

Simultaneous Localization And Mapping in a domestic environment

Report Author	Group Partner
Paul Rousse	Jiaheng Qiu

Abstract

Many applications of mobile robots take place in a domestic environment (cleaning robot, wheelchair). Localization and mapping are the first needed knowledge in order that the robot can move in the area. Non professional applications can not benefit from high quality sensors. We tried to work with PSD infrared range sensor. As in most domestical environments, the map can be drawn with lines (that stands for walls) that are either parallel either orthogonal. An Extended Kalman Filter that estimate state with raw measures has been used, in this way we managed to avoid any heavy algorithm of line extraction. We succesfully implemented an algorithm that work on simulation.

1 Introduction

In most applications of mobile robot it is crucial to know the position of the robot in order to success in its task. However, it is not always possible to get any previous knowledge about the environment where the robot evolve in. That is what the SLAM problem tries to solve: localize the robot while simultaneously building a map of the environment, and concurrently trying to minimize errors. The Simultaneous Localization And Mapping problem (SLAM) has been one of the most active field of research in robotic the last 3 decade. [Durrant-Whyte and Bailey, 2006] sum up most of the work on SLAM.

Many different algorithms for SLAM have been developped, mainly because the outcomes of the algorithm depends on the sensors you have, the type of features of your map and the robot motion model. The laser range finder meet a great success in the researcher community ([Jensfelt and Christensen, 1999];[Diosi and Kleeman, 2005]) because of the quality of the measure: "null" propagation time and fairly good range and angular accuracy. However the expensive cost of this sensor keep back costumer from any integration in any unprofessional mobile robot. This explain the interest of researchers and companies in developping algorithm wich can works with cheaper sensors such as the sonar sensor ([Zunino and Christensen, 2001]; [Choi et al., 2008]) or PSD infrared sensors ([Abrate et al., 2007]).

This challenge is mostly about how to build a SLAM algorithm with sparse and noisy data. It is not always possible to take advantages of previous algorithm developped for laser range sensor because most of the time the number of sensors are much fewer than the number of measure of the laser scan. Most of the work try to add constraints to the map in order to reduce the number of possibilities for the mapping and therefore to reduce the error of the map estimation. [Zunino and Christensen, 2001] use an algorithm that can take advantages of geometrics constraints of an indoor environment in order to perform the classification step with only a few and noisy data: *Triangulation Based Fusion* ([Wijk et al., 1998]). This method let to perform a SLAM localization with point features which are defined as intersection of 2 lines features. In practice, points are detected in corner of a wall or a furniture. However this method is relevant only when the robot cross a corner. And therefore, the estimation does not take care of the distance of the robot from a straight wall.

[Choi et al., 2005] has taken advantage of both point features and line features by combining the TBF method and a line extraction algorithm. The feature extraction is used as a measure for the EKF implementation. In [Choi et al., 2008], geometric constraints are used. Every measured point is

stored and used in order to measure features of the map. As for [Nguyen et al., 2006], geometric constraints (such as parallel and orthogonal walls) add information to the map generation and in this way it improve the estimation of localization and mapping.

1.1 Contribution

There are 3 main contributions in this article. First an implementation of a SLAM algorithm with line-feature. Unlike point feature SLAM implementation, line feature have the disadvantages of being continuous. Second, we implemented a geometric constraint of parallel and orthogonal walls inside the EKF (instead of using a constrained line extraction). Third, we manage to get rid of any heavy line extraction algorithm, and we show that a very simple algorithm can be used in order to create new features to the state vector.

1.2 Outline

The first section will present a review of the previous work. The second, the implementation of our SLAM algorithm. And the third part will focus on results and limitation of our algorithm.

2 Related work

Cost constraints have led researchers to develop SLAM algorithm with sparse and noisy sensors. Some solutions based on distance sensors has been proposed. The main idea is to use furnitures as a map for the robot. SLAM performed with sparse and noisy sensor based localization as been one of the challenges in domestic robot research. Most of researcher have used Hough transform in order to detect lines. If enough sensors was available, the line detection was be performed at each step ([Großmann and Poli, 2001]), if just a few are available, the line detection is performed on a set of points measured on several steps.

That is the strategy of [Choi et al., 2008] made an implementation of the SLAM algorithm in an indoor environment by using a smart pre-treatment of all incoming measures from sonar sensors from the beginning of the mapping. This pre-treatment allow to reduce outlier generations and to perform an easier point-to-line association. After this pre-treatment, line features are extracted thanks to a constrained Hough transform around mean angle of all

line features. The EKF estimate each line feature independently. Therefore the constrained is set inside the measure and not inside the EKF.

The Hough transform used in [Choi et al., 2008] is not a probabilistic tool, it is an approximation of a probabilistic tool ([Stephens, 1991]). Therefore, they developpe their own covariance calculation which does not take in account the uncertainty on measure. This empirical method make the EKF difficult to adjust (the uncertainty on the measure does not affect the empirical covariance computation on the Hough transform!

3 My method

Most of the SLAM line-feature implemetation focus on measuring the line parameters in the EKF. We have choosen to directly use the measures of sonars. This choice was made in order to avoid implementing any Hough transform that will take too much computational ressources.

3.1 Extended Kalman Filter

First, we will present the EKF method without any geometric constraints. The EKF model is the same as [Zunino and Christensen, 2001] for the predict step: the state of the EKF is represented by the state of the robot $\mathbf{x}_r = [x_r, y_r, \theta_r]^T$ followed by the N_f state vectors of features. The i^{th} feature is represented just by its parameters in polar coordinates, $\mathbf{x}_i = [\rho_i, \theta_i]^T$.

The predict step does not affect covariance matrices of features because they are not part of the dynamics (they are just measurements).

The observation model compute the distance from the sensor to the i^{th} wall feature in direction of the sensor. ψ is the angle of the sensor in the reference of the robot and d the sensor position along the robot.

$$h_i(\mathbf{x}) = \frac{x_r \cos \theta_i + y_r \sin \theta_i - \rho_i + d \cos(\theta_r - \theta_i)}{\cos(\theta_i - \theta_r - \psi)} \quad (1)$$

The computation of the jacobian show that the farer the distance measured is, higher is the uncertainty (because in this way the angle of the impacted measure is very low and the uncertainties on angles θ_r and θ_i have more influence). Therefore if we want to have a precise estimation of the wall we should have orthogonal measures from it. The most optimal way to set sensors is to set them al around the robot.

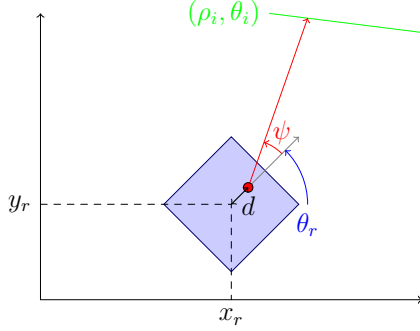


Figure 1: Robot parameters definitions. The robot (in blue) is performing one measure (in red) of the i^{th} feature (in green).

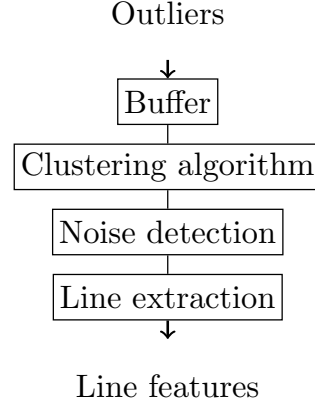


Figure 2: Line extraction algorithm.

3.2 Geometric constraints

Geometries constraints are both represented in the feature addition process and in the state vector of the EKF. We added a state variable α that represent the global orientation of the map according to the initial robot position.

The state vector of the EKF become:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_r & \alpha & \mathbf{x}_1 & \cdots & \mathbf{x}_{N_f} \end{bmatrix}^T \quad (2)$$

We compute the observation model with the geometric constraint:

$$h_i(\mathbf{x}) = \frac{x_r \cos \beta_i(\alpha) + y_r \sin \beta_i(\alpha) - \rho_i + d \cos(\theta_r - \beta_i(\alpha))}{\cos(\beta_i(\alpha) - \theta_r - \psi)} \quad (3)$$

with:

$$\begin{aligned} \beta_i(\alpha) &= \alpha - k \frac{\pi}{2} \\ \text{with } k &= \left\lfloor (\alpha - \theta_i) \frac{2}{\pi} + \frac{1}{2} \right\rfloor \end{aligned} \quad (4)$$

In other words, β_i is the closest angle from θ_i which verify $\alpha \equiv \beta_i \left[\frac{\pi}{2} \right]$.

In order to compute the jacobian matrix, we set:

$$\frac{\partial h}{\partial \alpha} = \frac{\partial h}{\partial \theta_i} \quad (5)$$

with i the observed feature.

3.3 Features addition

After each execution of the EKF step, outliers are saved in a buffer of a fixed size N_{buffer} . This buffer is used then to perform the line feature extraction.

An unsupervised clustering algorithm is then used in order to separate measures that do not come from the same wall. This

First we try to detect if dataset is just noise or a line feature. We do this by comparing the covariance of the a cluster of points to the covariance of the uncertainty of these points. If the covariance is Then we can perform the line extraction step. We use the maximum of likelihood to choose if the new feature will be a vertical or an horizontal line.

3.4 Implementation

The clustering algorithm really depends on the number of sensors and how much the measures are noisy. If the measures are not so noisy and they are few sensors, then the classification can be performed just with indexes of the sensors (the measure is classified by the sensor it is coming from). Otherwise a clustering algorithm is necessary: we used the DBSCAN discussed in [Ester et al., 1996]. The DBSCAN algorithm classify in the same cluster points which are closer than a given distance ϵ .

This choice is not the optimal one, and we believe that it is possible to choose a better one that take in account constraints and probability consideration (such as probability distribution parameters of the hit point after a measure). However, as long as data are not so noisy, the DBSCAN algorithm is enough.

4 Experimental results

4.1 Analyse of the EKF behavior

As shown in the figure 3, the algorithm successfully manage to draw the map with line features and at the same time correct the trajectory. In this simulation, we gave a good and precise idea of α .

The uncertainty on features always decrease (see figure 3d). This is a property of the EKF integration: during the predict step, because features do not take part in the dynamics (unlike the robot), the uncertainties do not increase. Since the update step always increases the belief of the state, the covariance matrix of the feature can just decrease. This is a real issue in case of slidding motion of the robot: whatever the uncertainty Q you have on the measure, the belief of the feature will increase. Therefore the EKF will

always change the robot position because this is the only variable in the state vector that is not fully certain (contrary to every other features parameters).

4.2 Line extraction

The line extraction algorithm is tuned with 2 main parameters:

- N_{buffer} the size of the dataset buffer
- Σ_{noise} the guessed covariance matrix of noise on the measure.

The success of the algorithm mainly depends if the second parameters.

The first one have to be big enough in order to have more points that just noise: if N_{buffer} is too low and measure too noisy, we will never detect a trend of line (but just a gaussian distribution of points). So to detect lines, we need to have enough data. In figure 3, it seems that $N_{buffer} = 100$ was a reasonable value. The draw back of this is that the computational cost increase as the noise on measure increase (in case of our implementation, the complexity of the DBSCAN algorithm is $\mathcal{O}(n \log n)$).

Σ_{noise} is a threshold that avoid line extraction algorithm if data are not relevant. So, if the data are noisy, we need to set Σ_{noise} to a larger value than if they are not.

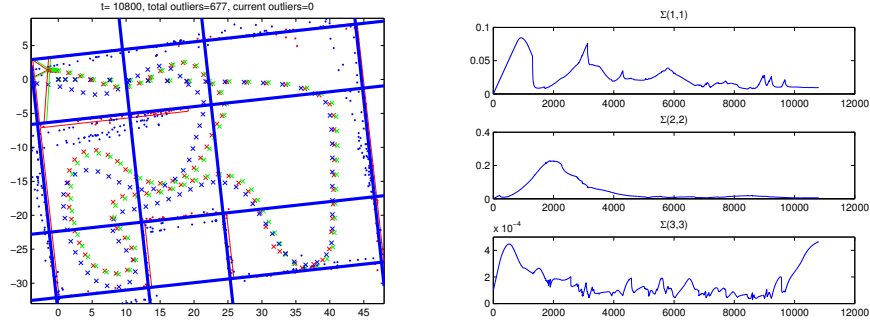
4.3 Data association

Unlike point feature, line feature are continuous, when crossing a corner, it is difficult to keep the data association correct (see figure 4). More over, our implementation of slam with line feature does assume every wall as an infinite line! This create even more spaces where the robot fail to do the right data association.

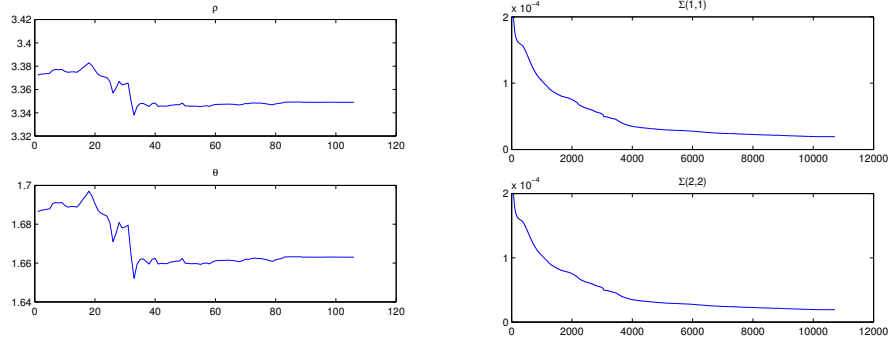
Currently, this is the main obstacle to the well execution of the algorithm. We will discuss in the conclusion the possible solutions in order to get over this difficulty.

5 Summary and Conclusions (0.5–1 page)

Summarize what you have done and make sure that you highlight your contributions. You should not introduce new results in the summary. Results should be introduced in the main sections above. Here you can speculate on how these results could be extended, what would happen in other settings or how the method could be used other domains and how to continue with

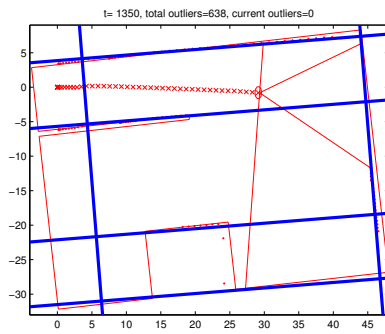


(a) Map of trajectories and features. (b) Uncertainty of the measure of the estimated position.

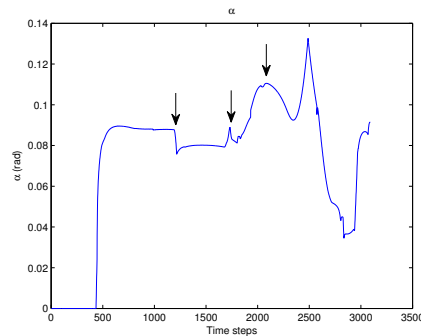


(c) Estimated parameters of the first feature. (d) Coefficient values of the first feature covariance matrix.

Figure 3: Map (3a) and sigma coefficients (3b) of a simulation with a low noise on odometry measure and sensors (about $10^{-2}(m/s, rad/s \text{ and } m)$). The estimate trajectory of the robot (red crosses) is corrected thanks to the map estimation (blue lines) and "stick" to the true trajectory of the robot (in green) instead of following the basic odometry estimate trajectory (blue crosses). Figures 3c and 3d are graphes from the first feature the robot added to its state.



(a) Robot measuring the first corner of the map.



(b) Plot of α .

Figure 4: Each arrow match to a timestep where the robot measure the distance in a corner. The wrong association of the measure to the feature avoid the good estimation of alpha.

the research in the future. In this section you can put statements that one cannot understand unless you have read the paper which is not possible in the abstract for example. You do not need to be as formal in this section.

References

- [Abrate et al., 2007] Abrate, F., Bona, B., and Indri, M. (2007). Experimental ekf-based slam for mini-rovers with ir sensors only.
- [Choi et al., 2005] Choi, J., Ahn, S., and Chung, W. K. (2005). Robust sonar feature detection for the slam of mobile robot. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3415–3420. IEEE.
- [Choi et al., 2008] Choi, Y.-H., Lee, T.-K., and Oh, S.-Y. (2008). A line feature based slam with low grade range sensors using geometric constraints and active exploration for mobile robot. *Autonomous Robots*, 24(1):13–27.
- [Diosi and Kleeman, 2005] Diosi, A. and Kleeman, L. (2005). Laser scan matching in polar coordinates with application to slam. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3317–3322. IEEE.

- [Durrant-Whyte and Bailey, 2006] Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localisation and mapping (slam): Part i the essential algorithms. *IEEE ROBOTICS AND AUTOMATION MAGAZINE*, 2:2006.
- [Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.
- [Großmann and Poli, 2001] Großmann, A. and Poli, R. (2001). Robust mobile robot localisation from sparse and noisy proximity readings using hough transform and probability grids. *Robotics and Autonomous Systems*, 37(1):1–18.
- [Jensfelt and Christensen, 1999] Jensfelt, P. and Christensen, H. I. (1999). Laser based pose tracking. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 4, pages 2994–3000. IEEE.
- [Nguyen et al., 2006] Nguyen, V., Harati, A., Martinelli, A., Siegwart, R., and Tomatis, N. (2006). Orthogonal slam: a step toward lightweight indoor autonomous navigation. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5007–5012. IEEE.
- [Stephens, 1991] Stephens, R. S. (1991). Probabilistic approach to the hough transform. *Image and vision computing*, 9(1):66–71.
- [Wijk et al., 1998] Wijk, O., Jensfelt, P., and Christensen, H. I. (1998). Triangulation based fusion of ultrasonic sensor data. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 4, pages 3419–3424. IEEE.
- [Zunino and Christensen, 2001] Zunino, G. and Christensen, H. I. (2001). Simultaneous localization and mapping in domestic environments. In *Multisensor Fusion and Integration for Intelligent Systems, 2001. MFI 2001. International Conference on*, pages 67–72. IEEE.