



WalkSLAM: A Walking Pattern-Based Mobile SLAM Solution

Lin Ma¹(✉), Tianyang Fang¹, and Danyang Qin²

¹ School of Electronics and Information Engineering,
Harbin Institute of Technology, Harbin, China
malin@hit.edu.cn

² Electrical Engineering College, Heilongjiang University, Harbin, China

Abstract. In indoor localization scenarios, a sheer coordinate with respect to a basis is insufficient to indicate the users' situation due to a lack of information about landmarks distributed in the environments. To extract landmarks' information manually, however, is inefficient and thus vulnerable to changes of the environments. Simultaneous localization and mapping can solve the localization and landmarks' information extracting problems. This paper presents WalkSLAM, a SLAM solution that estimates both the path taken by the user and the locations of Wi-fi devices in the indoor space, using a smartphone. This solution extends the previous work by introducing human walking patterns into the specific SLAM problem. Experiments demonstrate that the improvement consists of increased efficiency of the particle filter, and hence, of the overall algorithm, and a better estimation of the user's location and path.

Keywords: Indoor localization · Simultaneous localization and mapping · Walking pattern · Walk ratio

1 Introduction

In the last decades, both military and civil have seen increasing need for efficient and accurate location-based service. Location-based service, as its name indicates, controls features and provides functions regarding location data of the users. Thus, the basis of location-based service is to provide a location estimate of the user according to the measurements of devices distributed in the environment taken by the user's device, which requires mapping measurements into exact locations.

In order to provide massive location-based service in large-scale indoor scenario both efficiently and economically, the idea of using a smartphone to run SLAM algorithm while the user carries it and walk around the indoor [1–3] space comes out and draws researchers' immediate attention. But the inaccurate nature of smartphone sensors will lead to an undesirable uncertainty of both the location estimates and landmark estimates.

1.1 Related Work

Previous work has addressed this accuracy issue of smartphone SLAM in different levels. SmartSLAM [3] used path-smoothing model and building orientation model to supervise the heading orientation of the user in order to smooth the noisy path estimate; simultaneously localization and configuring [1] used fixed step length model and EKF to overcome the unreliability of motion measurements. However, fixed step length is still not optimal. Some researches also focused on elimination of Gaussian noise in a RSSI-based localization scenario [4].

This paper presents WalkSLAM, a SLAM solution runs on smartphones. WalkSLAM contributes previous work by adding human walking pattern elements into the SLAM algorithm, both on prior stage and posterior stage. The evaluation on live data shows a noticeable improvement on the path estimation and landmark estimation.

The rest of the paper is organized as follows: In Sect. 2, the problem statement and the model are discussed. In Sect. 3, we introduce a new SLAM algorithm named WalkSLAM. The implantation of WalkSLAM and some refining maneuvers are introduced shortly after. In evaluation section, live data is analyzed and the average errors of both path estimation and landmark estimation are discussed separately with a comparison between WalkSLAM solution and common FastSLAM solution.

2 Problem Definition and Modeling

The original aim of the research is to map an indoor environment using a smartphone without any prior information from the environment. The smartphone is capable of motion sensing and RSSI observing, and in a common SLAM system, the motion sensing ability guarantees the control info of a robot, where location estimation can be extracted, and the RSSI observing achieved by Wi-fi module inside the smartphone can obtain observation of the accessing points working as the landmarks. After this thought, a hidden Markov model [5–7] can be applied to our scenario.

Figure 1 represents a hidden Markov model. The poses make the core Markov chain, and in our situation, they are the locations on the path during the navigation; the state transition means a step of the user is taken. Because we do not know the actual coordinate of the locations, they become the hidden states.

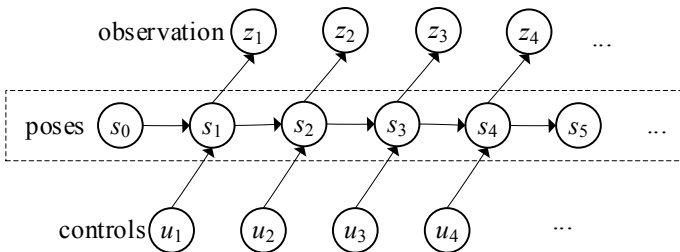


Fig. 1. Hidden Markov model in SLAM problems

Now, we can effectively describe our problem as a SLAM problem. The poses are the user's location, and the current pose will be denoted s_t . Poses evolve according to the motion model [7]:

$$s_t \sim p(s_t | u_t, s_{t-1}) \quad (1)$$

Among which s_t is the current pose, s_{t-1} the previous, and u_t the current control. The control consists of motion measurements from the device, which in our case is the smartphone. After initializing each pose, to map the surroundings, the device senses landmarks using sensors. Sensor observations follow the measurement model:

$$z_t \sim p(z_t | s_t, \theta) \quad (2)$$

where θ is a set consisting of all the landmarks. Now, the SLAM problem can be formulated. SLAM problem is to make a refined estimation of all landmarks θ and all poses $s^t = \{s_0, s_1, s_2, \dots\}$ along the user's path from the controls and observations, which can be described as $p(s^t, \theta | z^t, u^t)$. The controls, as stated above, are the motion measurements from motion sensors on the phone, which indicates a step's heading and orientation. After an initialization of the starting pose, given previous and current control, an estimate of the current state can be calculated. However, the motion sensors are alarmingly inaccurate; thus, in further discussion, we decide to add some prior information into the controls for better performance.

3 WalkSLAM

3.1 Particle Path Sampling

With the control and the observation of our SLAM system defined, we can present our SLAM problem solution in the following steps with implantation details. As shown in Fig. 2, the algorithm begins with the path particle sampling after the input of motion measurements u_t from motion sensors. Due to the nature of human walking pattern, we do not sample the pose s_t directly but rather estimate the transition from s_{t-1} to s_t which is denoted $\overrightarrow{(s_t - s_{t-1})}$.

The transition can be described as a combination of the fixed step length and the heading. For the filtering of the heading direction, instead of the path-smoothing method, we present a new filtering model:

$$\omega_t = \begin{cases} \omega_{t-1}, & |\omega_{t-1} - \omega_t| > \varepsilon_\omega \\ \omega_t, & |\omega_{t-1} - \omega_t| \leq \varepsilon_\omega \end{cases} \quad (3)$$

The rationale behind this model is that first the motion sensor is pretty noisy and unreliable according to previous reports [1, 8], and because of that, in some degree we can safely ignore small direction changes without worrying too much about losing the details of heading since Monte Carlo method involved in the sampling phase can guarantee the coverage of most minor direction changes.

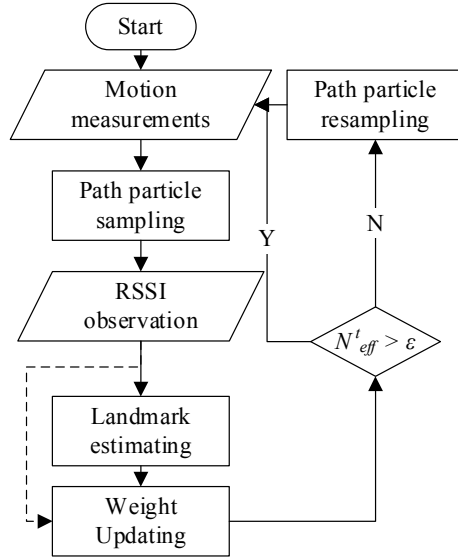


Fig. 2. SLAM algorithm loop. Notice that the dotted line connecting RSSI observation with weight updating step means the observation takes part in the weight’s calculation

After the path sampling, we get a set of path particles, and according to our design, the next step should be landmark estimation, where we initialize and update the landmarks’ locations using the observations. However, since our observation is the 1D RSSI measurements, a single observation is not enough to initialize a landmark’s location. Thankfully this problem can be solved using the range-only localization model presented in [9], and the major requirement of this method is several consecutive poses and relative observations.

The updating of landmarks is guided by the observations, so we need a filter to refine the landmark location, in this situation a 2D coordinate, with 1D RSSI as the observations. An extended Kalman filter can be applied to this situation, using the method mentioned in [1].

3.2 Importance Weight Updating

In robotic SLAM systems, the only indicator of the reliability of a particle is how the observation matches up the landmark estimations, and it is called importance weight factor:

$$w_i^{[m]} = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{p(s^{t,[m]}|z^t, u^t)}{p(s^{t,[m]}|z^{t-1}, u^t)} \tag{4}$$

However, in our WalkSLAM scenario, another factor should be taken into consideration to evaluate the degree of the particle path fitting human walking pattern [10], specifically the walk ratio indicator [11]. For convenience, we call the original

importance weight factor observation factor (OF), and the walk ratio indicator walk factor (WF). Then, the new weight becomes:

$$w_t^{[m]} = \text{OF} \cdot \text{WF} \quad (5)$$

As the matter of implantation, for the calculation of OF, we can use the conclusion from [7]:

$$\text{OF} = \frac{p(s^{t,[m]}|z^t, u^t)}{p(s^{t,[m]}|z^{t-1}, u^t)} \stackrel{\text{EKF}}{\approx} \sum_k p(z_t|\theta_k^{[m]}, s_t^{[m]})p(\theta_k^{[m]}) \quad (6)$$

Since we are always sure about the identity of each RSSI scan result, the equation can be simplified to:

$$\text{OF}^{[m]} = \sum_k p(z_t|\theta_k^{[m]}, s_t^{[m]}) \quad (7)$$

Then, we can calculate OF using the Gaussian function:

$$\text{OF}^{[m]} = \sum_k f(z_t|\|\theta_k^{[m]} - s_t^{[m]}\|, \delta_\omega) \quad (8)$$

For the calculation of WF, since the proposal distribution is made to be a Gaussian distribution, according to the Bayesian theory we can simplify the equation as:

$$\text{WF}^{[m]} \propto p(d^{t,[m]}|\Delta^{t,[m]}, \delta_{\text{WR}}) \quad (9)$$

For implementation purpose, we first calculate the average WR of the path particle,

$$\overline{\text{WR}} = \frac{1}{n} \sum_{k=1}^n d_k \cdot \Delta_k \quad (10)$$

Then, through a Gaussian function, we can get the WF of the particle as:

$$\text{WF}^{[m]} = f(\overline{\text{WR}}|R, \delta_{\text{WR}}) \quad (11)$$

where d_k is the step length and Δ_k the time that step costs. R is the average walk ratio of a healthy man, and δ_{WR} the variance of walk ratio.

To sum up, the new weight updating model becomes:

$$w_t^{[m]} = f\left(\frac{1}{n} \sum_{k=1}^n d_k \cdot \Delta_k | R, \delta_{\text{WR}}\right) \sum_k p(z_t|\theta_k^{[m]}, s_t^{[m]}) \quad (12)$$

Then, for further use, we normalize the weights:

$$w_t^{[m]} = w_t^{[m]} \left(\sum_{k=1}^n w_t^k \right)^{-1} \quad (13)$$

3.3 Resampling Particles

After getting the weights of the particle set, we now resample the particle set due to the weight of each particle. To evaluate the effectiveness of a particle set using the weight factor, the effectiveness index is introduced:

$$N_{\text{eff}}^t = \left(\sum_{k=1}^n w_t^k \right)^{-1} \quad (14)$$

As shown in Fig. 2, every time the value of N_{eff}^t gets low enough, the resampling step is triggered; this way we can resample only when we need to, to effectively use all the particles and lower the time consumption.

4 Implementation and Evaluation

In the test, we choose a typical indoor scenario which is a floor of office with a long corridor. The longest part of the corridor is 47.3 m at length, and the overall corridor is 3 m at width. Thirteen access points were accessible along the corridor. A live test in comparison with the ground truth is given in Figs. 3 and 4. Since our goal is to estimate both the path and the landmarks, the localization biases of them are evaluated.

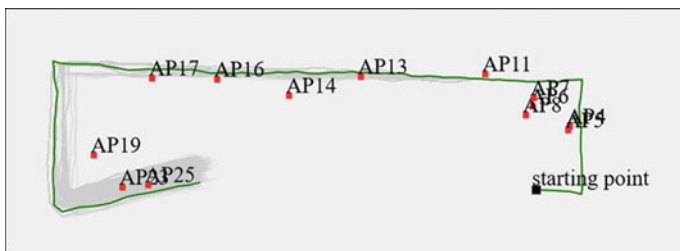


Fig. 3. Replay of a live test

Considering the additive nature of the pedometer bias, tests were run on different lengths to indicate the difference made by navigating different lengths and the results are given in Table 1, although due to the variance of step length the tests can only be grouped by a range of distance they cover.



Fig. 4. Ground truth of the same test with a floorplan

Table 1. Result of three long tests and three short tests

Path length	Test	Localization error (m)	Standard deviation
90–100 m	1	2.97	0.37
	2	2.34	0.29
	3	2.78	0.30
35–40 m	4	1.17	0.11
	5	1.41	0.17
	6	0.94	0.10

We can still conclude from the results that for a short run of the WalkSLAM, the average localization error falls below 3.0 m, which is good enough for a room-level localization scenario. Another essential factor of the evaluation is the localization error of the landmarks. In the three long runs, 12 access points are all observable; thus, we use them to calculate the error.

To better demonstrate the overall performance when localizing the landmarks, we take each landmark estimation in each run as an individual landmark estimation and use the cumulative distribution function to bring out the result. When compared to a common FastSLAM run with an S1 model applied and one without, as shown in Fig. 5, the improvement in performance of WalkSLAM is quite noticeable.

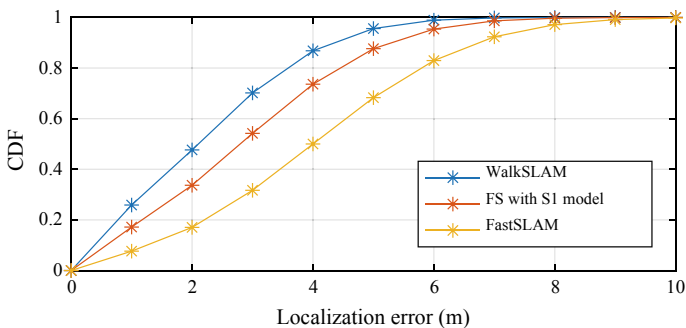


Fig. 5. Comparison of landmark estimating performance between WalkSLAM and traditional FastSLAM

5 Conclusion

In this research, a simultaneous localization and mapping solution that focuses on running on smartphones was developed. Compared to traditional SLAM solutions, in this new solution, the human walking pattern was considered into the algorithm, with a new sampling model and an advanced weight updating model presented, and the implantation methods given in detail. Six live tests were run in a typical indoor environment. The result indicated that the accuracy could fit a room-level localization scenario and the landmark estimation performance of WalkSLAM is noticeably better than that of a traditional solution.

Acknowledgments. This paper is supported by National Natural Science Foundation of China (61571162), Ministry of Education—China Mobile Research Foundation (MCM20170106).

References

1. Bulten W, Van Rossum AC, Haselager WFG. Human SLAM, indoor localisation of devices and users. In: 2016 IEEE first international conference on internet-of-things design and implementation (IoTDI). IEEE; 2016. p. 211–22.
2. Ferris BD, Fox D, Lawrence N. WiFi-SLAM using Gaussian process latent variable models. *Science*. 2007;7:2480–5.
3. Shin H, Chon Y, Cha H. Unsupervised construction of an indoor floor plan using a smartphone. *IEEE Trans Syst Man Cybern Part C Appl Rev*. 2012;42(6):889–98.
4. Patri A, Rath SP. Elimination of Gaussian noise using entropy function for a RSSI based localization. In 2013 IEEE second international conference on image information processing (ICIP-2013). IEEE; 2013. p. 690–4.
5. Arulampalam MS, Maskell S, Gordon N, Clapp T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans Signal Process*. 2002;50(2):174–88.
6. Dellaert F, Fox D, Burgard W, Thrun S. Monte Carlo localization for mobile robots. In: Proceedings 1999 IEEE international conference on robotics and automation (Cat. No.99CH36288C), vol. 2; May, 1999. p. 1322–8.
7. Montemerlo M, Thrun S, Koller D, Wegbreit B. FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: Proceedings of 8th national conference on artificial intelligence/14th conference on innovative applications of artificial intelligence, vol. 68, issue 2; 2002. p. 593–8.
8. Zhao N. Full-featured pedometer design realized with 3-axis digital accelerometer. *Analog Dialogue*. 2010;44:1–5.
9. Olson E, Leonard JJ, Teller S. Robust range-only beacon localization. *IEEE J Oceanic Eng*. 2006;31(4):949–58.
10. Sekiya N, Nagasaki H. Reproducibility of the walking patterns of normal young adults: test-retest reliability of the walk ratio(step-length/step-rate). *Gait Posture*. 1998;7(3):225–7.
11. Rota V, Perucca L, Simone A, Tesio L. Walk ratio (step length/cadence) as a summary index of neuromotor control of gait: application to multiple sclerosis. *Int J Rehabil Res*. 2011;34(3): 265–9.