapuis R, Trassoudaine L, Radar Scan Matching SLAM Using the Fourier-Mellin Transform[J]. Field And Service Robotics, 2010. 62: 151-161

[14] Hugh, D.W. and B. Tim, Simultaneous localization and mapping: part Ⅰ[J]. IEEE Robotics and Automation Magazine, 2006. 13(2): 99-110.