An Incremental SLAM Algorithm with Inter-calibration between State Estimation and Data Association

Xiucai Ji, Hui Zhang, Dan Hai, and Zhiqiang Zheng

College of Mechatronics Engineering and Automation,
National University of Defense Technology, 410073 Changsha, China
{jxc_nudt,zhanghui_nudt,haidan,zqzheng}@nudt.edu.cn

Abstract. In most SLAM (Simultaneous Localization and Mapping) approaches, there is only unilateral data stream from data association (DA) to state estimation (SE), and the SE model estimates states according to the results of DA. This paper focuses on the reciprocity between DA and SE, and an incremental algorithm with inter-calibration between SE and DA is presented. Our approach uses a tree model called *correspondence tree* (CT) to represent the solution space of data association. CT is layered according to time steps and every node in it is a data association hypothesis for all the measurements gotten at the same time step. A best-first search with limited back-tracking is designed to find the optimal path in CT, and a state estimation approach based on the least-squares method is used to compute the cost of nodes in CT and update state estimation incrementally, so direct feedback is introduced from the SE to DA. With the interaction between DA and SE, and combining with tree pruning techniques, our approach can get accurate data association and state estimation for on-line SLAM applications.

Keywords: SLAM, data association, state estimation, least-squares, backtracking search.

1 Introduction

The simultaneous localization and mapping (SLAM) problem requires an autonomous robot, moving in an unknown environment, to incrementally derive a map of the environment only from its relative observations of the environment, and then simultaneously determine its own position in the map. A solution to SLAM has been seen as a 'holy grail' for mobile robotics community as it provides a mean to make a robot truly autonomous [1-3].

The feature-based SLAM problem is consisted of a continuous and a discrete component [4]. The continuous component is the *state estimation* (SE) problem, which estimates the location of individual features in the environment and the position the robot relative to these features. The discrete part is the *data association* (DA) problem [5], which is the problem of determining the correspondence relationships between observed features and landmarks in the map built thus far. Data association is very

L. Iocchi et al. (Eds.): RoboCup 2008, LNAI 5399, pp. 97–108, 2009.

important to the consistent map construction in SLAM, since incorrect data association may make the state estimation divergent, and cause the entire SLAM process fail.

Some approaches have been put forward to solve the data association problem in SLAM, and most of them are classical methods for tracking [5, 6]. The *nearest neighbor* (NN) algorithm is the simplest [7]. But in complex environment with cluttered features, it may accept wrong data association hypotheses, and cause the state estimation divergent. Many methods such as the joint compatibility test (JCBB) [8], the integer programming method [9] and the graph theoretic approach [10] have been presented to overcome this shortcoming. These methods make incremental maximum likelihood decisions for all the features gotten at the same time step, and could get the local optimal data association. More sophisticated algorithms called the multiple-hypothesis method, such as the multi-hypothesis Kalman filter [2] and FastSLAM [11], generate many data association hypotheses, and later determine which is the best as more observes arrive. These algorithms are more robust, but need more computational overhead. Reference [4] presented a lazy data association algorithm, which searches the interpretation tree with backtracking to revise past data association. This method can find the global optimal data association, but it need calculate the inverse of high-dimensional matrixes.

All above algorithms lose sight of the interaction between data association and state estimation. There is only data flow from the data association model to the state estimation process, and the state estimation process has no direct influence to the he data association model. In this paper we focus on the inter-calibration between data association and states estimation, and introduce direct feedback from the state estimation process to the data association model. The data association problem is modeled by a tree structure, called *correspondence tree* (CT), and nodes in it are features extracted at-a-time with a data association hypothesis. An incremental state estimating approach based on the least-squares method is used, and its least-squares residual is used as the cost of a node in CT. A best-first backtracking search strategy to CT is designed to search the best data association. Because the state estimation process provides feedback to the data association model, and at the same time past data association can be revised with backtracking, it's guaranteed that the global optimal data association is found. Additionally, since only a path of CT is expanded at most time, our method has moderate computational cost.

2 Formulation of SLAM

In this section we present the formulation of the SLAM problem with unknown data association. This paper uses the same mathematical framework and notations as that adopted in [7]. We are interested in the maximum a posterior (MAP) estimation for the entire robot trajectory $X \triangleq \{x_i\}, i \in 1 \ldots M$ and the map of landmarks $L \triangleq \{l_j\}$, $j \in 1 \ldots N$, given the measurements $Z \triangleq \{z_k\}$, $k \in 1 \ldots K$ and the control inputs, $U \triangleq \{u_i\}$, with unknown data association $C \triangleq \{c_k\}$. If $c_k = n$, the measurement z_k corresponds to landmark l_n. Let $\Theta \triangleq (X, L)$, and then the SLAM problem with unknown data association can be formulated as the following optimization problem:

$$\left\{\Theta^*, C^*\right\} \triangleq \arg\max_{\Theta,C} P(\Theta \mid Z,U,C) = \arg\max_{X,L,C} P(X,L \mid Z,U,C). \tag{1}$$

In (1), the solution space of C is discrete, while ’s is continuous. Direct method solving (1) is to solve the sub-problem max$P($ |$Z,U,C)$ for every value of C by traversing the whole solution space of C. This method is impractical, because the solution space of C will grow exponentially with the increas of measurements and landmarks. Most approaches solve (1) in two steps. Firstly, calculating a sub-optimal value of C by solving the following equation:

$$C^* \approx \arg\max_{C} P(\Theta \mid Z,U,C). \tag{2}$$

This is the *data association* problem. And then a *state estimation* problem with known data association is solved:

$$\Theta^* = \arg\max_{\Theta} P(\Theta \mid Z,U,C^*). \tag{3}$$

3 Data Association

3.1 Correspondence Tree

Every value of C represents a correspondence relationship between Z and L. Let c^i denotes the data association of the measurements z^i gotten at time step i, and $C^t \triangleq \{c^i\}$, $Z^t \triangleq \{z^i\}$, i=1…t. The solution space of C^t can been represented as a *correspondence tree* (CT) of t levels, as shown in Fig.1. The tree has a *null* root, level i

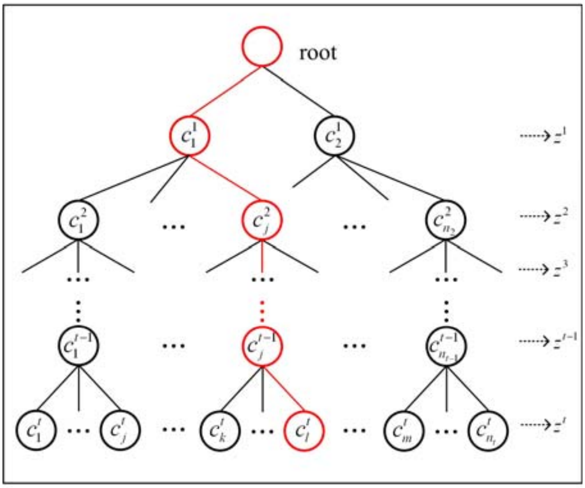


Fig. 1. An example of correspondence tree. c_j^i is the jth possible data association for the observations z^i at time step i, and n_i is the number of possible data associations for the observations z^i.

corresponds to measurement z^i and every node c^i corresponds to a data association hypothesis of z^i then a path from the root to a node in level i corresponds to a data association hypothesis of Z^i as shown by red lines in Fig.1.

The CT tree is layered according to time steps, so it is more preferable than the interpretation tree [12] to represent the data association problem in SLAM. The incremental maximum likelihood data association approaches, such as the joint compatibility test [8] and the graph theoretic approach [10], choose the best data association of z^t based on the data association of z^{t-1} and then freeze it forever, that is

$$\hat{c}^t = \arg\max_{c^t} P(\Theta \mid Z, U, \hat{C}^{t-1}, c^t). \tag{4}$$

In nature, these methods are *hill-climbing* searching to CT, which is a greedy tree searching method, and can only find a local optimal solution. Multiple-hypothesis methods actually are *breadth-first* searching to CT, so they could find the global optimal solution. But because these methods need to expand and save the whole or most parts of CT, their computation complex is very high.

3.2 Limited Backtracking Data Association

From above analysis we know that tree searching methods can be used to solve the data association problem. We take

$$\Gamma(c^t) = \min - \log P(\Theta \mid Z, U, C^t) \tag{5}$$

as the evaluation function to calculate the cost of node c^t. In fact, this function calculates the cost of path C^t to which c^t belongs. The cost presents the desirability of expanding node c^t, and the node with the least cost will be expanded first. Fig. 2 illustrates the basic idea of our searching strategy: our approach maintains an H-layered path above a

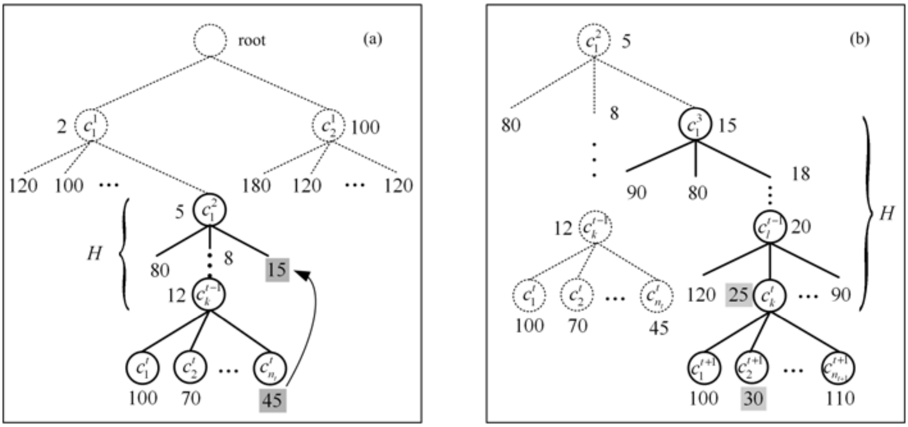


Fig. 2. A example of best-first with limit backtracking search. (a) At time step t, a backtracking appears; (b) When the node with least cost is in level t, the searching at time step t stops, and the H-layered path (*real line*) is saved. At time step t+1, the node with least cost is in level t is expand first.

node in the deepest level at the moment, at the same time the entire frontier of the path is also saved. At time step *t*, after new measurements z^t arrives, the node with the least cost in level *t*-1 will be expanded first to produce all of its children nodes, all children nodes are assessed and their costs are saved. Then all frontier nodes in the *H*-layered path are compared: if the frontier node with the least cost is among those new children nodes, the searching stops, the children node with the least cost will be expanded first at the time step $t+1$, the new *H*-layered path with its frontier are memorized and other nodes in CT are abandoned (dashed nodes in Fig. 2); If a frontier node not in level *t* has a lower cost than the least one of those children nodes, the algorithm will backtrack to this frontier node and expand it, and such procedure will go on until the frontier node with least cost is in level *t*. Then the frontier node will be expanded first at the time step $t+1$, the new *H*-layered path its frontier above the node are memorized, and other nodes in CT are also abandoned.

The reasons why a path of *H* layer is kept are that:

- Since the cost of nodes increase monotonically with the depth of CT, if we maintain the whole optimal path, when CT is expanded deep enough, it's possible that the cost of an early frontier node is lower than the current frontier node of the optimal path, and then meaningless backtracking will appear;
- It's shown that good data association for a measurement can be obtained in a few time steps, and in some circumstance it only needs 2 time steps to get a near optimum solution [6], so we needn't keep the whole path, and for most circumstance, *H*=5 is enough.

In fact, our approach is a best-first search with limit backtracking. It's well known that in a limited tree, if the cost of nodes increase monotonically with the depth of the path which it belongs to, best-first searching with backtracking is certain to find the global optimal path. In addition, our searching method needn't expand the whole CT, and only needs to keep one path at most time, so it has moderate computational cost.

4 SLAM as a Least-Squares Problem

According to the Bayesian networks model, the SLAM problem with known data association $C^{'}$ can be formulated by the following equation [12]:

$$P(X,L,Z,U \mid C') = P(x_0)\prod_{i=1}^{M} P(x_i \mid x_{i-1},u_i)\prod_{k=1}^{K} P(z_k \mid x_{i_k},l_{c_k}), \tag{6}$$

where $P(x_0)$ is a prior on the initial state, $P(x_i|x_{i-1},u_i)$ is the robot *motion model*, and $P(z_k \mid x_{i_k},l_{c_k})$ is the *landmark measurement model*. The MAP estimate Θ^* can be obtained by minimizing the negative log of the joint probability of (6):

$$\Theta^* = \arg\max_{\Theta} P(X,L \mid Z,U,C') = \arg\min_{\Theta} -\log P(X,L,Z,U \mid C'). \tag{7}$$

It's assumed that the motion and measurement noises are Gaussian. The motion model is defined by $x_i = f_i(x_{i-1},u_i)+\omega_i$, where $\omega_i \sim N(0,\Lambda_i)$. The landmark measurement model is defined by $z_k = h_k(x_{i_k}, l_{j_k}) + \upsilon_k$, where $\upsilon_k \sim N(0, \Sigma_k)$. Combining them with (7), leads to the following least-squares problem:

$$\Theta^* = \arg\min_{\Theta} \left\{ \sum_{i=1}^{M} \| x_i - f_i(x_{i-1}, u_i) \|_{\Lambda_i}^2 + \sum_{k=1}^{K} \| z_k - h_k(x_{i_k}, l_{j_k}) \|_{\Sigma_k}^2 \right\} \tag{8}$$

where

$$\| e \|_{\Sigma}^2 \triangleq e^T \Sigma^{-1} e = (\Sigma^{-T/2} e)^T (\Sigma^{-T/2} e) = \| \Sigma^{-T/2} e \|_2^2 \tag{9}$$

is squared Mahalanobis distance given a covariance matrix Σ.

Linearizing the non-linear items in (8), we can obtain the following linear problem:

$$\delta\Theta^* = \arg\min_{\Theta} \left\{ \sum_{i=1}^{M} \| F_i^{i-1} \delta x_{i-1} + G_i^i \delta x_i - a_i \|_{\Lambda_i}^2 + \sum_{k=1}^{K} \| H_k^{i_k} \delta x_{i_k} + J_k^{j_k} \delta l_{j_k} - c_k \|_{\Sigma_k}^2 \right\}, \tag{10}$$

where F_i^{i-1}, $H_k^{i_k}$, and $J_k^{j_k}$ are Jacobians of f_i and h_k, $G_i^i = I$, and a_i and c_k are odometry and observation measurement prediction error. According to (9), we can eliminate Λ_i and Σ_k from (10), and obtain a standard linear least-squares problem:

$$\theta^* = \arg\min_{\theta} \| A\theta - b \|_2^2 . \tag{11}$$

Please see [12] and [13] for more detail derivation.

This standard least-squares problem can be solved by the QR matrix factorization of the Jocobian $A \in \mathbb{R}^{m \times n}$:

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix}, \tag{12}$$

where $Q \in \mathbb{R}^{m \times m}$ is orthogonal, and $R \in \mathbb{R}^{n \times n}$ is upper triangular. Multiplying with the orthogonal matrix Q^T doesn't change the norm:

$$\| A\theta - b \|_2^2 = \| Q^T A\theta - Q^T b \|_2^2 = \| R\theta - d \|_2^2 + \| e \|_2^2, \tag{13}$$

so the solution of $R\theta = d$ is the least-squares solution of (11), where $[d, e]^T \triangleq Q^T b$, and $\| e \|_2^2$ is the residual of the least-squares problem (11):

$$\| e \|_2^2 = \min \| A\theta - b \|_2^2 = \min - \log P(\Theta \mid Z, U, C'). \tag{14}$$

According to (5), we take $E \triangleq \| e \|_2^2$ as the cost of the final node of the path C'. So there will be direct interaction and reciprocal promotion between data association and state estimation: the state estimation provides direct feedback to select the best data association, and the best data association will lead to more precise state estimation.

5 Incremental SLAM Algorithm

For real-time application, it needs the SLAM algorithm rapidly calculate the costs of nodes i.e. the residual of the least-squares problem (11). In this section, we present such an incremental SLAM algorithm.

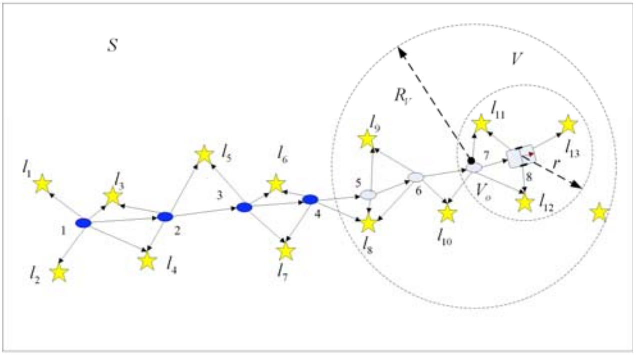


Fig. 3. The illustration of variables set V and S. Vo is the center of V, R_V is its radius, and r is the maximum perception radius of sensors.

5.1 Variable Ordering

We divide the variables in $\Theta=\{X, L\}$ into a static set S and a variable set V:

$$\Theta = S \cup V \tag{15}$$

The set V includes all variables that close to the robot's current position. For example, in Fig. 3, $V=\{l_8, l_9, l_{10}, l_{11}, l_{12}, l_{13}, x_5, x_6, x_7, x_8\}$, and S={ l_1, l_2, l_3, l_4, l_5, l_6, l_7,x_1, x_2, x_3, x_4}. Let V_o denote the center of V, which is one of the robot positions on its trajectory, and R_V denote the radius of V. When the robot's current position is far away from V_o:

$$\| \hat{x}_t - V_o \| > R_V - r \ , \tag{16}$$

we set $\hat{x}_t$ as the new center of V, where r is the maximum perception radius of sensors. All variables whose distance from the new V_o is smaller than R_V will compose the new V, and other variables left compose S. In our approach, R_V is chosen carefully so that V includes all variables related to the saved H-layered path of CT up to current time step.

For the SLAM problem, the Jacobian matrix A in (11) is sparse [12]. The QR factorization of A will lead to non-zero fill-ins for matrix R. However, such fill-ins can be avoided through column reordering of A. Since finding an optimal ordering is NP-hard, various approximate algorithms have been developed, and among them the *approximate minimum degree* (AMD) works well [14]. In our algorithm, A is ordered in the following way:

- S is put in the front of A and V follows:

$$A = \left[A_S \quad A_V \right], \tag{17}$$

- Every time V_o changes, we use AMD to column order A_S and A_V for the new S and V.
- At time steps that V_o isn't changed, the current pose and new landmarks are put into V, and ordered at the end of A_V. The current pose is in front of new landmarks.
- We row order A according to time, and early equations are ordered on the top. At every time steps, the motion model equations are ordered ahead of measure-ments and old landmarks' measurements are ordered ahead of those of new landmarks.

Fig. 4 depicts the row and column ordering of the Jacobian matrix A in the SLAM scene shown in Fig. 3, and the side number is the row order of A, the 16th and 17th row are the equations relating to variables in V and S, and are the links between V and S.

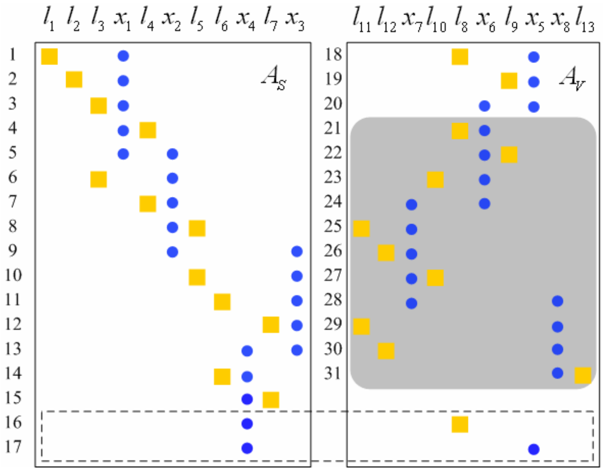

l_1 l_2 l_3 x_1 l_4 x_2 l_5 l_6 x_4 l_7 x_3
l_{11} l_{12} x_7 l_{10} l_8 x_6 l_9 x_5 x_8 l_{13}
A_S
A_V
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

Fig. 4. The row and column ordering of the Jacobian matrix A in the SLAM scene shown in Fig. 3. On the top is the column order of variables and the side number is the row order. The 16th and 17th row are the equations relating to variables in V and S. The shadowed part is related to the saved H-layered path of CT.

The above column ordering will reduce fill-ins to R, and the row ordering will facilitate the backtracking search of our data association algorithm. Because we row order A according to time steps, equations that related to the saved H-layered path of CT is at the bottom of A. For example, the shadowed part in Fig. 4 is related to a 3-layered path.

5.2 Incremental State Estimation

Since V includes all variables related to the saved H-layered path of CT up to current time step, the calculation of the costs of nodes in the backtracking search to CT only relates to variables inside V. Assuming that the optimal data association up to time step t has been found, and after QR factorization, the equations only including variables in V is $R_V^t \delta V^t = d_V^t$, with a least-squares residual $E^t = \| e^t \|_2^2$, i.e. the cost of the last node in the saved optimal path of CT, where $R_V^t = Q_V^t A_V^t$.

At time step t+1, the last node of the saved optimal path is first expanded according to z^{t+1}'s association hypotheses. Let $c^{'}$ be one children nodes. According to $c^{'}$, we divide z^{t+1} into two parts: measurements from old landmarks and others from new landmarks. Following the variables ordering rules in section 5.1, we can get the following equations:

$$\begin{pmatrix} R_V^t & 0 & 0 \\ F & G & 0 \\ J_{old} & H_{old} & 0 \\ 0 & H_{new} & J_{new} \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta V^t \\ \delta x_{t+1} \\ \delta L_{new} \end{pmatrix} = \begin{pmatrix} d_V^t \\ b_x \\ b_{Lold} \\ b_{Lnew} \\ e^t \end{pmatrix}. \tag{18}$$

We denote (18) as $A_V^{t+1}\delta V^{t+1}=b^{t+1}$. It's easy to verify that new landmark measurements have no influence to the cost of $c^{'}$. The QR factorization of the coefficient matrix in (18) can be obtained by eliminating lower diagonal non-zeroes of *F*, *G*, J_{old} and H_{old} row-by-row through Givens rotations [13], and then we will get

$$\begin{pmatrix} \bar{R}_V^t & T \\ 0 & \bar{R}_x \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \delta V^t \\ \delta x_{t+1} \end{pmatrix}=\begin{pmatrix} \bar{d}_V^t \\ \bar{d}_x \\ e^{t+1} \end{pmatrix} \tag{19}$$

Thus the least-squares residual of (18) is

$$E^{t+1}=\| e^t \|_2^2+\| e_{t+1} \|_2^2=E^t+\| e_{t+1} \|_2^2 . \tag{20}$$

Apparently $E^{t+1}\geq E^t$, and the costs of nodes in CT increase monotonically, so our algorithm can find the global optimal path of CT. Equation (20) shows that the cost of children nodes can be incrementally computed from its parent's cost. After the optimal node in level *t*+1is founded, the estimation update of variables in *V* can be obtained by back-substitution [13].

Above computation is enclosed in *V*, so our algorithm has $O(n^2)$ time complex, where *n* is the size of *V*. Since the size of *V* is limited and relatively small, our algorithm has a moderate computation cost.

5.3 Pruning

In our approach the following techniques are adapted to prune those branches in CT with little chance to be expanded:

- A range *W* around the current robot position is set. Landmarks out of the range will not be considered in the expanding of nodes in CT. *W* is decided by $R_W= r+\rho$, where R_W is the radius of *W*, *r* is sensors' maximum perception radius, and $\rho>0$ is a compensator for errors.
- A gate is applied to every landmark in *W*. Only measurements that are close enough to the prediction position of a landmark are considered as possible candidates to be associated with the landmark. The gate is given by [9]:

$$\tau_{jk}=\upsilon_{jk}^T s_j^{-1}\upsilon_{jk}<\varepsilon\, , j=1\ldots N; k=1\ldots K, \tag{21}$$

  where $\upsilon_{jk}=z_k-\hat{z}_j$ is the innovation and s_j is it's covariance. τ_{jk} is a squared Mahalanobis distance between z_k and $\hat{z}_j$, and follows the χ^2 distribution. If a measurement is out of the gate of all landmarks, it matches a new landmark.
- For a new landmark, a punishment ε i.e. the gate in (21) is added to the cost of its corresponding node in CT, since new landmarks will give zero residual.

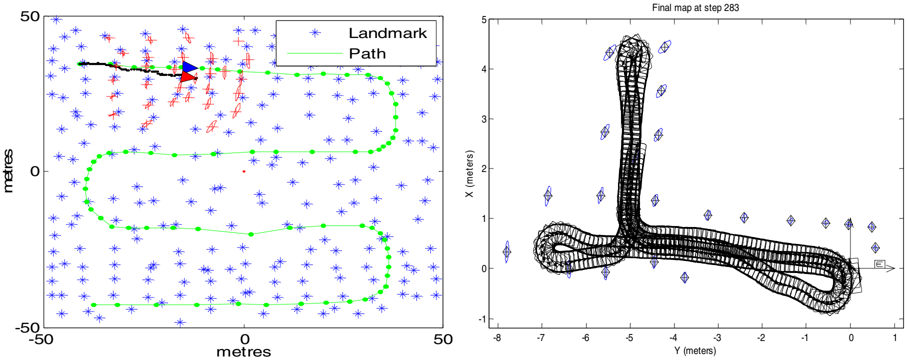
During the pruning procedure, covariance matrices of variables are needed. In our approach, the efficient method presented in [14] is applied.

6 Experiment Results

The algorithm presented in this paper was tested with simulation experiments and a real robot experiment dataset. In these experiments, we have compared our algorithm with EKF-based SLAM [15] using the *nearest neighbor* and the *joint compatibility branch and bound* (JCBB)[8] data association. The experiment results have shown that our algorithm is more accurate.

6.1 Simulation

In order to evaluate our approach, a simulation environment with about 200 landmarks has been created as shown in Fig. 5(a), and significant motion and measurement noise have been added.



(a)

(b)

Fig. 5. (a). A typical simulation experiment scene. (b). The estimated robot trajectory and 2-sigma ellipses of landmark locations for the 'sjursflat' dataset with our algorithm.

For 20 times we applied EKF-based SLAM with NN and JCBB data association and our algorithm to perform the SLAM process respectively. Every time, we changed the landmarks positions randomly and some landmarks were significantly close to each other. In the experiments, we saved a 5-layered path of the CT tree, and the gate for pruning was $\varepsilon = 6.63$ with a probability of 0.99 for a measurement to match a close landmark. The correct data association rates of every algorithm are shown in Table. 1. Our algorithm had finished all the experiments, and the correct data association rate was 96.8%. Backtracking search had occurred for 157 times. The wrong data associations appeared to landmarks far from the robots, and only observed for one or two times with high measurement noise. The correct data association rate for the EKF-based SLAM with NN was 79.6%. When landmarks were too closed, the NN algorithm failed. The correct data association rate for the EKF-based SLAM with JCBB was 85.3%, and when landmarks were too closed, it worked better than the NN algorithm, but when there is large motion error and landmarks have similar deployments it may accept wrong data association. But our algorithm will accept such wrong data association temporally, and the revise it after several time steps.

Table 1. Comparison of correct data association rate (CDAT) of different algorithm

	Our algorithm	NN	JCBB
CDAT(%)	96.8	79.6	85.3

6.2 Sjursflat Dataset

The ‘sjursflat’ dataset shows the ground floor of Sjur Vestli’s house, recording using a SmartRob 2 with a forward looking SICK LMS200, which has been distributed with ‘the CAS Robot Navigation Toolbox’ [16].

In our experiment, the same parameters were used as simulations, and a beacon-based slam was done. There were 283 time steps and 20 landmarks were estimated. Our algorithm had gotten similar results as the EKF-based method in [16], and Fig. 6 shows the resulting trajectory and map. In this experiment, our data association algorithm had degenerated as an incremental maximum likelihood method, and no backtracking had appeared, because landmarks were not close to each other, but our algorithm had matched all the 615 measurements, while the NN algorithm matched 609 measurements, for ambiguous associations were ignored.

7 Conclusion

The SLAM problem includes two parts: data association and state estimation. Former algorithms deal with them separately. This paper presents a new on-line algorithm for the SLAM problem. The characteristic of our approach is that there is direct interaction between data association and state estimation. The state estimation process evaluates every data association hypothesis, and provides direct feedback to the data association model. At the same time the data association model can revise incorrect former data association by backtracking the correspondence tree when a potential better state estimation is proposed. We have evaluated our algorithm by simulations and a real robot dataset, and compared it with the classical methods. Experiment results have validated the high accuracy of our algorithm.

We have just essentially invalidated our approach. In the future work, we will try to use this method in SLAM for the RoboCup Rescue Competition.

References

1. Frese, U.: A Discussion of Simultaneous Localization and Mapping. Autonomous Robots 20(1), 25–42 (2006)
2. Leonard, J., Durrant-Whyte, H.: Dynamic Map Building for an Autonomous Mobile Robot. The International Journal on Robotics Research 11(4), 286–298 (1992)
3. Cheeseman, P., Smith, R., Self, M.: A Stochastic Map for Uncertain Spatial Relationships. In: 4th International Symposium on Robotic Research, pp. 467–474. MIT Press, Cambridge (1987)
4. Häehnel, D., Burgard, W., Wegbreit, B., Thrun, S.: Toward Lazy Data Association in SLAM. Intl. J. of Robotics Research, 421–431 (2003)

5. Bar-Shalomand, Y., Fortmann, T.E.: Tracking and Data Association. Academic Press, San Diego (1988)
6. Cox, I.: A Review of Statistical Data Association Techniques for Motion Correspondence. Int'l. J. Computer Vision. 10, 53–65 (1993)
7. Guivant, J., Nebot, E.M.: Optimization of the Simultaneous Localization and Map Building Algorithm for Real Time Implementation. IEEE Trans. Robot. Automat. 17, 242–257 (2001)
8. Neira, J., Tardos, J.D.: Data Association in Atochastic Mapping Using the Joint Compatibility Test. IEEE Trans. Robot. Automat. 17, 890–897 (2001)
9. Zhang, S., Xie, L., Adams, M.D.: An Efficient Data Association Approach to Simultaneous Localization and Map Building. Int'l. J. Robotics Research. 24, 49–60 (2005)
10. Bailey, T., Nebot, E.M., Rosenblatt, J.K., Durrant-Whyte, H.F.: Data association for Mobile Robot Navigation: a Graph Theoretic Approach. In: IEEE Int. Conf. Robot. Automat, pp. 2512–2517. IEEE Press, San Francisco (2000)
11. Montemerlo, M., Thrun, S.: Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM. In: Proc. IEEE Int. Conf. Robotics Automation, pp. 1985–1991. IEEE Press, Los Alamitos (2003)
12. Dellaert, F., Kaess, M.: Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. Technical Report, Georgia Tech. (2005)
13. Kaess, M., Ranganathan, A., Dellaert, F.: Fast Incremental Square Root Information Smoothing. In: Intl. Joint Conf. on Artificial Intelligence, pp. 2129–2134, Hyderabad (2007)
14. Kaess, M., Ranganathan, A., Dellaert, F.: iSAM: Fast Incremental Smoothing and Mapping with Efficient Data Association. In: IEEE Intl. Conf. on Robotics and Automation, pp. 1670–1677. IEEE Press, Rome (2007)
15. Leonard, J., Cox, I., Durrant-Whyte, H.F.: Dynamic Map Building for an Autonomous Mobile Robot. Intl. J. of Robotics Research. 11, 286–289 (1992)
16. Kai, O.: Arras.: The CAS Robot Navigation Toolbox, `http://www.cas.kth.se/toolbox`