# Towards a Consistent SLAM Algorithm using B-Splines to Represent Environments

Minjie Liu, Shoudong Huang, $Member, IEEE$, Gamini Dissanayake, $Member, IEEE$
and Sarath Kodagoda

*Abstract*— This paper presents a statistically consistent SLAM algorithm where the environment is represented using a collection of B-Splines. The use of B-Splines allow environment to be represented without having to extract specific geometric features such as lines or points. Our previous work proposed a new observation model that enables raw measurements taken from a laser range finder to be transferred into relative position information between the control points of a B-Spline and the robot pose where the observation is made. One of the unresolved issues in the work was the estimation of the observation covariance, which is addressed through an analytical approach in this paper. As the uncertainty associated with the observation model is accurately defined, and an optimization approach is used in the estimation process, the proposed SLAM algorithm can produce consistent estimates. Both simulation and experimental data are used for evaluation of the results.

## I. INTRODUCTION

Consistency is an important issue in estimation techniques and algorithms. An estimator is called consistent if the estimate is unbiased and the covariance calculated by the estimator matches the actual mean square error [1].

As an estimation problem, simultaneous localization and mapping (SLAM) also faces the same issue [2]. For point feature based SLAM problem, it is recognized that the Extended Kalman Filter (EKF) SLAM algorithm may produce inconsistent estimates under various conditions. Large orientation error of the robot leads to inconsistencies [2][3]. Recently it has also been shown that the major cause of inconsistency of a SLAM algorithm is due to the fact that the Jacobian with respect to the same feature/robot pose is evaluated at different estimate values [3][4]. Results also show that SLAM algorithms using optimization techniques can improve the consistency significantly as Jacobians are re-evaluated over different times [5][6].

Alternative method to solve the SLAM problem is "trajectory based SLAM" [7][8]. In this method relative pose information between consecutive scan frames are computed using scan matching techniques and then an optimization is performed to smooth the whole robot trajectory. Many promising results have been achieved in this direction. The consistency of the robot trajectory can be checked provided that the ground truth of the robot trajectory is available [9].

However it is impossible to evaluate the quality of the map due to the fact that, this method does not provide a model to represent the environment.

A number of research groups have tried to use more complex geometric primitives to represent the environment. The Symmetries and Perturbations Model (SP-model) was introduced in [10]. Some successful applications of SP-model based SLAM have been reported. However, it is not easy to use SP-model to represent complex features in the environment. Furthermore, it has been demonstrated in [11] that SP-model leads to inconsistent result in indoor environment.

Nieto et al. [12] proposed the Scan SLAM algorithm, which is a marriage of EKF-SLAM and scan correlation. In Scan SLAM, landmarks are no longer defined by analytical models, instead they are defined by templates composed of raw sensor data. Although these templates can be augmented as more data becomes available, the templates themselves are not included in the EKF state vector. Hence the uncertainty of the templates can not be represented.

Very recently, Pedraza et al. developed the BS-SLAM [13] where B-Splines are used to represent the environment. Using B-Splines to represent the environment has some clear advantages: 1) B-Splines are able to represent complex environment. 2) some appealing properties for B-Splines e.g. any affine transformation can be applied to the curve by applying transformation to the control points. However, in [13], it has been observed that the linearization process of EKF lead to inconsistent estimate in some scenarios.

In [14] we proposed a new observation model for B-Spline SLAM. The observation model is expressed as a function of relative positions between the control points of the observed spline and the observation point. With the new observation model, the B-Spline SLAM problem can be transferred into a point-feature based SLAM problem and can be solved by optimization-based point-feature SLAM algorithms. In [14], the error of the control points has been considered from two independent sources: Spline fitting error and the chord length error. The chord length error is derived empirically. This makes the proposed algorithm less robust as different sensor arrangement would result in different chord length error.

This paper shows how to analytically derive the covariance matrix of the estimated control points. It considers that the spline fitting error and the chord length error are not independent. Simulation and experimental results are presented to demonstrate the consistency of the proposed method.

This paper is organized as follows. Section II describes the

basic concept of B-Splines and its key properties. Section III briefly introduces spline fitting technique. In Section IV, the covariance matrix estimate for the new observation model is derived. Implementation issues are discussed in Section V. Section VI provides some consistency analysis on the new observation model. Section VII shows results on B-spline SLAM using both simulation and experimental data. Section VIII concludes the paper.

## II. FUNDAMENTAL OF B-SPLINES

### A. Definition of a B-spline

A B-Spline curve of order $k$, is defined as

$$s(t) = \sum_{i=0}^{n} x_i \beta_{i,k}(t) \tag{1}$$

where $x_i(i = 0, ..., n)$ are the control points, $\beta_{i,k}(t)$ are the normalized B-Spline basis functions of order $k$ defined over the knot vector $T = [\varepsilon_0, ..., \varepsilon_{n+k}]$. A common form for clamped knot vector [15] for an order $k$ spline is:

$$T = [\underbrace{0, ..., 0}_{k}, \varepsilon_k, ..., \varepsilon_n, \underbrace{1, ...., 1}_{k}] \tag{2}$$

with $0 \le \varepsilon_k \le ... \le \varepsilon_n \le 1$.

The basis functions $\beta_{i,k}(t)$ are governed by the Cox-de Boor recursion formulas [15].

### B. Two key properties of a B-Spline

*1) Affine invariance property:* Affine transformation of a B-Spline can be achieved by transformation of the control points. This property makes it possible to solve the B-Spline SLAM problem using algorithms developed for point feature based SLAM.

*2) Differentiable property:* The basis function of a B-spline is differentiable:

$$\frac{\partial \beta_{i,k}(t)}{\partial t} = \frac{k-1}{\varepsilon_{i+k-1} - \varepsilon_i} \beta_{i,k-1}(t) - \frac{k-1}{\varepsilon_{i+k} - \varepsilon_{i+1}} \beta_{i+1,k-1}(t) \tag{3}$$

and the derivative of a B-spline of order $k$ is :

$$\frac{ds(t)}{dt} = (k-1) \sum_{i=0}^{n} \frac{x_i - x_{i-1}}{\varepsilon_{i+k-1} - \varepsilon_i} \beta_{i,k-1}(t) \tag{4}$$

## III. SPLINE FITTING

Deriving a spline function from a set of data points is a common problem in many areas. To be useful for SLAM the covariance matrix estimate of control points is also needed. This can be achieved using parameterized spline fitting.

In the parametrization process, for each data point $d_j = [p_j, q_j]^T$, $j = [1, ..., m]$ a time parameter value $t_j$ is assigned, for the whole set of data points a time parameter sequence $r_v = [t_1, ..., t_m]$ is formed. With a fixed time parameter sequence, the spline fitting problem becomes a minimization problem:

$$\min_{X} \sum_{j=0}^{m} \| \sum_{i=0}^{n} x_i \beta_{i,k}(t_j) - d_j \|^2 \tag{5}$$

where $X = [x_0, ..., x_n]^T$ and $x_i(i = 0, ..., n)$ are the control points. Define the positions of the raw data point as a matrix:

$$M = \begin{bmatrix} p_1 & \cdots & p_m \\ q_1 & \cdots & q_m \end{bmatrix}^T. \tag{6}$$

The least square solution for (5) is:

$$X = [B^T B]^{-1} B^T M = \Phi M \tag{7}$$

where

$$\Phi = [B^T B]^{-1} B^T \tag{8}$$

and B is the *collocation matrix*:

$$B = \begin{bmatrix} \beta_{0,k}(t_1) & \cdots & \beta_{n,k}(t_1) \\ \vdots & \ddots & \vdots \\ \beta_{0,k}(t_m) & \cdots & \beta_{n,k}(t_m) \end{bmatrix}. \tag{9}$$

For our B-Spline based SLAM, the time parameter sequence need to be invariant to the observation point. As length of a spline is invariant with observation point, chord length method [16] is used in this paper. The chord length method uses the ratio between the cumulated chord length and the total chord length to approximate the time parameter $t_j$:

$$\begin{aligned} l_1 &= 0 \\ l_j &= \sum_{i=1}^{j-1} \|d_{i+1} - d_i\| \\ l_{m-1} &= \sum_{i=1}^{m-1} \|d_{i+1} - d_i\| \\ t_j &= l_j/l_{m-1} \end{aligned} \tag{10}$$

where $\| \cdot \|$ is the Euclidean norm.

## IV. COVARIANCE MATRIX OF THE ESTIMATED CONTROL POINTS

Assuming the uncertainty of the raw measurement is $S$, the covariance matrix of control points can be derived from (7):

$$P_s = \frac{\partial \Phi M}{\partial M} S \frac{\partial \Phi M}{\partial M}^T \tag{11}$$

Since the size of $M$ is $m \cdot 2$ and size of $\Phi$ is $n \cdot m$, by the product rule for matrix calculus [17] we have:

$$\frac{\partial \Phi M}{\partial M} = (M^T \otimes I_n) \frac{\partial \Phi}{\partial M} + (I_2 \otimes \Phi) \frac{\partial M}{\partial M} \tag{12}$$

where

$$\frac{\partial \Phi}{\partial M} = \frac{\partial \Phi}{\partial B} \frac{\partial B}{\partial M} = \frac{\partial (B^T B)^{-1} B^T}{\partial B} \frac{\partial B}{\partial M} \tag{13}$$

Apply product rule on equation (13):

$$\frac{\partial \Phi}{\partial M} = ((B \otimes I_n) \frac{\partial (B^T B)^{-1}}{\partial B} + (I_m \otimes (B^T B)^{-1}) \frac{\partial B^T}{\partial B}) \frac{\partial B}{\partial M} \tag{14}$$

Using matrix calculus, $\partial (B^T B)^{-1}/\partial B$ can be derived as:

$$\frac{\partial (B^T B)^{-1}}{\partial B} = -(((((B^T B)^{-1})^T \otimes I_n)/(I_n \otimes (B^T B))) \frac{\partial (B^T B)}{\partial B} \tag{15}$$

where

$$\frac{\partial (B^T B)}{\partial B} = (B^T \otimes I_n) \frac{\partial B^T}{\partial B} + (I_n \otimes B^T) \frac{\partial B}{\partial B} \tag{16}$$

From (9) and (10), $\partial B/\partial M$ becomes:

$$\frac{\partial B}{\partial M} = \frac{\partial B}{\partial r_v} \frac{\partial r_v}{\partial M} \tag{17}$$

From (3) we have

$$\frac{\partial B}{\partial r_v} = \begin{bmatrix} \frac{\partial B_{1,1}}{\partial t_1} & \cdots & \frac{\partial B_{1,1}}{\partial t_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial B_{m,n}}{\partial t_1} & \cdots & \frac{\partial B_{m,n}}{\partial t_m} \end{bmatrix} \tag{18}$$

And $\partial r_v / \partial M$ can be derived using chain rule:

$$\frac{\partial r_v}{\partial M} = \frac{\partial r_v}{\partial E} \frac{\partial E}{\partial C} \frac{\partial C}{\partial M} \tag{19}$$

with

$$\begin{aligned} E &= [\triangle e_1, ..., \triangle e_{m-1}] \\ C &= [\triangle c_1, ..., \triangle c_{m-1}] \end{aligned} \tag{20}$$

where $\triangle e_j$ is the real length of the curve piece $j$ and

$$\triangle c_j = ||d_{j+1} - d_j||$$

is the chord length of the curve piece $j$. From (10) the Jacobian of $t_j$ w.r.t $E$ can be derived as:

$$\begin{aligned} \frac{\partial t_j}{\partial E} &= \partial(l_j/l_{m-1})/\partial E \\ &= (l_{m-1}(\partial l_j/\partial E) - l_j(\partial l_t/\partial E))/l_{m-1}^2 \end{aligned} \tag{21}$$

where

$$\frac{\partial l_j}{\partial E} = [\frac{\partial l_j}{\partial \triangle e_1}, ..., \frac{\partial l_j}{\partial \triangle e_{m-1}}] \tag{22}$$

As the function of the curve is unknown, it is not possible to estimate the real curve piece length. But it is possible to get a bound between the chord length and the real length from [18]:

$$0 \le \triangle e_j - \triangle c_j \le \frac{1}{12}\triangle e_j{}^3 \|S(t)''\|_{[t_j, t_{j+1}]}^2 \tag{23}$$

where $S(t)$ is the actual spline equation. As laser scanner has fine resolution, $\triangle c_j$ is approximately equals to $\triangle e_j$. Also the spline estimate $\hat{S}(t)$ and the real spline $S(t)$ should be close enough. Therefore, we have:

$$\triangle e_j \le \triangle c_j + \frac{1}{12}\triangle c_j{}^3 \|\hat{S}(t)''\|_{[t_j, t_{j+1}]}^2 \tag{24}$$

$\|\hat{S}(t)''\|_{[t_j, t_{j+1}]}$ can be derived using (4). Hence $\frac{\partial E}{\partial C}$ and $\frac{\partial C}{\partial M}$ can be calculated.

## V. IMPLEMENTATION ISSUES

### A. Data association

In traditional feature based SLAM problem, data association process is to associate the observations to the features in the state. In comparison with conventional feature based SLAM, data association for the B-Spline based SLAM contains two stages: 1). Pairing observation with map splines. 2). Identifying time parameter sequence $r_v$ for new observation.

*1) Pairing observation with state objects:* At step $n$, a new scan $B_n$ has been acquired. In order to perform the pairing, the new scan and the map estimate $V_{n,u}$, which contains $u$ splines, need to be in the same coordinate system. As the odometry may contain large uncertainty, we apply Iterative Closest Point (ICP) [19] scan matching algorithm on current scan $B_n$ and previous scan $B_{n-1}$ to get a more reliable relative pose information.

After $B_n$ and $V_{n,u}$ being in the same coordinate system, the following process were performed: a). Scan $B_n$ has been segmented result in $v$ segments corresponding to different

spline objects. b). For segment $i(1 \le i \le v)$ the laser beam angle of extreme points ($Seg_{i,s}$ and $Seg_{i,e}$) are calculated. These angles are compared against the laser beam angle of extreme points ($\hat{S}_{j,s}$ and $\hat{S}_{j,e}$) for map spline $j(1 \le j \le u)$. Also, taking into account of range information for segment $i$, the pairing process completes. Fig. 1. illustrates this idea.
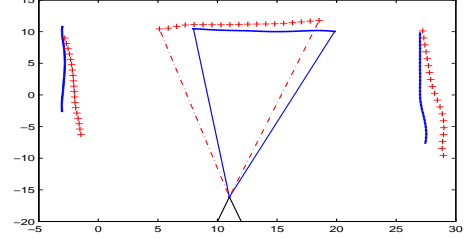


Fig. 1. Map estimates are shown in blue line, new scan points are shown in red crosses. Laser beam angle for extreme points of data from segment $i$ and map spline $j$ are calculated. If $\hat{S}_{j,s} \le Seg_{i,s} \le \hat{S}_{j,e}$ or $\hat{S}_{j,s} \le Seg_{i,e} \le \hat{S}_{j,e}$ then segment $i$ is associated with map spline $j$

*2) Identify time sequence for new observations:* For our proposed algorithm, we begin with initializing a set of B-Splines derived from appropriately segmented scan data and include control points of these splines in the state vector. We use the term "full spline" to describe the B-splines represented in the state vector. As new laser scans been acquired, new observation about "full splines" become available. The new observation may be about the whole "full spline", part of the "full spline" or previously un-observed part of the "full-spline". In the following we use "part of spline" to describe part of the "full spline" and "extension of spline" to describe previously un-observed part of the "full-spline". Whenever "extension of spline" is observed, the "full-spline" will be updated.

To be able to estimate the same set of control points of a spline for various observation, the time parameter sequence $r_v$ need to be uniquely defined which is invariant to the observation points. Thus we calculate the time parameter sequence for the new spline data according to the current "full spline".

To compute the time parameter sequence for new observation data, only the start and end point of the observed spline data need to be corresponded with the spline estimate $\hat{S}$. The ray tracing [20] method has been used to find the time parameter $t$ for the extreme points. Suppose time parameter for start and end point has been identified as $t_s$ and $t_e$, time parameter for remaining spline data points can be calculated by a modified version of (12):

$$\begin{aligned} l_1 &= 0 \\ l_j &= \sum_{i=1}^{j-1} ||d_{i+1} - d_i|| \\ l_{m-1} &= \sum_{i=1}^{m-1} ||d_{i+1} - d_i|| \\ t_j &= l_j(t_e - t_s)/l_{m-1} + t_s \end{aligned} \tag{25}$$

### B. Spline fitting for new observations

After the new spline data has been correctly associated with current "full spline" $\hat{S}$, the observation model of the new spline data can be derived.

*1) Case 1: re-observe "full spline":* If the new observation is made to whole $\hat{S}$, control points and the associated covariance matrix for current observation can be derived using the process detailed in Section III and Section IV.

*2) Case 2: re-observe "part of the spline":* When the new observation corresponds to part of $\hat{S}$, we use the following process to estimate the control point.

Assume the knot vector for an order $k$ spline is (2) time parameter sequence for the new observation is $r_v = [t_1, ..., t_m]$ while $0 \leq \varepsilon_a < t_1 < t_m < \varepsilon_b \leq 1$. The collocation matrix for this is:

$$B_{new} = \begin{bmatrix} \beta_{a,k}(t_1) & \cdots & \beta_{b,k}(t_1) \\ \vdots & \ddots & \vdots \\ \beta_{a,k}(t_m) & \cdots & \beta_{b,k}(t_m) \end{bmatrix}. \qquad (26)$$

Using our proposed method, control points $x_a$ to $x_b$ and associated covariance matrix can be derived.

*3) Case 3: observe "extension of the spline":* If unexplored part of a spline has been observed, $\hat{S}$ needs to be updated. Common area between $\hat{S}$ and the new observation need to be firstly identified (Fig. 2. illustrates this). An updated length of the spline can be calculated. Time parameter sequence for the new observation according to new length will be calculated. Estimate of the control points can be derived from method described in *Case* 2.
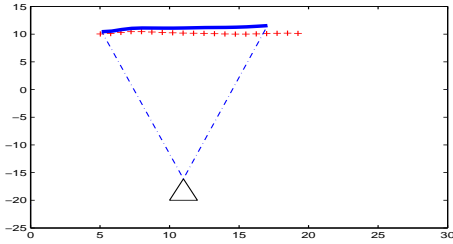


Fig. 2. Map estimate are shown in blue line, new scan points are shown in red crosses. Laser beam angle for extreme points of map spline $\hat{S}$ are calculated. After common area being identified. the spline length has been updated.

## VI. CONSISTENCY ANALYSIS IN SIMULATION

In this section, we use simulation to demonstrate the consistency of the proposed covariance analysis. The consistency of the algorithm has been tested by computing the average normalized estimation error squared (NEES) of different runs and then perform a chi-square test [3].

In our simulation, the range finder observations are generated by finding the intersection points between the reference spline and the artificial laser beams from a fixed robot pose. The field view of the sensor is $[-\frac{\pi}{2}, \frac{\pi}{2}]$ and the sensor range used is $6m$. To simulate the real laser data, we use $0.5^o$ for the laser resolution and we used a zero mean Gaussian noise with $6mm$ standard deviation to add on the range readings.

### A. Estimate control points when "full spline" is observed

When "full spline" is observed, we utilize the proposed spline fitting algorithm in Section III to get the control points' estimate. Fig. 3 shows a simulation result of this situation. The control points' estimate and associated $2\sigma$ uncertainty ellipses are compared with the ground truth control points [1].

To investigate the consistency of the observation model, fifty simulation data sets are generated and the average NEES obtained is 33.4578, which is within the 95% confidence gate 33.9244.
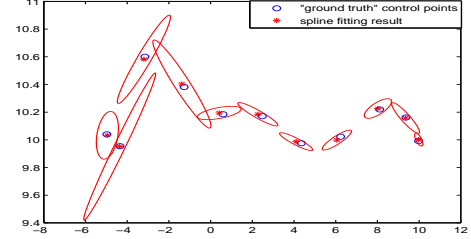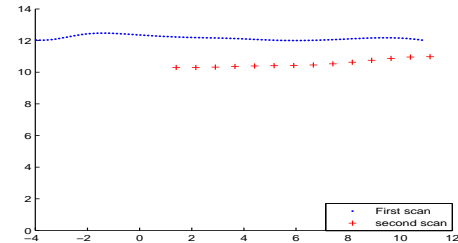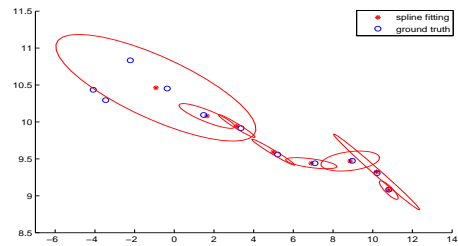


Fig. 3. Spline fitting with noisy data

### B. Estimate control points when "part of spline" is observed

A simulation result for the scenario when only "part of spline" is observed is shown in Fig. 4(a). The uncertainty ellipses are shown in Fig. 4(b).

Again, 50 simulation data sets with added independent noise (from the second scan only) are generated. The average NEES obtained is 16.0368, which is smaller than the 95% confidence gate 26.2962.



(a) Raw data for observing part of a spline. First scan is "full spline", second scan is "part of spline"



(b) Spline fitting result when "part of spline" is observed. The 3 left most control points are not estimated here, as this range has not been observed

Fig. 4. Spline fitting with noisy data when "part of spline" is observed

---

[1]In simulation, the ground truth of the control points can be obtained using the spline fitting with noise-free data, see [14] for details

## C. Estimate control points when an extension to the spline is observed

A simulation result for the scenario when "extension of the spline" is observed is shown in Fig. 5.

For this case, after the spline fitting using the second scan, a new spline fitting needs to be performed using the first scan (spline re-fitting) since the "full spline" has been updated. Further consistency analysis of the observation model is conducted in the next section using SLAM results.
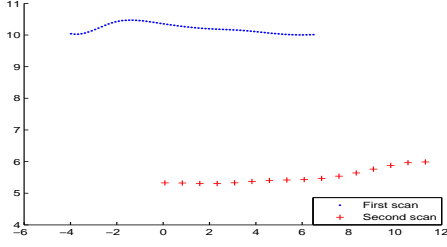


(a) Raw data for observing extension of a spline. After first scan was made, the "full spline" length is considered to be $L_1$. Spline fitting was performed based on current spline length $L_1$. After second scan was made, section where previously have not been observed has been identified. The "full spline" length was updated to $L_2$. Spline fitting was performed based on new full spline length $L_2$.
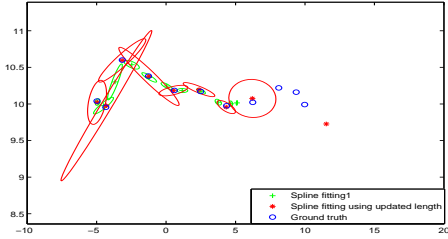


(b) Spline fitting results. "Spline fitting 1" refers to spline fitting result using $L_1$. "Spline fitting using updated length" refers to spline fitting result using $L_2$.

Fig. 5. Spline fitting with noisy data when extension of the spline is observed
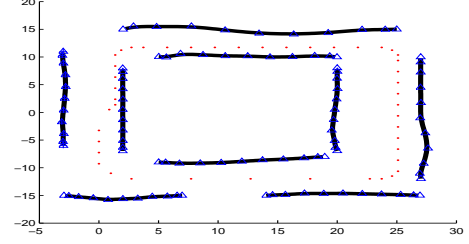
## VII. SLAM RESULTS

### A. Point feature based I-SLSJF SLAM algorithm

I-SLSJF [21] is a local submap joining algorithm that fuses the local maps together using Extended Information Filter (EIF) and least squares approach. I-SLSJF was shown to be able to produce consistent estimates for point feature based SLAM problems in an efficient way. When each local map is built by a unrelated observations in a single time step, I-SLSJF is similar to SAM [6]. However, in the B-Spline based SLAM, correlations in the observation at certain time step is unavoidable. Therefore SAM algorithms cannot be directly used here. Therefore I-SLSJF is used in our implementation.
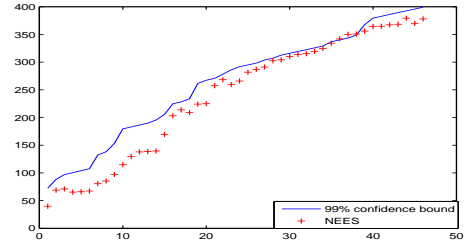
### B. SLAM result using simulation data

A simulation experiment containing 9 splines in the environment and 46 robot poses was created to further evaluate the consistency of the proposed SLAM algorithm.
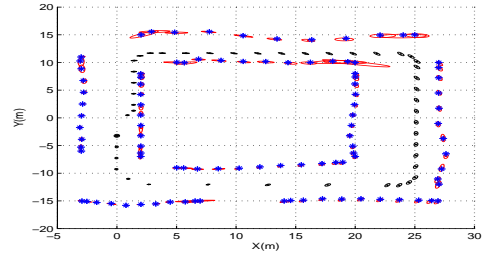
The simulation environment and the SLAM results obtained using I-SLSJF [21] are shown in Fig. 6. Comparing with the "ground truth", the estimate of control points and robot poses appear to be very accurate. To evaluate the consistency of the observation model and the SLAM results, the NEES of the final estimate (for the final map containing 99 control points and 46 poses) obtained is 391.1603 , while the 99% confidence gate is 399.3297.



(a) A simulation environment with 9 splines, red dots show the robot poses, blue triangles show the "ground truth" of the control points



(b) Comparison of NEES v.s. 99% confidence gate at each step



(c) SLAM result using the simulation data. Blue crosses show the "ground truth" control points. Estimated control points almost coincide with "ground truth"

Fig. 6. Simulation environment and SLAM result

### C. SLAM using real data

In order to validate the proposed algorithm, experiments have been performed with laser scans of University of Freiburg from the Robotics Data Set Repository [22].

We selected 55 scans from the data set. Due to the imperfection of the data association, manual association are performed on some of the scans. In the experiment, fixed degree and knot vector were used. Fig. 7(b) shows the control points estimates obtained from I-SLSJF. The map contains 60 cubic splines. Each spline contains 11 control points. Fig. 7(c) depicts the map using cubic splines derived from the control point estimates.
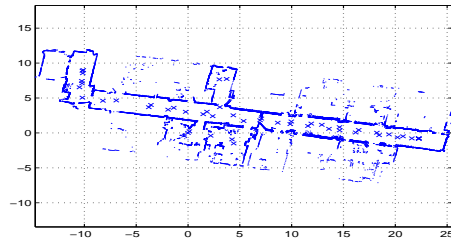
(a) Trajectory SLAM result. Firstly, Scan matching are performed using the selected scans, then optimization is applied on the scan matching result
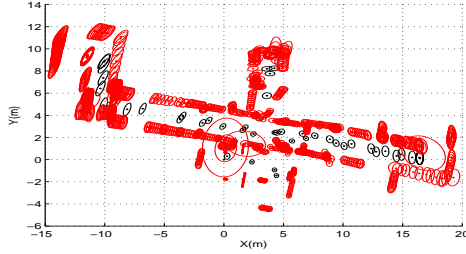


(b) Control points and robot pose estimate
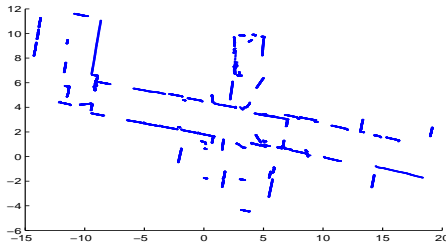


(c) Spline derived from the estimated control points

Fig. 7.   SLAM result using the University of Freiburg dataset

## VIII.  CONCLUSIONS AND FUTURE WORKS

In this paper, a method to estimate the covariance matrix for control points in a B-Spline based SLAM algorithm is proposed. The error for control points is from two dependent sources: the raw data error and the chord length error. It has been shown that, under various conditions, the proposed error analysis leads to consistent spline fitting result. Some initial SLAM results using simulation and real data further proves that the estimate is consistent. With the new observation model, accuracy for SLAM problem will be improved.

We plan to make use of more large scale real data sets with our new observation model and the I-SLSJF algorithm to further evaluate the proposed approach. In the future, we are planning to improve the data association process such that manual association is not required. We will also focus on extend the new observation model to 3D SLAM using 3D splines as features. Some further theoretical investigation on the consistency issue for SLAM with B-Spline features is also underway.

## REFERENCES

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, 2001. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons.

[2] T. Bailey, J. Nieto, J. Guivant, M. Stevens and E. Nebot. Consistency of the EKF-SLAM algorithm. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3562-3568, Beijing, China, October 9-15, 2006.

[3] S. Huang and G. Dissanayake. Convergence and consistency analysis for Extended Kalman Filter based SLAM. *IEEE Transactions on Robotics*, 23(5): 1036-1049, 2007.

[4] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. Analysis and improvement of the consistency of Extended Kalman Filter-based SLAM. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, CA, pp. 473-479, May 19-23 2008.

[5] F. Dellaert and M. Kaess. Square root SLAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 25(12): 1181-1203, 2006.

[6] S. Huang, Z. Wang, G. Dissanayake and U. Frese. Iterated D-SLAM map joining: Evaluating its performance in terms of consistency, accuracy and efficiency. *Autonomous Robots*, 27: 409-429, 2009.

[7] P. Newman, D. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1180-1187, Orlando, Florida, USA, 15-19 May 2006.

[8] G. Grisetti, D. L. Rizzini, C. Stachniss, E. Olson and W. Burgard. On-line constraint network optimization for efficient maximum likelihood mapping. In *Proceedings of 2008 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1880-1885, Pasadena, California, May 2008.

[9] W. Burgard, C. Stachniss, G. Grisetti, B.Steder, R. Kummerle, C. Dornhege, M. Ruhnke, A.Kleiner, and J. D. Tardos. A comparison of SLAM algorithms based on a graph of relations. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[10] J. Tardos. Representing partial and uncertain sensorial information using the theory of symmetries. In *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1799-1804, May 1992,

[11] D. Rodriguez-Losada, F. Matia, A. Jimenez, and R. Galan. Consistency improvement for SLAM-EKF for indoor environments. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 418-423, Orlando, Florida, USA, May 2006.

[12] J. Nieto, T. Bailey and E. Nebot. Scan-SLAM: Combining EKF-SLAM and scan correlation. *International Conference on Field and Service Robotics*, 2005.

[13] L. Pedraza, G. Dissanayake, J. Valls Miro, D. Rodriguez-Losada, and F. Matia. Extending the limits of feature-based SLAM with B-Splines. *IEEE Transactions on Robotics*, 25: 353-366, 2009.

[14] M. Liu, S. Huang and G. Dissanayake. A new observation model for B-Spline SLAM. *Australasian Conference on Robotics and Automation*, Sydney, Australia, 2009.

[15] L. Piegl and W. Tiller. *The NURBS book*. 2nd ed. Berlin: Springer-Verlag 1997.

[16] J. Hoschek, and D. Lasser. *Fundamentals of Computer Aided Geometric Design*. AK Peters 1993

[17] J. R. Magnus. On the concept of matrix derivative. To appear in *Journal of Multivariate Analysis*. Available online: http://center.uvt.nl/staff/magnus/wip12.pdf

[18] M. S. Floater. Arc length estimation and the convergence of polynomial curve interpolation. *BIT Num. Math*. 45 (2005), 679-694.

[19] P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Trans. on Pattern Analysis and Machine Intel*. 14(2): 239-256, 1992.

[20] A. J. Sweeney and R. H. Barrels. Ray tracing free-form B-spline surfaces. *IEEE Computer Graphics and Application*, 6(2): 41-49, Feb. 1986.

[21] S. Huang, Z. Wang, G. Dissanayake and U. Frese. Iterated SLSJF: A sparse local submap joining algorithm with improved consistency, *Australiasan Conference on Robotics and Automation*. Canberra, December 2008. available at: http://www.araa.asn.au/acra/acra2008/papers/pap102s1.

[22] A. Howard and N. Roy. The robotics data set repository (Radish). Available online: http://radish.sourceforge.net