A new observation model for B-Spline SLAM

Minjie Liu, Shoudong Huang, Gamini Dissanayake

ARC Centre of Excellence for Autonomous Systems
Faculty of Engineering and Information Technology
University of Technology, Sydney
{mliu,sdhuang,gdissa}@eng.uts.edu.au

Abstract

Recently, a novel laser data based SLAM algorithm using B-Spline as features has been developed in [Pedraza et al., 2007]. EKF is used in the proposed BS-SLAM algorithm and the state vector contains the current robot pose together with the control points of the splines. The obervation model used for the EKF update is the intersections of the laser beams with the splines contained in the map. In this paper, we propose a new observation model for B-Spline SLAM. By properly defining the control points for the splines, the observation model can be expressed as a function of relative positions between control points and the robot pose, which is the same format as what used in point feature based SLAM. This new observation model make it possible to apply optimization based techniques to B-Spline SLAM, which has the potential to resolve the inconsistency issues of B-Spline SLAM.

1 Introduction

Reliable localization is the key concept of any autonomous robotic system [Borenstein et al., 1996]. In some scenarios, robot does not have any prior knowledge of the environment and the problem becomes simultaneous localization and mapping (SLAM). Solutions to SLAM enable robots to build a map of an unknown enviornment and navigate in the environment concurrently.

Ever since SLAM firstly been introduced, many approches have been investigated. Most of the early SLAM work is point-feature based [Dissanayake et al., 2001] [Guivant et al., 2001] [Leonard et al., 1992]. The main drawback with point-feature based SLAM is that measurements acquired from typical sensors does not corresponding to point feature in the environment. After the raw sensor data is acquired, post processing is required to extract point features. This process may potentially introduce information loss and data association error. Further more, in some situation, the environment does not have enough significant structure to enable point features to be robustly extracted from.

When laser sensor is used, one popular way to perform SLAM is the so called “trajectory based SLAM” [Newman et al., 2006] [Grisetti et. al 2008] where the 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. Although many promising results have been achieved in this way, the lack of a proper model to represent the environment is a major limitation of these approaches.

A number of research groups have tried to use more complex geometric to represent the environment. In 1992 the Symmetries and Perturbations Model (SP-model) [Tardos, 1992] was introduced. It provides a general way to represent and process uncertain geometrical data. It also makes update using partial observation possible (e.g. observe a point on a line). Some promising result for SP-model based SLAM has been reported in [Castellanos and Tardos, 1999] and [Weingarten, 2006]. However, the SP-model based SLAM requires local coordinate system attached to all features [Folkesson et al., 2007]. Moreover, some types of features are difficult to be modeled using SP model.

Line segments have been used as features in [Rodriguez-Losada et al., 2006]. Although line-based SLAM has been implemented in real time [Marzorati et al., 2007] [Gee and Mayol-Cuevas, 2006], its inconsistency has been reported in [Rodriguez-Losada et al., 2006]. Further more line segment-based SLAM is not suitable for outdoor environments where complex geometry presents.

Nieto et. al [Nieto et al., 2005] proposed the Scan SLAM algotithm, 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 sensed data. Although these templates can be augmented as more data becomes available, the templates themselves are not included in the EKF state vector and hence the uncertainty of their shapes can not be presented although the uncertainty of their position is covered by the EKF estimate.

Very recently, using B-Spline to solve the SLAM gained the momentum. Using B-Splines to represent the environment has some clear advantage: 1) B-Splines are able to represent complex environment. When representing complex geometry, the approximation

accuracy increases as the number of control points and/or the degree of the spline increases. 2) B-Splines have some nice properties. For example, any affine transformation can be applied to the curve by simply applying it to the control points. 3) B-Splines can be extended, therefore information regarding certain geometric feature can be fully associated with the feature. In 2009, Pedraza et al., developed the BS-SLAM [Pedraza et al., 2007]. It uses B-Splines as features and uses EKF to solve the SLAM problem. B-Splines are represented by control points in the state vector.

However, it is shown in [Pedraza et al., 2007] that the proposed BS-SLAM can produce inconsistent estimate in some scenarios. One reason for the inconsistency might be the use of EKF [Bailey et al., 2006] [Huang and Dissanayake, 2007] and the optimization based techniques have the potential to resolve the inconsistency problem [Huang et al., 2008] [Dellaert and Kaess, 2006] [Huang et al., 2009].

In this paper, we show that when the control points are properly defined, the observation model can be expressed as a function of relative positions between the control points of the observed spline and the observation point. With this 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. Some initial results demonstrate the consistency of the new observation model.

This paper is orginzed as follows. Section 2 provides the basic concept of B-splines. Section 3 shows how control points can be defined such that the new observation model can be achieved. In Section 4, some implementation issues are discussed such that the observation to the control points can be obtained under different situations. Section 5 shows some intial SLAM results using simulation and real data. Section 6 concludes the paper and addresses future work for SLAM using splines as features.

2 Fundamental of B-Splines

2.1 Definition of B-spline

A B-Spline curve of order k is defined as

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

where $x_i \in (i = 0,....,n)$ are the control points, $\beta_{i,k}(t)$ are the normalized B-Spline basis functions of B-Spline order k defined over the knot vector $T = [\varepsilon_{0,} \ldots \varepsilon_{n+k}]$.

The knot vector is a non-descending sequence. There are two types of knot vectors: clamped and unclamped [Piegl and Tiller, 2007]. In this paper, only clamped knot vector is used for convenience. A common form for clamped knot vector for an order k spline is:

$$T : [\underbrace{0,0,...,0}_{k}, \varepsilon_k, ..., \varepsilon_n, \underbrace{1,1,...,1}_{k}] \quad (2)$$

When clamped knot vector is used, the first and last control point defines the start and end point of the spline curve.

The basis functions $\beta_{i,k}(t)$ are governed by the Cox-de Boor recursion formulas [Boor, 1978]:

$$\beta_{i,1}(t) = \begin{cases} 1, & if \quad \varepsilon_i \le t \le \varepsilon_{i+1} \\ 0, & otherwise \end{cases}$$

and for $k > 1$

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

Where t is the "time" parameter reperesenting which point of the spline $s(t)$ is corresponding to. An example of a clamped spline with order 4 is shown in Fig. 1.

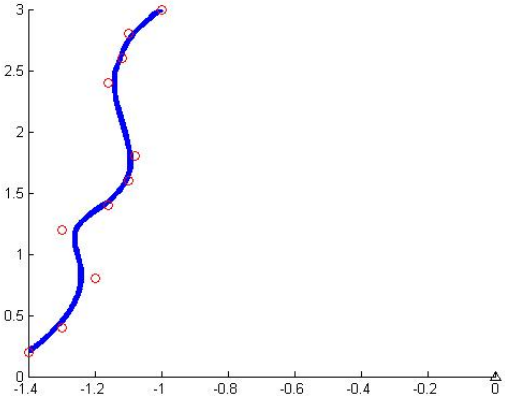


Fig. 1. An example of B-Spline with order 4. The knot vector for this spline is [0,0,0,0,0.125,0.25,...0.75,0.875,1,1,1,1]. The control points are shown in red circle.

2.2 Spline fitting

Spline fitting is a fundamental problem in computer graphics. The problem is: given a set of data points $\{d_0, \ldots, d_m\}$ which correspond to an unknown curve, find the B-spline function to approximate the data points.

In spline fitting, the order of B-spline k, the knot vector T and the time sequence $\{t_0, \ldots, t_m\}$ (t_j corresponding to data point d_j) need to be firstly defined. Then the basis function $\beta_{i,k}(t_j)$ for each point d_j can be derived. The spline fitting problem becomes a minimization problem: given the spline order k, the knot vector $T = [\varepsilon_{0,} \ldots \varepsilon_{n+k}]$, the time sequence $\{t_0, \ldots, t_m\}$, find the control points $X = \{x_1, \ldots, x_n\}$ to minimize the sum of the squared distance of data points d_j to their projection on spline $s(t)$. That is,

$$\min_X \sum_{j=0}^{m} \| \sum_{i=0}^{n} x_i \beta_{i,k}(t_j) - d_j \|^2 \quad (4)$$

The least square solution for this is:

$$X = [B^T B]^{-1} B^T d_j = \Phi d_j \quad (5)$$

where

$$\Phi = [B^T B]^{-1} B^T \quad (6)$$

and B is the *collocation matrix*:

$$B = \begin{bmatrix} \beta_{0,k}(t_0) & \cdots & \beta_{n,k}(t_0) \\ \vdots & \ddots & \vdots \\ \beta_{0,k}(t_m) & \cdots & \beta_{n,k}(t_m) \end{bmatrix} \quad (7)$$

3 New Observation Model for B-spline Observation

As shown in the previous section, given the order, the knot vector and the control points, the location and the shape of the B-spline is completely decided. However, when using a number of points $\{d_0,\ldots,d_m\}$ which correspond to certain geometry (e.g. from the laser scan) to compute the control points through spline fitting, even when the order and the knot vector are all given, there are still infinite possible sets of control points that fit the points due to the infinite number of possible time sequence $\{t_0,\ldots,t_m\}$. Only when the order, the knot vector and the time sequence are all given, the set of control points can be uniquely determined.

In the following we show how to decide the time sequence such that the same set of control points can be estimated from the scan data obtained from different observation points.

3.1 Time sequence that is invariant to the observation point

For any curve, regardless from which angle it is observed, its curve length is fixed. Therefore the time sequence computed from the ratio between curve lengths is invariant to the obervation point.

Hoschek and Lasser [Hoschek and Lasser, 1993] proposed the chord length method for approximating the time sequence "t". They use the ratio between the cumulated chord length and the total chord length to approximate the time sequence "t":

$$l_0 = 0$$

$$l_c = \sum_{j=1}^{c-1} \| d_{j+1} - d_j \| \quad (8)$$

$$l_t = \sum_{i=1}^{m-1} \| d_{i+1} - d_i \| \quad (9)$$

$$t_c = l_c / l_t \quad (10)$$

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

When the time sequence is derived from this method, the same set of control points can be estimated from the raw scan data no matter where the observation point is.

Once the time sequence is derived, the estimated control points can be simply obtained by the spline fitting algorithm described by (4)-(7).

3.2 Covariance matrix of the estimated control points

To use the estimated control points as observation model to SLAM algorithm, the covariance matrix need to be correctly modelled. There are two elements contribute to the covariance matrix of the spline control points: uncertainty of the spline fitting due to the noise from raw scan, and the inaccuracy of the time sequence estimation using chord length.

Assume the uncertainty of the raw data is:

$$R_{old} = diag[R_{d_0},...,R_{d_m}] \quad (11)$$

where R_{d_j} is the uncertainty of the scan data d_j. Then the covariance matrix according to the spline fitting becomes:

$$P_s = \Phi^T R_{old} \Phi \quad (12)$$

where Φ is given in (6).

For the error introduced by approximating the time sequence, one source is from the sensor angular resolution, which is the same for all the estimated control points:

$$P_{t1} = diag[R_{res},\ldots,R_{res}] \quad (13)$$

Another source of error is due to the difference between chord length and curve length. Because we use the clamped B-Spline to interpolate raw data. The first and last control point corresponding to the start and end point of raw data. As mentioned earlier, we use the ratio between cumulated chord length and total chord length to approximate the time parameter "t". The time parameter "t" at these two control points are fixed as 0 and 1.When approximating the time sequence towards the spline center, the error for "t" accmulates and the control points towards the middle have larger uncertainties. The approximation error can be derived using the following formula:

$$P_{t2} = diag[0,\ldots,\min(t_c,1-t_c),\ldots,0]\cdot cons \quad (14)$$

where t_c is defined in (10) and $cons$ is a constant term depending on the sensor noise, sensor angular resolution, etc.

The covariance matrix of the observed control points is then computed taking into account all these factors:

$$P = P_s + P_{t1} + P_{t2} \quad (15)$$

3.3 Simulation results on the observation model

To demonstrate the consistency of the proposed observation model, intensive simulations have been carried out. In our simulation, the laser points is generated by finding the intersection points between artificial laser beams from a fixed robot pose and the reference spline. The idea of this is shown below in Fig. 2.

3.3.1 Estimate the "ground truth" of control points using noise free simulation data

To get the approximation of "ground truth" of the corresponding control points, the length of the reference spline must be approximated as accurate as possible. Here a small angular resolution (0.1^o) was used and no noise is added in the simulated scan data. The intersection points obtained by the simulator is shown in Fig. 3(a). The time sequence was defined using equations (8)-(10). Using the spline fitting method described in Section 2.2, an estimate of the control points are obtained. The associated covariance matrix computed by (15) is very small and the estimated control points can be used to approximate the "ground truth". To validate this, a new spline equation was derived using the obtained "ground truth" control points. The curve of the new spline was compared with the reference spline in Fig. 3(b). The mean square error for this approximation is $2.4\cdot10^{-3}m^2$.

To further prove the validity of the “ground truth” control points, new observation of the spline was taken from another position (Fig. 3(c)). The control points estimated from the noise-free second scan were derived and transferred back to the coordinate defined by first pose. The result is shown in Fig. 3(d).

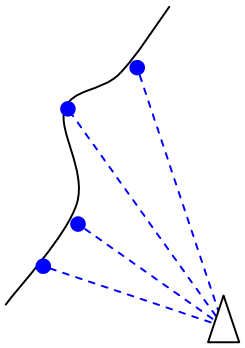


Fig. 2. Finding the scan points

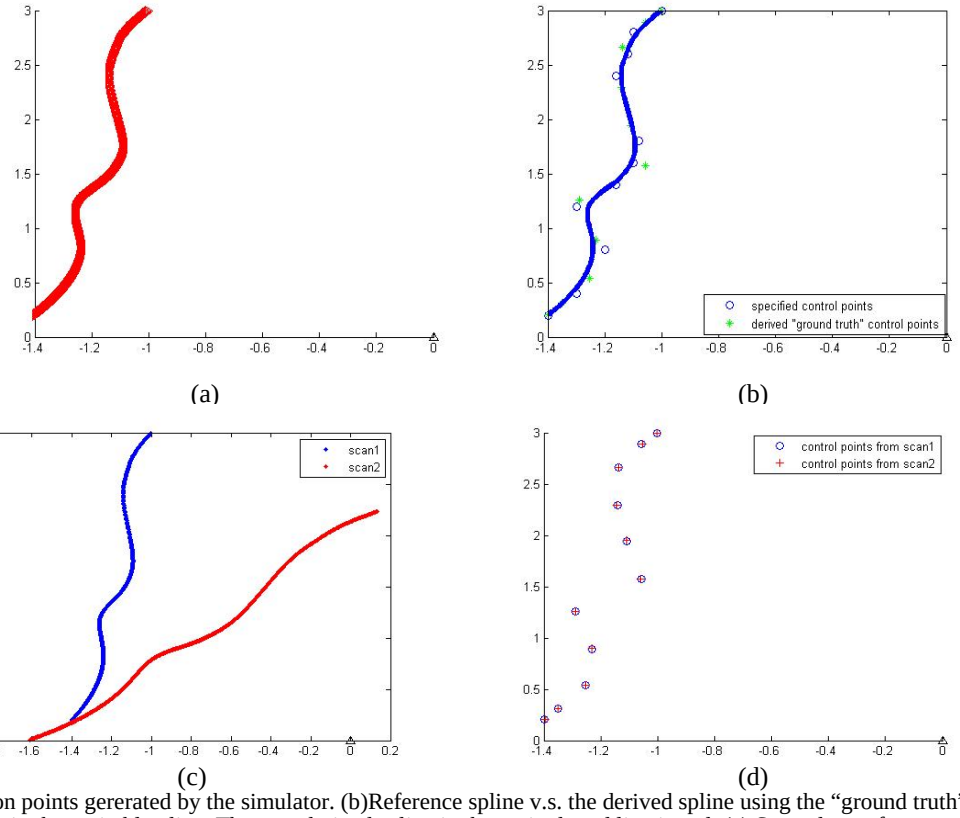


Fig. 3. (a) Intersection points gererated by the simulator. (b)Reference spline v.s. the derived spline using the “ground truth” control points. The reference spline is shown in blue line. The new derived spline is shown in dotted line in red. (c) Second scan from another robot pose vs. First scan. (d) Control points derived from second scan (transferred into the global coordinate) vs. Control points derived from first scan

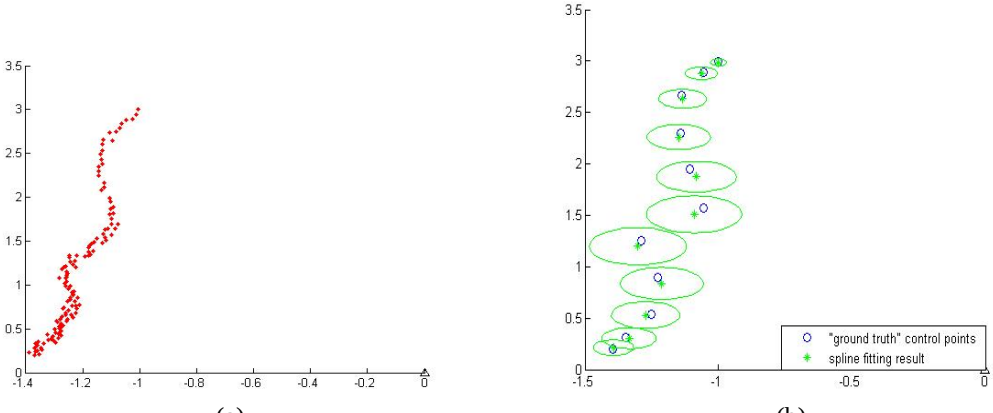


Fig. 4. (a) Noise containminated simulation data. (b) Control points from spline fitting using noisy data v.s. “ground truth” control points.

3.3.2 Estimate the control points through noisy scan data

To simulate the real laser data, we use 0.5^{o} for the laser resolution and we used a zero mean Gaussian noise with 12mm standard deviation to add on the range readings. The simulated noisy scan data we used for spline fitting is shown in Fig. 4(a). The control points derived from the proposed spline fitting algorithm and the associated 2σ uncertainty ellipses are plotted and compared with the "ground truth" control points in Fig. 4(b). It can be seen that the covariance matrix computed using the approach stated in Section 3.2 is reasonable.

Although we have shown how the control points can be estimated when the whole spline is observed, there are a lot of implementation issues such as when only part of the spline is obverved, or an extension of the spline is observed, etc. In the next section, we will discuss how to perform the spline fitting and get the observation model in these situations.

4 Implementation Issues

In reality, it is impossible to perceive whether the whole length of a spline has been observed since we may not know whether there is still an extension of the spline or not. Because of this, we propose to implement the spline fitting in the following way.

We always assume what has been observed is the "full spline". When a spline has been re-observed, the new measurements need to be compared with current "full length" spline. Judgements need to be made on whether the new observation corresponding to the "full spline", "part spline" or "extension of the spline". For all the three cases the control points will be derived according to the possibly extended "full spline". Details of the implementation process are discussed below. It is based on the assumption that data segmentation and data association (which segment belongs to which spline) have previously been done correctly.

4.1 Identify relation between new spline data and "full spline"

When new observation of a spline has been made, the relation between the new spline data and the current "full spline" need to be identified (e.g. whether new data corresponding to full length, part length or an extension of the current "full spline"). The new spline data D_{obs} and the current "full spline" $S^{*}(t)$ need to be in the same coordinate frame in order to check the relationship. As the odometry information may contain large uncertainty, we utilizes standard *Iterate Closest Point (ICP)* scan matching algorithm to get more reliable relative pose information.

The first step of our matching method is to calculate the laser beam angle for both the start and end point of the spline data (d_s, d_e) and current "full spline" (s_s, s_e). The idea of this is shown in Fig. 5.

As observation to extreme points of a spline can not be made most of the time, a threshold angle h has been used to judge whether the extreme point has been observed. If the extreme point of the new data does not coincide with the "full spline" extreme points, extreme angle for new spline data (d_s, d_e) are compared against "full spline" extreme angle (s_s, s_e) to check whether the extreme point of new data lies on the current "full spline". Once the relation between the spline extreme points and data extreme points being established, the relationship between the new spline data and "full spline" is identified. Fig. 6. depicts the checking process in detail.

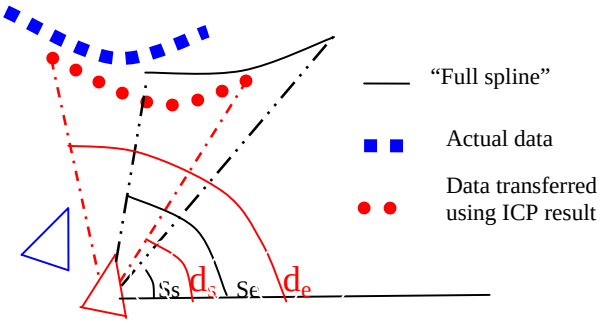


Fig. 5. Process to determine laser beam angle for extreme points

4.2 Control points for new observation

As stated ealier, to be able to estimate the control points, the time sequence "t" need to be uniquely defined. Therefore only knowing the relation between the new spline data and the "full spline" is not sufficient. Time sequence of the new spline data with respect to the possibly extended "full spline" need to be derived.

Because we only check which part of a spline the new observed data corresponding to, only the start point d_{start} and end point d_{end} of the observed spline data need to be checked. The *ray tracing* [Sweeney and Barrels, 1986] method has been used to find the time "t" for the extreme points.

In some situation, the direction for extreme points may not intersect with the spline estimate. To determine the "common range" for the new data and the "full spline", we iterativly find the intersection angle by changing the extreming angle with sensor resolution. The idea of this process is shown in Fig. 7.

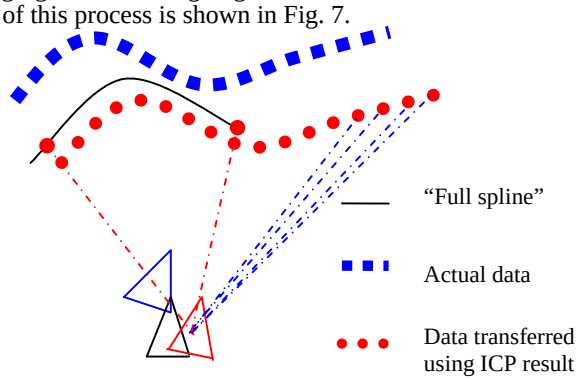


Fig. 7. Process to determine t_{start} and t_{end} when new part of spline is observed

After the time sequence for new observation is derived, the new control point estimate can be derived using spline fitting algorithm described by formula (4)-(7).

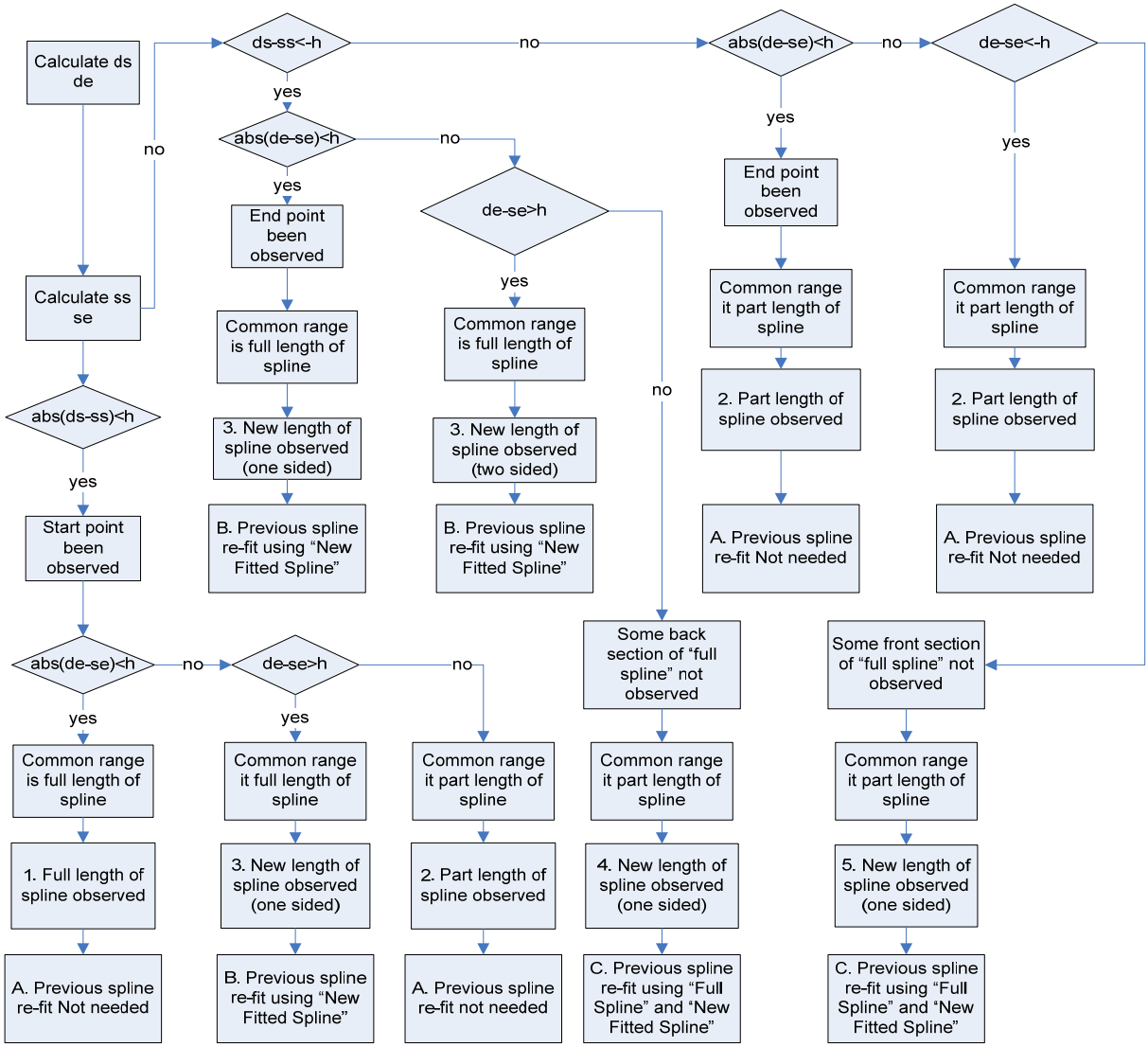


Fig. 6. Process to identify relation between new spline data and current "full spline"

5 Initial Results

5.1 SLAM Simulation Result

A small scale SLAM simulation experiment containing three splines was conducted to further evaluate the consistency of the estimate of the control points. Fig. 8(a) shows the shape of the three splines and the robot trajectory. Fig. 8(b) shows the raw data used for the SLAM simulation. Fig. 8(c) shows the control points estimate results obtained using I-SLSJF [Huang et al., 2008] with the new observation model. Comparing with the "ground truth", the estimate of control points and robot poses appear to be consistent.

5.2 SLAM using real data

In order to validate the proposed algorithm, experiments have been performed with scans of Intel Lab from the Robotics Data Set Repository [Howard and Roy, 2003]. Due to the lack of association method, we manually selected 23 scans from the dataset. In the experiment, fixed degree and knot vector were used. Fig. 9(a) shows the control points estimate results obtained from I-SLSJF. The map contains 30 cubic splines. Each spline contains 13 control points. Fig. 9(b) depicts the map using cubic splines derived from the control point estimates. The performance can be further improved using lower order spline with less knots for less complex features e.g. flat features.

6 Conclusions and Further Work

This paper further investigates the laser based SLAM problem using B-Spline as features. A new observation model is derived such that the observation can be naturally expressed as a function of relative positions between the control points of the observed spline and the observation point. Some initial results using both simulation data and real laser data have demonstrate the consistency of the new observation model.

With this new observation model, the B-spline

SLAM problem can be transferred into a point-feature based SLAM problem. Some initial SLAM results have shown that B-Spline SLAM can be successfully solved using I-SLSJF algorithm [Huang et al., 2008].

It is expected that most of the existing point-feature based SLAM algorithms, such as SAM, I-SLSJF, I-DMJ, [Huang et al., 2009] could be applied to estimate the control points. Especially, these SLAM algorithms that based on optimization techniques and exploiting sparse structure of the observations, have the potential to improve both the consistency and efficiency of the B-Spline SLAM.

Our future work include further improving the implimentations of B-spline SLAM such as spline segmentation, scan matching, data associations, as well as to apply the efficient point-feature based SLAM algorithms to B-Spline SLAM for large-scale data sets.

References

[Bailey et al., 2006] 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)*, pages 3562-3568, Beijing, China, October 9-15, 2006.

[Boor, 1978] C. de Boor. A Practical Guide to Splines. Springer 1978.

[Borenstein et al., 1996] J. Borenstein, H.R. Everett, L. Feng. Navigating Mobile Robots: Systems and Techniques, A.K. Peters, Ltd, Wellesley, MA, 1996

[Castellanos and Tardos, 1999] J. A. Castellanos and J. D. Tardos. Mobile Robot Localization and Map Building: A Multisensor Fusion Approach. Kluwer Academic Publishers, 1999.

[Dellaert and Kaess, 2006] 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.

[Dissanayake et al., 2001] G. Dissanayake, P. M. Newman, H. F. Durrant-Whyte, S. Clark, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17:229-241, 2001.

[Folkesson et al., 2007] J. Folksson, P. Jensfelt and H.I. Christensen. The M-Space Feature Representation for SLAM. *IEEE Transactions on Robotics*, 23(5): 1024-1035.

[Gee and Mayol-Cuevas, 2006] A. P. Gee, W. Mayol-Cuevas. Real-Time Model-Based SLAM Using Line Segments. In *2nd International Symposium on Visual Computing*, pages 354-363. November 2006.

[Grisetti et. al 2008] G. Grisetti, D. L. Rizzini, C. Stachniss, E. Olson and W. Burgard. Online constraint network optimization for efficient maximum likelihood mapping. In *Proceedings of 2008 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1880-1885, Pasadena, California, May 19-23, 2008.

[Guivant et al., 2001] J. Guivant, E. Nebot, and H. Durrant-Whyte. Simultaneous localization and map building using natural features in outdoor environments. *Proceedings of the Sixth Intelligent Autonomous Systems (IAS)*, pages 581–586, Venice, Italy, 25–28 July 2000,.

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

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

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

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

[Huang et al., 2009] 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.

[Leonard et al., 1992] J. J. Leonard, H. F. Durrant-Whyte, and I. J. Cox. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(4):286–298, 1992.

[Marzorati et al., 2007] D. Marzorati, M. Matteucci, D. Migliore and D. G. Sorrenti. Integration of 3D Lines and Points in 6DoF Visual SLAM by Uncertain Projective Geometry. In *Proceedings of the 2007 European Conference on Mobile Robots*, pages 96-101, Freiburg, Germany, Sep 2007.

[Newman et al., 2006] 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)*, pages 1180-1187, Orlando, Florida, USA, 15-19 May 2006.

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

[Pedraza et al., 2007] 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.

[Piegl and Tiller, 2007] L. Piegl and W. Tiller The NURBS book. 2nd ed. Berlin: Springer-Verlag 1997.

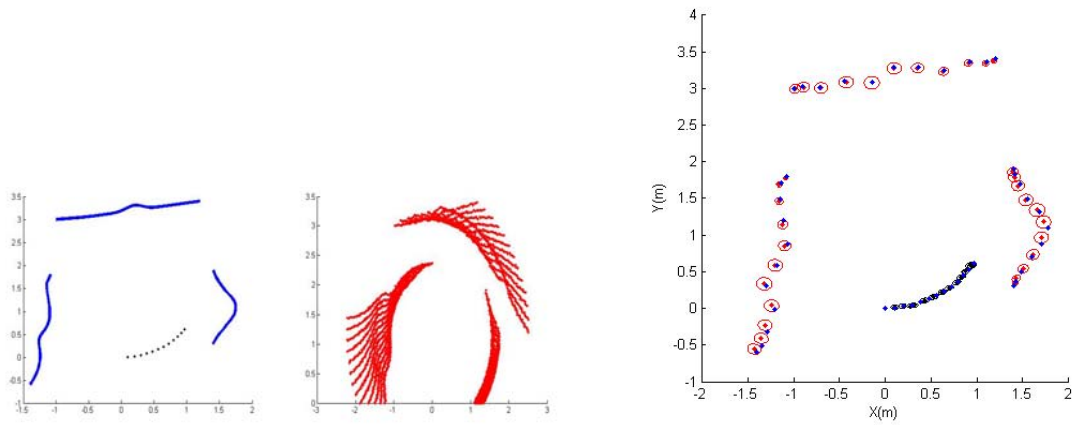
[Rodriguez-Losada et al., 2006] D. Rodriguez-Losada, F.Matia, and R.Galan. Building geometric feature based maps for indoor service robots. *Robotics and Autonomous Systems*, 54(7):546–558 2006.

[Rodriguez-Losada et al., 2006] 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)*, pages 418-423, Orlando, Florida, USA, 15-19 May 2006.

[Sweeney and Barrels, 1986] 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.

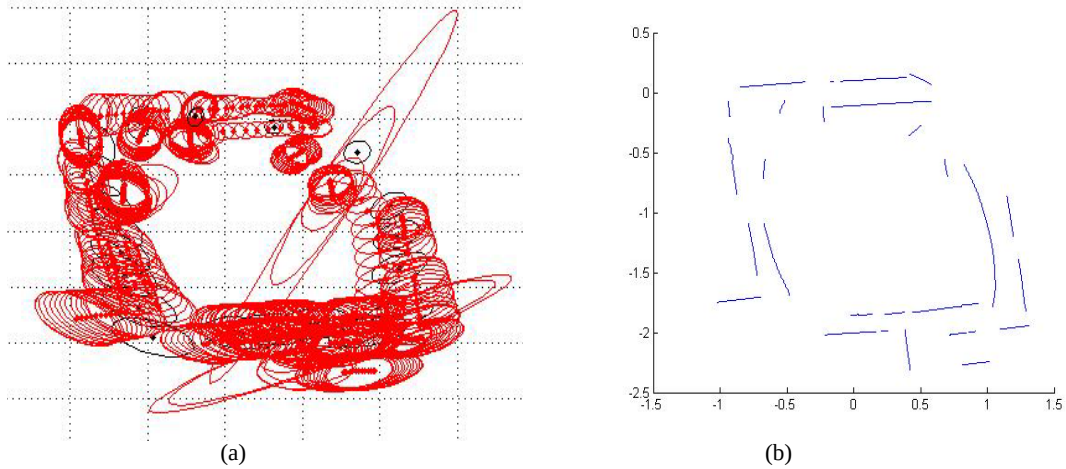
[Tardos, 1992] J. Tardos. Representing partial and uncertain sensorial information using the theory of symmetries. In *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA'92)*, 2: 1799-1804, May 1992,

[Weingarten, 2006] J. Weingarten, Feature-based 3D SLAM. Ph.D. Thesis, Lausanne, EPFL. 2006



(a) (b) (c)

Fig. 8. SLAM simulation result (a) Features and the robot trajectory used for the simulation. (b) Actual scans used for the simulation. (c) I-SLSJF SLAM result vs. “ground truth”



(a) (b)

Fig. 9. SLAM experimental result using Intel lab data (a) Control points and robot pose estimate. (b) Spline derived from the estimated control points.