A neural network approach to indoor mobile robot localization

Huijun Li, Ying Mao, Wei You, Bin Ye, Xinyi Zhou
School of Information and Control Engineering, China University of Mining and Technology
Xuzhou 221116, China
plutoli@163.com, 1632298575@qq.com, uv_cumt@qq.com, yebin@cumt.edu.cn

***Abstract*—In order to improve the real-time performance and accuracy of localization for mobile robot in indoor environment, a neural network data fusion approach is proposed to eliminate the affection caused by errors from environment or measurements. In the approach, the odometry data are firstly obtained by calculating the collected encoder data through the Dead Reckoning (DR), then we fuse the odometry data and the lidar data by inputting them into a three-layer neural network. Experimental results show that the trained network improved the robot localization performance and its position accurate is within 6cm with good real time response.**

Keywords- mobile robot; indoor localization; odometry; neural network

I. INTRODUCTION

Localization is a key problem in mobile robot research. In recent years, the application of indoor mobile robots has become more and more extensive, and many experts and scholars have also proposed different solutions to the problem of indoor mobile robot localization [1,2,3,4]. Based on Radio Frequency Identification (RFID) technology [5] or other external sensors are commonly used for localization, they can achieve a decimeter order location accuracy [6]. However, these methods need to change the indoor environment, and when external sensors fail, the localization accuracy will be greatly reduced. Another mainstream method is to use probabilistic model [7] to fuse multi-sensor information to complete the localization, this approach usually requires expensive sensors and a lot of computing resources, and the localization accuracy is dependent on the accuracy of the initial position.

Neural networks are widely used in data fusion, we consider applying it to sensor data fusion to improve localization accuracy. Odometry data and lidar data are used as input features of the 3-layer nonlinear neural network. Through the neural network self-learning and adaptive abilities, the trained network is deployed to the mobile robot for testing, by changing the location of the obstacles in the original environment. Experimental results show that the three-layer neural network can provide accurate localization data in real-time.

II. CALCULATE ODOMETRY INFORMATION

Dead Reckoning (DR) is one of the elementary methods for mobile robot localization. Its principle is that the robot moving slowly in a high speed sampling period is regarded as a straight line movement with unchanged direction.

Here we consider a two-wheel differential chassis mobile robot platform. The rotation speeds of the left and right wheels are measured and denoted by n_l、 n_r. The conversion of rotation speed to line-speed is:

$$v = \frac{30n}{\pi r} \tag{1}$$

where r is the wheel radius of robot, n is the rotation speed feedbacking from the encoder.

Motion diagram of adjacent sampling periods is shown in Fig. 1. It is assumed that the robot moves linearly in adjacent sampling periods, then the speed of the robot is:

$$v = \frac{v_r + v_l}{2} \tag{2}$$

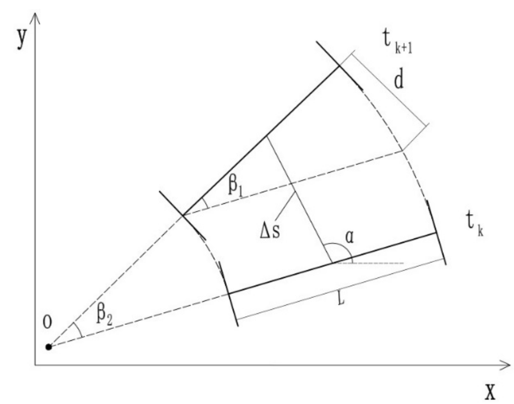

Figure 1. Motion changes in adjacent segments

where L is the distance between the two wheels of the robot. Point o is the rotation center of the robot at adjacent moments. β_1 is the angle between the axis of the robot wheels at the current moment and that at the previous moment. β_2 is the angle at which the robot moves around the center o at adjacent moments. It can be seen from the geometry that $\beta_1 = \beta_2$. d is the distance that right wheel covers more than the left, and $d = (v_r - v_l)\Delta t$. Δt is one sampling period. Δs is the distance traveled by the robot during the sampling period. α is the angle between the x-axis and the robot center point at adjacent moments.

It is assumed that the robot's direction does not change during adjacent sampling periods, so:

$$\Delta s = v \cdot \Delta t \tag{3}$$

$$\beta_1 \approx sin\beta_1 = \frac{d}{L}. \tag{4}$$

Substitute the formula (2) into formula (3), we can obtain

$$\Delta s = \frac{v_r + v_l}{2}\Delta t \tag{5}$$

2473-3636/20/$31.00 ©2020 IEEE
DOI 10.1109/DCABES50732.2020.00026

The angular velocity of the robot moving around the center is:

$$\omega = \frac{\beta_2}{\Delta t} = \frac{\beta_1}{\Delta t} \tag{6}$$

Substitute the formula (5) into formula (6):

$$\omega = \frac{v_r - v_l}{L} \tag{7}$$

$$\alpha_{k+1} = \alpha_k + \omega \cdot \Delta t \tag{8}$$

Substitute the formula (1) and (7) into formula (6):

$$\alpha_{k+1} = \alpha_k + \frac{30(n_r - n_l)}{\pi r L}\Delta t \tag{9}$$

The distance that robot moves within the sampling period is Δs, decompose Δs into the x and y axes of the world coordinate system to complete the odometry solution:

$$x_{k+1} = x_k + \Delta s \cdot \cos(\alpha_{k+1}) \tag{10}$$

$$y_{k+1} = y_k + \Delta s \cdot \sin(\alpha_{k+1}) \tag{11}$$

Together with formula (5)、(9) 、(10) and (11), get the final odometry solution expression:

$$x_{k+1} = x_k + \frac{v_r + v_l}{2}\Delta t \cdot \cos\theta_k \tag{12}$$

$$y_{k+1} = y_k + \frac{v_r + v_l}{2}\Delta t \cdot sin\theta_k \tag{13}$$

where $\theta_k = \alpha_k + \frac{30(n_r - n_l)}{\pi r L} \cdot \Delta t$.

It can be seen from the formula (12) and (13) that derive odometry data by encoder data requires knowing the initial position of the robot, and there will be cumulative errors over time. Therefore, it is necessary to fuse the lidar data to reduce the error, and through the self-learning ability of neural network to reduce the impact of odometry error on localization.

III. NEURAL NETWORK MODEL

A. Structure of the Three-layer Network

For a nonlinear neural network, the more hidden layers and the number of neurons in each layer, the better the network model performs, but more layers of the network means increased training time[8], and is inefficient in actual work. If the number of hidden layers is small, the analytical ability of neural network will be insufficient[9]. Due to the high dimension of the input features, and the output layer has only two dimensions. Therefore, we design a three-layer neural network where the number of neurons in the next layer is less than that in the previous layer, so as to achieve the effect of dimensionality reduction.

According to the error in training and that on the test data set, we gradually reduce the number of hidden layers and neurons. Finally, we find that using a 3-layer neural network is enough to achieve a better localization accuracy. The final designed network structure is shown in Fig. 2.

In Fig. 2, E_x、E_y is the odometry data which can be obtained from the formula (12) and (13). L_i $(i = 1,2 \cdots 360)$ is the ranging data obtained from the single-line laser. h_i^j is the j-th neuron in i-th layer. x、y are the output which are the localization result.

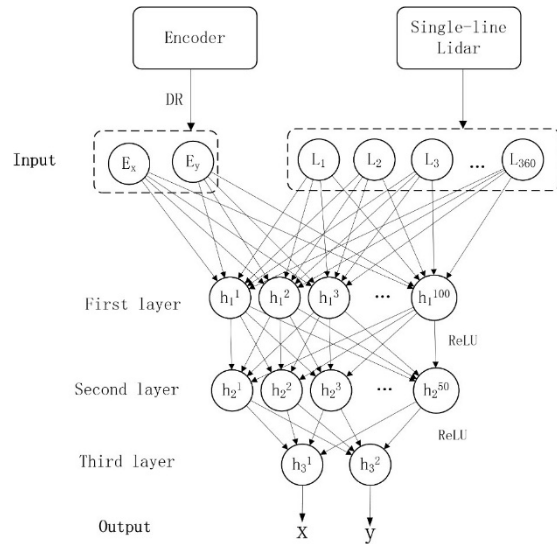

Figure 2. Neural network structure

The input data includes odometry data and 360 ranging data of single-line laser, a total of 362 feature data. The number of neurons in the first hidden layer is 100, after the first layer of dimensionality reduction, obtaining 100 new feature data. We use Rectified Linear Unit (ReLU) as the activation function, which performing a nonlinear transformation on the data. After ReLU nonlinearity, the 100 data are used as the input of the second layer, the role of the second layer is similar to the first layer to complete the dimensionality reduction of the data. Input the nonlinearized data of the second layer to the third layer network and output the localization result after linear adjustment.

Define the loss function as the squared loss, then the expression error of the data of the evaluation sample is:

$$loss_i = (\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2 \tag{14}$$

where $\hat{x}_i$ 、 $\hat{y}_i$ is the position predicting by the neural network, x_i 、 y_i is the i-th set of position data in the training sample. In our experiment, we collect 4746 pairs of training data, so $i \in (1,4746)$.

B. Maximum Error Iterative Training

We collect 4746 pairs of training data samples and 1200 pairs of test samples from the actual environment. To compare the localization accuracy, a UWB device was arranged in the experimental area in Fig. 3 to collect the actual position of the robot, at the same time, this data is used as the true value of neural network training. In order to solve the problem of small average error and large variance of fixed parameter trained network, adopting the maximum error iterative training method.

The maximum error iteration method is, each new training process inherits the network structure obtained from the previous training, and the adjusted parameters are substituted into the new training process. At the beginning, set the learning rate (lr) as 0.01, weight decay (wd) as 0.01. When localization error of the training data are relatively uniform, that is, the error variance is small and the average value may be large. Adjust wd to 0.5 times the initial value and continue training. At this time, all data localization errors decrease synchronously, and the mean and variance of localization errors decrease. When the error gradually converges to a fixed value, adjust lr to 0.5 times of the initial value and adjust wd to 0.25 times of the initial value, and continue training. The variance and mean of the localization error are further reduced, but when the overall errors converge to a constant value. By analyzing the error, we find that only a few data localization errors are large. At this point, set convergence condition to the data with a large error, and adjust lr to 0.25 times the initial value, maintain wd to 0.25 times of the initial value in the previous step. After the training, the errors of the previous data with large localization errors are significantly reduced, and the variance and mean of the localization errors of the overall data are also decreasing. Finally, adjust lr and wd to 0.01 times of their initial values. After the retraining, the localization errors converge to ±9cm, and the variance of the errors are smaller. The pseudocode for this algorithm is shown in Algorithm 1.

Algorithm 1 Maximum error iteration training

Each new training process inherits the network structure obtained from the previous training

```
Set lr_0 = 0.01, wd_0 = 0.01
if loss → err1 and var(loss) < value_const then
    loss is the set of loss_i, the value_const can be observed from the training process
    lr = lr_0, wd = 0.5wd_0,
      if loss → err2 then
        lr = 0.5lr_0, wd = 0.25wd_0
          if loss → err3 then
            Compare the loss_i, find localization errors of some data are bigger, set these data as the convergence condition
            lr = 0.25lr_0, wd = 0.25wd_0
              if loss → err4 then
                lr = 0.01lr_0, wd = 0.01wd_0
                  if loss → err5 then
                    Finish training
                  end if
              end if
          end if
      end if
end if
```

IV. Experimental Results and Analysis

In order to verify the approach proposed in this paper, we improve the Turtlebot robot by adding a STM32 circuit board as the lower computer and installing a single-line laser, and use the improved robot as the experimental platform. The lower computer collects the signal of the encoders. The upper computer model is Inter NUC8I5BEK, see Table Ⅰ for specific parameter configuration. The data collected by the lower computer is sent to the upper computer through serial communication, at the same time, the upper computer receives the signal from the lidar, all data are processed on the upper computer.

TABLE 1 Upper Computer

Inter NUC8I5BEK	
CPU	I5-8259U
RAM	DDR4-2440Hz/4G
Hard disk	SSD/240G

A. Indoor Environmental Testing

Figure 3. Experimental area

The most commonly used localization method in robot localization is the probabilistic localization method based on particle filtering. Deploy the open sourced algorithm Adaptive Monte Carlo Localization (AMCL) based on particle filtering in ROS on the mobile robot platform.

Put the deployed experimental platform with AMCL algorithm and proposed algorithm into the 8m×6m environment shown in Fig. 3, randomly set target points for localization testing in the environment. Collect the neural network localization results、AMCL localization result and UWB localization data for comparison. The three-layer neural network can complete the movement process smoothly. Figures 4 shows the comparison of AMCL and the proposed algorithm in terms of localization accuracy.

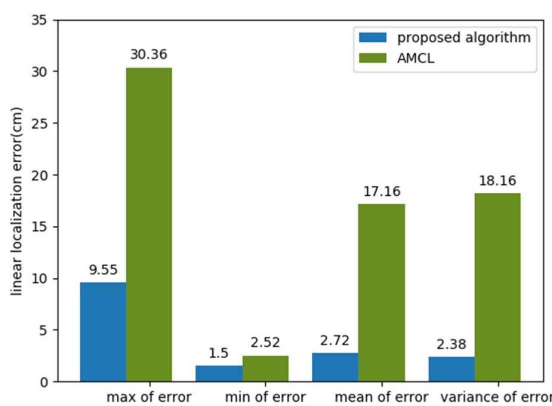

Figure 4. Experiment area test data

From Fig. 4, we can see that proposed algorithms is better than AMCL in all aspect, which proves that proposed algorithms works better than classical probabilistic localization method.

B. Comparison of Localization Performance in Changed Environment

Change the position of obstacles in the test area for testing again, and the test result show in Fig 5 and Fig 6.

It can be seen from experiments and Fig 5、Fig 6 that the mean value of the linear localization error in this paper is 5.23cm, while the average linear localization error of AMCL is 18.46cm. In [10] mentioned that the WIFI localization algorithm based on GAN has an average linear localization error of about 29cm. The improved Monte Carlo localization error in [11] is 21.6cm, the localization accuracy of this paper improves obviously. In terms of localization speed, the proposed algorithm in this paper completes the localization process in 18.6ms, compared with 30ms in [10] and 768.2ms of AMCL, the speed is improved greatly. In addition, the localization error variance of this paper is 2.38cm^2, and its stability is better than that of AMCL (20.58cm^2) and that in [10] (0.09m^2). The experiment results show that the proposed algorithm has better stability, higher localization accuracy, faster localization speed, and better performance in all aspects.

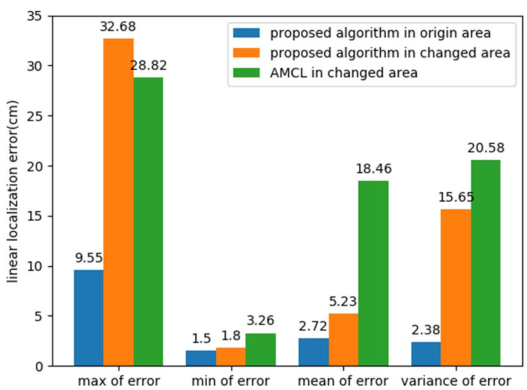

Figure 5. Comparison between the method proposed and AMCL algorithm

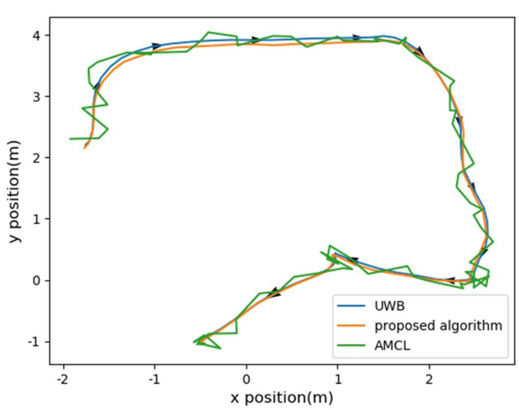

Figure 6. Movement track comparison

V. CONCLUSION

Applying neural network fusion sensors for localization improves the stability of the mobile robot system. At the same time, the self-learning of the neural network can handle the disturbance of the sensor and the external environment, improve the anti-interference ability of localization, and provide localization accuracy higher than the general probability localization method. For the localization in different indoor environments, there are still some shortcomings in the proposed algorithm. When put the robot platform into a different environment to test the neural network, we find that there is a large error in one of the x and y directions, and a small error in the other direction. Combining more sensors and applying a more complex neural network structure to adapt to different indoor environments is the direction of further research.

REFERENCES

[1] Huaping Liu,Xiangmei Zhen,Fuchun Sun. Orientation estimate of indoor mobile robot using laser scans [J]. Journal of Tsinghua University(Science and Technology) , 2018, 58(07):609-613.

[2] Y. Dobrev,S. Flores and M,Vossiek. Multi-modal sensor fusion for indoor mobile robot pose estimation[C]//2016 IEEE/ION Position,Location and Navigation Symposium (PLANS). Savannah:USA,2016:553-556.

[3] Zali,M. Bozorg and M. T. Masouleh. Localization of an indoor mobile robot using decentralized data fusion[C]//2019 7th International Conference on Robotics and Mechatronics (ICRoM). Tehran:Iran,2019:328-333.

[4] Eman and H. Ramdane. Mobile Robot Localization Using Extended Kalman Filter[C]//2020 3rd International Conference on Computer Applications & Information Security (ICCAIS).Riyadh:Saudi Arabia,2020:1-5.

[5] E. DiGiampaolo and F. Martinelli. Mobile Robot Localization Using the Phase of Passive UHF RFID Signals[J]. IEEE Transactions on Industrial Electronics, 2014, 61(1):365-376.

[6] A. Motroni, A. Buffi and P. Nepa. Localization of a mobile device equipped with an RFID reader[C]//2017 IEEE International Conference on RFID Technology & Application (RFID-TA). Warsaw:Poland,2017: 74-79.

[7] J. Luan, R. Zhang, B. Zhang and L. Cui. An Improved Monte Carlo Localization Algorithm for Mobile Wireless Sensor Networks[C]//2014 Seventh International Symposium on Computational Intelligence and Design.Hangzhou:China,2014: 477-480.

[8] D. C. Cireşan, U. Meier, L. M. Gambardella and J. Schmidhuber. Deep, Big, Simple Neural Nets for Handwritten Digit Recognition[J]. Neural Computation, 2010, 22 (12):3207-3220.

[9] Heaton, Jeff. Introduction to Neural Networks with Java[M]. New York,USA:Heaton Research, 2008.

[10] Mengyuan Li,Fengyu Zhou,Tian Tian,Lei Yin. Design of the Indoor WiFi Cloud Positioning System Based on GAN and Its Application to Service Robots [J]. Robot, 2018, 40(05):693-703.

[11] Ting Wang. Research on Monte Carlo Self-Localization Algorithm for Soccer Robot in RoboCup Humanoid League [D]. Jilin University, 2012.