Simultaneous mapping and localization with the Sparse Extended Information Filters

Thomas Bordier, Patrick Bordes, Huiqin Chen

January 20, 2017

1 Introduction

The Simultaneous mapping and localization problem is a primordial issue in robotics. Its applications go from the self-driving car, the unmanned aerial vehicles to the new domestic robots. In this problem, a robot explores an unknown environment and tries to build a map. The robot, at each time step, is given manual commands and has a restricted vision of the environment. The main issue is that there is uncertainty on the robot position, the robot movements and the robot measures.

The classical solution of the SLAM problem uses the Kalman filter method, an algorithm that copes with statistical noise and measurement inaccuracies to estimates unknown variables (the robot and environment features positions). This method uses Bayesian inference to estimate the posterior probability distribution for the pose of the robot and for the parameters of the map. The paper we studied [1] introduces a substantial improvement of the method (sparsification) and enables to have quicker computation times.

In this paper, we will first describe the problem and its formulation in terms of Graphical Models. Then, we are going to detail the steps of the algorithm: motion update, measurement update, sparsification and state estimation. Finally, we will present our implementation and our results.

2 Sparse Extended Information Filters

The sparse extended information filters (SEIF) are based on the extended information filters(EIF) which are computationally equivalent to extended Kalman filters (EKFs). First, we describe the SLAM problem as a probability problem with the graph model HMM. In the second part, we present the solution by SEIF.

2.1 SLAM problem

The SLAM problem can be described as the problem of estimating the position of a robot and several features in an environment at the same time. The features are the landmarks in the environment, like a tree, a chair or a table. Given a sequence of controls $u^t = u_1, u_2, ...u_t$ for the robot and a sequence of measurements from robots for features $z^t = z_1, z_2, ..., z_t$ over time t, we want to estimate the real positions for robot and features at instant t, which are denoted by $\xi_t = (x_t y_1, ..., y_N)^T$. x_t present the robot position given by its two Cartesian coordinates and the robot's heading direction at instant t. We suppose that there are N features in the environment and y_i are the position for the i_{th} features given by its two Cartesian coordinates. z_t carry spatial information on the relation of the robot's pose and the location of a feature, for example, z_t might be the approximate range and bearing to a nearby feature. We assume that the position of the robot changes with time and the features don't move during the estimation. We can use a Hidden Markov graph model to express the relation between the different variables shown in figure 1. Some information of conditional independence can be deduced from this model and will be used in the "Solution " part.
The difficulty of the SLAM problem is caused by the noise for the measurement and the commands. In the view of probability, we can regard ξ_t as a Gaussian random multi-variable. The estimation of ξ_t can be transformed to estimate the expectation of the mean of ξ_t, which has the maximal value of density. The principal approach using in EKF, EIF and SEIF is to estimate the mean μ_t of the posterior $p(\xi_t|z^t, u^t)$ which is represented by a multivariate Gaussian distribution over the state ξ_t, the covariance matrix is denoted by Σ_t:

$$p(\xi_t|z^t, u^t) \propto exp\left\{-\frac{1}{2}(\xi_t - \mu_t)^T \Sigma_t^{-1}(\xi_t - \mu_t\right\} \quad (1)$$

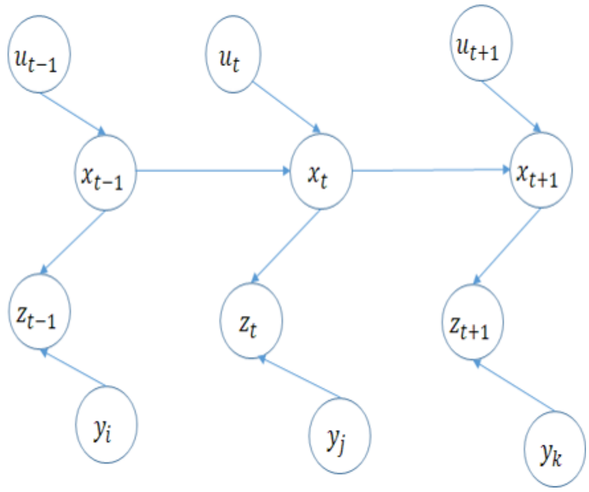

Figure 1: representation of data by HMM

By replacing the information matrix $H_t = \Sigma_t^{-1}$ and the information vector $b_t = \mu_t^T H_t$, and removing the constant normalizer, we obtain that:

$$p(\xi_t|z^t, u^t) \propto exp\left\{-\frac{1}{2}(\xi_t^T H_t \xi_t + b_t \xi_t\right\} \tag{2}$$

The goal is to compute the the expression of H_t and b_t. Once H_t and b_t are known, we can immediately obtain $\Sigma_t = H_t{-}1$ and $\mu_t = H_t^{-1} b_t^T$.

2.2 Solution of SEIF

Using the Bayes rule, equation (1) can be factorized into the following product:

$$p(\xi_t|z^t, u^t) = p(\xi_t|z_t, z^{t-1}, u^t) \propto p(z_t|\xi_t, z^{t-1,u^t})p(\xi_t|z^{t-1}, u^t) \tag{3}$$

According to figure 1,we can deduce that z_t est conditionally independent with z^{t-1} and u^t given ξ_t. So the equation (3) can be rewritten:

$$p(\xi_t|z^t, u^t) \propto p(z_t|\xi_t)p(\xi_t|z^{t-1}, u^t) \tag{4}$$

To estimate the parameters of the posterior $p(\xi_t|z^t, u^t)$, we notice that it is equivalent to estimate the parameters of $p(z_t|\xi_t)$ and $p(\xi_t|z^{t-1}, u^t)$. We refer to the computation of $p(z_t|\xi_t)$: as "measurement updates and the computation of $p(\xi_t|z^{t-1}, u^t)$ as "motion updates". The solution of SEIF consists of four steps: motion updates, measurement updates,state estimation and sparsification.

2.2.1 Motion Updates

This step concerns the computation of the parameters of $p(\xi_t|z^{t-1}, u^t)$. Since there is no new measurement, only the position of the robot is updated. The positions of the features remain unchanged. Using the Bayes rule, we have:

$$p(\xi_t|z^{t-1}, u^t) = \int p(\xi_t, \xi_{t-1}, z^{t-1}, u^t) = \int p(\xi_t|\xi_{t-1}, z^{t-1}, u_t)p(\xi_{t-1}|z^{t-1}, u^t)d\xi_{t-1} \tag{5}$$

According to figure 1, we can deduce that ξ_t is conditionally independent with z^{t-1} given ξ_{t-1} and ξ_{t-1} is conditionally independent with u_t given u^{t-1}. It yields :

$$p(\xi_t|z^{t-1}, u^t) = \int p(\xi_t|\xi_{t-1}, u_t)p(\xi_{t-1}|z^{t-1}, u^{t-1})d\xi_{t-1} \propto p(\xi_t|\xi_{t-1}, u_t) \tag{6}$$

The concern is the computation of the parameters of $p(\xi_t|\xi_{t-1}, u_t)$. To solve this problem, we need to model robot motion by non linear function with added independent Gaussian noise:

$$\xi_t = \xi_{t-1} + g(\xi_{t-1}, u_t) + S_x \delta_t \tag{7}$$

and

$$g(\xi_{t-1}, u_t) \approx g(\mu_{t-1}, u_t) + \nabla_\xi g(\mu_{t-1}, u_t)[\xi_{t-1} - \mu_{t-1}] = \hat{\Delta}_t + A_t\xi_{t-1} - A_t\mu_{t-1} \tag{8}$$

S_x is a projection matrix of the form $S_x = (I0...0)^T$ and the identity matrix I has the same dimension as the robot pose vector x_t and δ_t. δ_t is gaussian noise with mean 0 and covariance matrix U_t. Combination of equation (7) and equation (9) leads to:

$$\xi_t \approx (I + A_t)\xi_{t-1} + \hat{\Delta}_t - A_t\mu_{t-1} + S_x\delta_t \tag{9}$$

Hence, by replacing $\xi_t$and δ_t by their respective means and covariance matrix, we get:

$$\bar{\mu}_t = \mu_{t-1} + \hat{\Delta}_t \tag{10}$$

$$\bar{\Sigma}_t = (I + A_t)\Sigma_{t-1}(I + A_t)^T + S_xU_tS_x^T \tag{11}$$

The information form of $p(\xi_t|\xi_{t-1}, u_t)$ is easily recovered by

$$\bar{H}_t = \bar{\Sigma_t}^{-1} = ((I + A_t)H_{t-1}^{-1}(I + A_t)^T + S_xU_tS_x^T)^{-1} \tag{12}$$

$$\bar{b_t} = \bar{\mu_t}^T\bar{H}_t = (\mu_{t-1} + \hat{\Delta}_t)^T\bar{H}_t = (b_{t-1}H_{t-1}^{-1} + \hat{\Delta}_t^T)\bar{H}_t \tag{13}$$

So $p(\xi_t|z^{t-1}, u^t)$ can be expressed as :

$$p(\xi_t|z^{t-1}, u^t) \propto exp\left\{-\frac{1}{2}(\xi_t^T\bar{H}_t\xi_t + \bar{b}_t\xi_t\right\} \tag{14}$$

2.2.2 Measurement Updates

This step deals with the computation of the parameters of $p(z_t|\xi_t)$. Just like we did in the "motion update" part, the measurement z_t is modeled by a deterministic nonlinear function $h(\xi_t)$ with added Gaussian noise:

$$z_t = h(\xi_t) + \epsilon_t \tag{15}$$

ϵ_t is an independent noise with zero mean and covariance Z. $h(\xi_t)$ can be approximated by Taylor series expansion of h:

$$h(\xi_t) \approx h(\mu_t) + \nabla_\xi h(\mu_t)(\xi_t - \mu_t) = \hat{z}_t + C_t^T(\xi_t - \mu_t) \tag{16}$$

By combination of equation (15) and (16), we get

$$\begin{aligned} p(z_t|\xi_t) &\propto exp\left\{-\frac{1}{2}(z_t - h(\xi_t))^TZ^{-1}(z_t - h(\xi_t))\right\} \\ &= exp\left\{-\frac{1}{2}(z_t - \hat{z}_t + C_t^T(\xi_t - \mu_t))^TZ^{-1}(z_t - \hat{z}_t + C_t^T(\xi_t - \mu_t))\right\} \\ &\propto exp\left\{-\frac{1}{2}\xi_t^TC_tZ^{-1}C_t^T\xi_t + (z_t - \hat{z}_t + C_t^T\mu_t)^TZ^{-1}C_t^T\xi_t\right\} \end{aligned} \tag{17}$$

Now, we can come back to equation (4) with equation (14) and equation (17):

$$\begin{aligned} p(\xi_t|z^t, u^t) &\propto p(z_t|\xi_t)p(\xi_t|z^{t-1}, u^t) \\ &\propto exp\left\{-\frac{1}{2}\xi_t^TC_tZ^{-1}C_t^T\xi_t + (z_t - \hat{z}_t + C_t^T\mu_t)^TZ^{-1}C_t^T\xi_t\right\} \cdot exp\left\{-\frac{1}{2}(\xi_t^T\bar{H}_t\xi_t + \bar{b}_t\xi_t\right\} \\ &= exp\left\{-\frac{1}{2}(\xi_t^T(\bar{H}_t + C_tZ^{-1}C_t^T)\xi_t + (\bar{b}_t + (z_t - \hat{z}_t + C_t^T\mu_t)^TZ^{-1}C_t^T)\xi_t\right\} \end{aligned} \tag{18}$$

Thus, we can deduce that

$$H_t = \bar{H}_t + C_tZ^{-1}C_t^T \tag{19}$$

$$b_t = \bar{b}_t + (z_t - \hat{z}_t + C_t^T\mu_t)^TZ^{-1}C_t^T \tag{20}$$

So far, the solution of SLAM is obtained by equation (19) and (20). However, the computation is heavier when the number of features increases, since it involves the inversion of large matrices. To solve this problem, SEIF propose to approximate the information matrix H_t by a sparse matrix, which means only a few components are non zeros.

2.2.3 Sparsification

The goal of sparsification is to keep H_t the more sparse as possible. Here, we remove links H_{x_t,y_i} between the robot pose x_t and an individual feature y_i. The idea is to control the number of active features: if we want to delete a link between the robot and the ith feature, we set H_{x_t,y_i} to 0 and increase the pre-existing links between features.

We partition the set of features Y into three categories:

$$Y = Y^+ + Y^0 + Y^-$$

Y^+ are the features that we want to remain active, Y^- the inactive features, and Y^0 the ones where we want to set the robot-feature link to 0.

Bayes rule leads to:

$$\begin{aligned} p(x_t, Y|z^t, u^t) &= p(x_t, Y^+, Y^0, Y^-|z^t, u^t) \\ &= p(x_t|Y^+, Y^0, Y^-, z^t, u^t)p(Y^+, Y^0, Y^-|z^t, u^t) \end{aligned} \tag{21}$$

Since x_t doesn't depend on the value of the inactive features (since they haven't been seen yet), we can arbitrarily set $Y^- = 0$:

$$p(Y^+, Y^0, Y^-|z^t, u^t) = p(Y^+, Y^0, Y^- = 0|z^t, u^t)$$

We approximate the distribution p by an alternate distribution $\hat{p}$ where x_t doesn't depend on Y_0 in the first term. By applying another Bayes rule, we have:

$$\begin{aligned} \hat{p}(x_t, Y|z^t, u^t) &= p(x_t|Y^+, Y^- = 0, z^t, u^t)p(Y^0, Y^+, Y^-|z^t, u^t) \\ &= \frac{p(x_t, Y^+|Y^- = 0, z^t, u^t}{p(Y^+|Y^- = 0, z^t, u^t)}.p(Y^0, Y^+, Y^-|z^t, u^t) \end{aligned} \tag{22}$$

Since the three distributions in the expression are Gaussian, let's call H_t^1 (resp H_t^2, H_t^3) the information matrices of $p(x_t, Y^+|Y^- = 0, z^t, u^t)$ (resp. $p(Y^+|Y^- = 0, z^t, u^t)$ and $p(Y^0, Y^+, Y^-|z^t, u^t)$). We won't detail the computation of these matrices. At the end of the day, the sparsified matrix $\hat{H}_t$ can be computed easily:

$$\hat{H}_t = H_t^1 - H_t^2 + H_t^3$$

The information vector is computed this way:

$$\begin{aligned} \hat{b}_t &= \mu_t^T \hat{H}_t \\ &= \mu_t^T (H_t - H_t + \hat{H}_t) \\ &= \mu_t^T H_t + \mu_t^T (\hat{H}_t - H_t) \\ &= b_t + \mu_t^T (\hat{H}_t - H_t) \end{aligned} \tag{23}$$

2.2.4 State Estimation

The last step is to estimate the state μ_t from the information form H_t and b_t. Instead of recovering μ_t by $\mu_t = H_t^{-1} b_t^T$ which asks for the inversion of the whole information matrix, it can be describes as a optimization problem:

$$\begin{aligned} \mu_t &= argmax_{\xi_t} p(\xi_t|z^t, u^t) = argmax_{\xi_t} C \cdot exp\left\{ -\frac{1}{2}\xi_t^T H_t \xi_t + b_t^T \xi_t \right\} \\ &= argmin_{\xi_t} \frac{1}{2}\xi_t^T H_t \xi_t - b_t^T \xi_t \\ &= argmin_{\xi_t = (\xi_{1,t}, \xi_{2,t}, \ldots \xi_{i,t} \ldots)^T} \frac{1}{2} \sum_i \sum_j \xi_{i,t}^T H_{i,j,t} \xi_{j,t} - \sum_i b_{i,t}^T \xi_{i,t} \end{aligned} \tag{24}$$

We can handle it independently since the problem is separable. For each component $\xi_{i,t}$, the optimal value is obtained by setting the derivation with respect to this component:

$$\frac{\partial}{\partial \xi_{i,t}} \left\{ \frac{1}{2} \sum_i \sum_j \xi_{i,t}^T H_{i,j,t} \xi_{j,t} - \sum_i b_{i,t}^T \xi_{i,t} \right\} = \sum_j H_{i,j,t} \xi_{j,t} - b_{i,t}^T = 0 \tag{25}$$

This leads to

$$\xi_{i,t}^{[k+1]} = H_{i,i,t}^{-1}[b_{i,t}^T - \sum_{j \neq i} \xi_{j,t}^{[k]}] \tag{26}$$

Compared to the inversion of the whole information matrix, equation (26) only need to compute the inversion of a single value $H_{i,i,t}$.

3 Experiments Results

3.1 Dataset

The experiments are led on simulated data and a robot simulator was implemented. The landmark generator takes an integer L for entry and builds landmarks in a square shape with $N = L \times L$ landmarks.

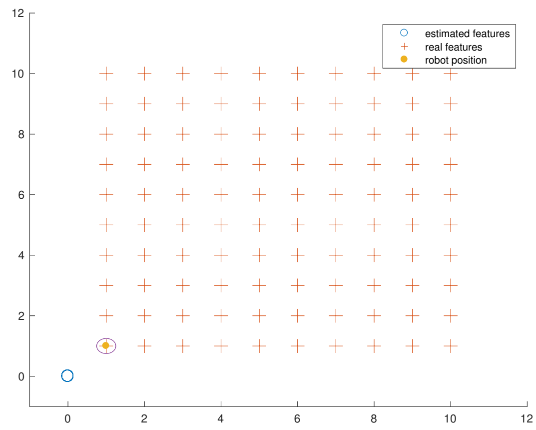

Figure 2: Initial state of the experiment with $L = 10$, e.g. 100 landmarks on a grid

3.2 Protocol of experiments

The algorithm proposed in the article which we have given the details above, was implemented in a way that allows the robot to identify which landmark it is measuring. This situation is the one where the robot sees a feature with a unique number on it. We chose to not tackle this issue in our implementation. The code for our experiments is available here: SLAM.

The performance of the SEIF is evaluated by the cumulative error between the true position of the landmarks and the predicted landmarks with the implemented algorithm.

$$E(x_{prediction}, x_{truth}) = \sum_{i=1}^{L \times L} (x^i_{prediction} - x^i_{truth})^2$$

The hyperparameters, they are set as the following:

- Motion noise variance 10^{-5}
- Measurement noise variance 10^{-3}
- Maximum sensor range for robot 0.5
- The number of landmarks $L \times L$ will vary
- The commands of the robot are a periodic movement visiting the 9 first landmarks.

3.3 Results

First the influence of hyperparameters was tested in order to confirm the intuition we had regarding their contribution to performance. By augmenting the movement noise covariance, we can witness an increase in error, which confirms the idea that the error accumulates over time.

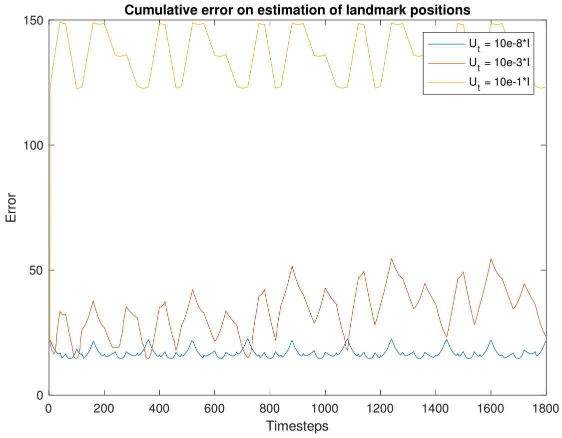

Figure 3: Variation of the noise associated to the robot's movement for a periodical movement on the grid. U_t is the covariance matrix associated with the motion noise.

With the parameters stated in the previous part, we obtain the following map estimation (blue circles are the estimated landmarks). We have noticed a great influence of the choice of range for the robot.

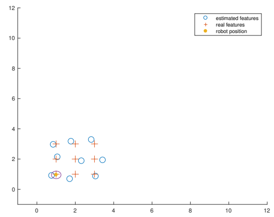

Figure 4: Results for a map with 9 features and a robot which moves near each feature

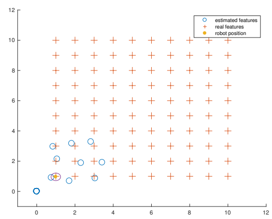

Figure 5: Results for a map with 100 features and a robot which moves near the 9 first ones

The main improvement of the SEIF is the computing time and the memory gains. We measured the difference between EKF and SEIF for an increasing number of landmarks. This table was computed by running several 10 times each setting and dividing by the number of iterations of the algorithm.

Landmarks	**100**	**225**	**400**
With Sparsification	0,0199	0,0725	0,264
No Sparsification	0,0213	0,0681	0,255

Table 1: Computation evaluation, the values are in seconds/iteration

We notice no major improvement when attempting to compare the computation time for every iteration. This can be explained by the fact that we do not use our implementation of state estimation presented in the article because we did not manage for it to work. Instead we compute directly the inverse, which is the time-consuming part of the algorithm EKF.

Conlusion Despite the fact that we did not manage to make the sparse extended information filter work end-to-end because of a missing block, we did manage to implement and test the extended Kalman filter.

References

[1] Sebastian Thrun, Yufeng Liu, Zoubin Ghahramani, Daphne Koller, Andrew Y. Ng, Hugh Durrant-Whyte *Simultaneous mapping and localization with the Sparse Extended Information Filters*

[2] Cyril Stachniss *Sparse Extended Information Filter for SLAM* Uni Freiburg.