Simultaneous Localization And Mapping in a domestic environment

Report Author Group Partner
Paul Rousse Jiaheng Qiu

Abstract

Many application of mobile robots take place in a domestic environment (cleaning robot, wheelchair). Localization and mapping are needed in order the robot know where he his. Scanning laser range sensor are most of the time too expensive to be used on these application, cameras have a high CPU cost and might be too sensitive to lighting conditions. We tried to work with ir range sensor because of a lower cost and the easy data acquisition. An Extended Kalman Filter has been used to approximate the state of the robot and map features. As in most domestical environments, the map can be drawn with walls either parallel either orthogonal. We succefully implemented an algorithm that work on simulation Unfortunately we did not have any real robot to check the algorithm on a real situation.

1 Introduction

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

Many different algorithms for SLAM have been devlopped, mainly because the success of the algorithm depends on the sensors you have, the features on your map and the robot motion. The laser range finder met a great success in the researcher community ([Jensfelt and Christensen, 1999];[Diosi and Kleeman, 2005]). The popularity of this sensor come fom the quality of the measure: "null" propagation time and fairly good range and angular accuracy. However the expensive cost of this sensor make it not usable in an unprofessional mobile robot integration. Cheaper sensors have been used such as sonar sensor ([Zunino and Christensen, 2001]; [Choi et al., 2008]) or PSD infrared sensors ([Abrate et al.,]).

If ultrasonic sensors provide a better range estimate than PSD infrared sensors, they have a severe disadvantage: the time propagation is not negligeable when the distance measured id too far nor the robot speed too high. Therefore, the data are noisier and the fire rate is limited. The feature association task become very hard. In order to get ride of this, [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. However this method is relevant only when the roboot cross a corner. And therefore, the estimation does not take care of the distance of the robot from the wall.

[Choi et al., 2005] has taken advantages of both point features and line features by combining the TBF method and an Extended Kalman Filter (EKF) implementation. Later he adds some realistic constraints that can be met in an indoor environment. [Choi et al., 2008] tried to take in account this by measuring parameters of walls (angle and distance from the robot) and try to estimate them in a constrained EKF. 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 we can have a better 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 contiuous, therefore it is difficult to choose when we need to create a new feature or when we need to keep new points associated to the previous line-feature. Second, we implemented a the constraint of parallel and orthogonal wall inside the EKF (not just in the measure of features). Third, we show that a very simple algorithm can be used in order to create new features. In this way we avoid to use any Hough transform or heavy algorithm.

1.2 Outline

The first section will present a review of the previous workk. The second, the implementation of our SLAM algorithm: the state representation in the Extended Kalman Filter (EKF), the constrained model of the map and the feature addition. And the third part will focus on results and limitation of our algorithm.

2 Related work

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-treamtment 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 featurse 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 uncartainty on the measure does not affect the empirical covariance computation on the Hough transform!

3 My method (1–4) pages

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.

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

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.

3.2 Geometric constraints

Geometrics 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:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_r} \\ \alpha \\ \mathbf{x_1} \\ \vdots \\ \mathbf{x_{N_f}} \end{bmatrix} \tag{2}$$

We compute the observation model with the geometric constraint:

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

with:

$$\begin{array}{l} \beta_i(\alpha) = \alpha - k\, \frac{\pi}{2} \\ with\ k = \left\lfloor (\alpha - \theta_i)\, \frac{2}{\pi} + \frac{1}{2} \right\rfloor \end{array} \tag{4}$$

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:

$$\frac{\partial h}{\partial \alpha} = \frac{\partial h}{\partial \theta_i} \tag{5}$$

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.

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

In order to extract line feature of the dataset, we used the maximum of likelihood between vertical and horizontal line

3.4 Implementation (0–2 pages)

The description of your method should be provided at a level of abstract where implementation details are avoided as much as possible. For conveying general knowledge it is typically uninteresting to know that you implemented your system in C++ or .NET and that you make use of package so and so. However to assess your results it may be important to know some of these details. This section, if present, provides the implementation details. Limit the description to what is important. What language you used to implement your system is in many cases not interesting either but it might be interesting to know that you used a kd-tree to make access to certain data more efficient or that you made this run super fast using the GPU. This is particularly important if you make comparisons of speed. Comparing EKF in Matlab to PF using GPU it would be important to point out theses details but if in the end you are focused on accuracy then it does not matter at all.

Exactly what you put here varies from case to case. Instructions for how to run your system would not end up here unless it is related to the problem you deal with. Remember that it is not a software manual or a user guide.

4 Experimental results (1–4 pages)

In this project your task is to create a working implementation of some estimation method so an experimental evaluation is to be expected. Make experiments that backs up the claims you make. If you have improved / modified some other method, compare your new method with the old. If possible include other methods in the comparison.

Here you should have figures and tables that show the results. It is best if there are numerical measures such as mean square error or a histogram over errors. this is even better if the errors are normalized with the covariance. Often one plots the errors over time as bars or points and the 2 sigma bound as a curve. A figure that one can see that the estimate is following the true values is good if you can show it. All that requires ground truth so if you do not have that your job here is more difficult be creative and show something that indicates it works. For example the innovation process can still be shown.

When writing a paper in a well researched area you would be expected to compare your result to all or a significant fraction of related results and show why your method / contribution is worth publication. This does not apply here but some comparisons will help.

Make sure to provide your experimental setup carefully. You want some-one else to be able to replicate what you did.

A paper where everything works flawless according to the experiments is typically looked upon with skepticism. Nothing is perfect! If you get perfect results it often means that your tests where not challenging enough. As a reader I want to know what the limitations are so push it to the limit or at least provide solid arguments for where such limits might be.

It is essential to provide an analysis of your results. It is not the reader, but you, that should interpret the results. Do not assume that the reader is an expert so even result that to you seem obvious may be worth to point out. When you write a longer report, your thesis for example, the warning lights should go off when you end up with figures without any texts on several pages. In this case you have probably not provided enough analysis.

As already said but worth repeating, all figures that you have in the report must be referenced in the text. Provide captions that allows a reader to browse your paper and get the gist of your results. Summarize your findings at the end of each experiment if long.

If you have statement / hypotheses that you cannot really back up with the results put these at the end. Here you can speculate a bit and be less formal. But be clear that this is not a claim but speculation.

5 Summary and Conclusions (0.5–1 page)

Summarize what you have done and make sure that your 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 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.,] Abrate, F., Bona, B., and Indri, M. 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.