Extending the Limits of Feature-Based SLAM With B-Splines

Luis Pedraza, Diego Rodriguez-Losada, Fernando Matía, Gamini Dissanayake, and Jaime Valls Miró

Abstract

This paper describes a simultaneous localization and mapping algorithm for use in unstructured environments that is effective regardless of the geometric complexity of the environment. Features are described using B-splines as modeling tool, and the set of control points defining their shape is used to form a complete and compact description of the environment, thus making it feasible to use an extended Kalman filter based SLAM algorithm. This method is the first known EKF-SLAM implementation capable of describing general free-form features in a parametric manner. Efficient strategies for computing the relevant Jacobians, perform data association, initialization and map enlargement are presented. The algorithms are evaluated for accuracy and consistency using computer simulations, and for effectiveness using experimental data gathered from different real environments.

Index Terms

Mobile robots, Kalman filtering, spline functions, simultaneous localization and mapping (SLAM).

Manuscript received December 17, 2007. This work was supported in part by the Dirección General de Investigación of Spain under Project Robonauta: DPI2007-66846-C02-01.

L. Pedraza (corresponding author), D. Rodriguez-Losada and F. Matía are with the Intelligent Control Group, Universidad Politécnica de Madrid, C/ José Gutiérrez Abascal 2, E-28006 Madrid, Spain (e-mail: luis.pedraza@ieee.org).

G. Dissanayake and J. Valls Miró are with with the Mechatronics and Intelligent Systems Group, Faculty of Engineering, University of Technology, Sydney, NSW 2007, Australia.

Extending the Limits of Feature-Based SLAM With B-Splines

I. Introduction

One of the current key challenges of the simultaneous localization and mapping (SLAM) problem, is the development of appropriate parameterizations to represent environments of increasing complexity. While a substantial body of literature exists in methods for representing unstructured environments, most of them are not suitable for use in one of the common frameworks developed during the past decades, as it is the solution based on an extended Kalman filter (EKF). For example, the use of occupancy grids [1], based on dividing the environment into small cells of predefined size, and classifying them as occupied or not, and its many variants, would result in an impracticably large state vector. The importance and effectiveness of these techniques is undeniable, but we propose a different approach to the problem.

Much of the early SLAM work relied on simple point features for describing the environment [2], [3], [4]. While this approach simplifies the formulation of the SLAM estimator, two main disadvantages arise when relying solely on this representation. The first and obvious problem emerges when the environment does not have sufficient structure to robustly extract point features; for example in an underground mine [5]. The second and more significant issue is the fact that only a small fraction of information available from popular sensors such as laser range finders is exploited. Much of the data that do not correspond to the expected features are discarded.

A number of SLAM algorithms that do not use use a specific geometric model have recently emerged. Here, the complete robot trajectory is present in the state vector, and the raw sensor information from different robot poses is processed to obtain an accurate relationship between these poses [6]. Equivalent versions of fastSLAM that exploits complete laser scans have also been presented [7], [8]. While these strategies have been successfully used to generate accurate visualizations of complex structures and detailed maps of the environments, they can not exploit the inherent information gain available in feature based SLAM, where map quality is increased due to precise mathematical description of features, stochastic information relating all of them is available, and the interpretability of the map is increased.

Several strategies for incorporating a larger fraction of the information gathered by the sensors, using more complex geometric primitives, have recently been reported. The use of line segments [9], [10], [11], and polylines [12], while promising, raise some issues related to the consistency of the solution and require further work. While segment based solutions have been successful in typical indoor environments, the increasing presence of curved geometries, more popular every day in modern constructions, can create significant problems. In particular, attempting to interpret information using an incorrect geometric model is one of the major causes of failure of many estimation algorithms. Some efforts have been made when circle features are available [13], but that is still a major simplification. Thus more generic representations of the environment can potentially improve the robustness, effectiveness and reliability of the SLAM implementations.

A hybrid strategy consists of modeling the region surrounding a point feature using a shape model in a coordinate frame attached to the feature [14], allowing the point features map to be updated while the shape models are only used to increase the information content of the observation. A full parameterization which captures all the information available in the observations, able to be updated in a statistically consistent manner, remains an interesting challenge.

Recently, we presented a novel feature based solution to the SLAM problem, based on the utilization of B-spline curves to represent the boundary between occupied and unoccupied regions in complex environments [15]. B-splines provide naturally compact descriptions for both straight and curved geometries, consisting of a set of control points defining their shape. These points, grouped in a state vector, which conveys no angular information for the static elements of the map, fully describe the environment. In [15], computationally efficient strategies for (a) initializing and extending the state vector, (b) formulating a suitable observation equation, and (c) evaluation of appropriate Jacobians for easy implementation of EKF equations, were introduced.

This work extends those results presenting additional experimental evidences, using both real and simulated data, making an emphasis on the importance of some critical aspects: (i) a study of the impact of the defining elements of B-spline curves on the expected performance of the algorithm, (ii) provide some general guidelines to adequately choose the parameters involved in the segmentation process, and (iii) a deep consistency analysis, showing the limitations of the algorithm, as occurs with any other EKF-SLAM implementation [16].

The paper is organized as follows. Section II introduces fundamental concepts regarding the extense theory of B-spline curves. Section III shows how these powerful tools naturally fit into the EKF-SLAM framework. Finally, extensive experimental results and conclusions are presented in sections IV and V.

II. Basic theory of splines

In this section, some fundamental concepts of the B-splines theory are presented. The term *spline* refers to a wide class of functions that are used in a range of technical and scientific applications, where interpolation or smoothing of noisy data in a flexible yet computationally efficient way is required. A spline of degree κ (order $\kappa - 1$) is a piecewise polynomial curve; i.e. a curve composed by several polynomial pieces of degree κ. Their most common representation is based on the linear combination of a special type of basis functions, known as B-spline basis functions.

A. B-Splines Definition

Letting $\mathbf{s}(t)$ be the position vector along a m-dimensional curve as a function of the parameter $t \in \Re$, a spline curve of order κ, with control points $\mathbf{x}_i \in \Re^m \, (i = 0 \ldots n)$ and knot vector $\Xi = \{\xi_0, \ldots, \xi_{n+k}\}$ can be expressed as:

$$\mathbf{s}(t) = \sum_{i=0}^{n} \mathbf{x}_i \beta_{i,\kappa}(t) \tag{1}$$

being $\beta_{i,\kappa}(t)$ the normalized B-spline basis functions of order κ. These functions are defined by the Cox-de Boor recursion formulae [17], [18]:

$$\beta_{i,1}(t) = \begin{cases} 1 & \text{if } \xi_i \le t < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and for all $\kappa > 1$

$$\beta_{i,\kappa}(t) = \frac{(t - \xi_i)}{\xi_{i+\kappa-1} - \xi_i} \beta_{i,\kappa-1}(t) + \frac{(\xi_{i+\kappa} - t)}{\xi_{i+\kappa} - \xi_{i+1}} \beta_{i+1,\kappa-1}(t) \tag{3}$$

The knot vector Ξ is any nondecreasing sequence of real numbers ($\xi_i \le \xi_{i+1}$ for $i = 0, \ldots, n + \kappa - 1$) and its structure can lie in two different categories: *clamped*, when the multiplicity of the extreme knot values is equal to the order κ of the curve, and *unclamped* when this does not occur [17], [19]. When clamped knot vectors are used, first and last control points are coincident, respectively, with the beginning and end of the spline curve, as Fig. 1 illustrates.

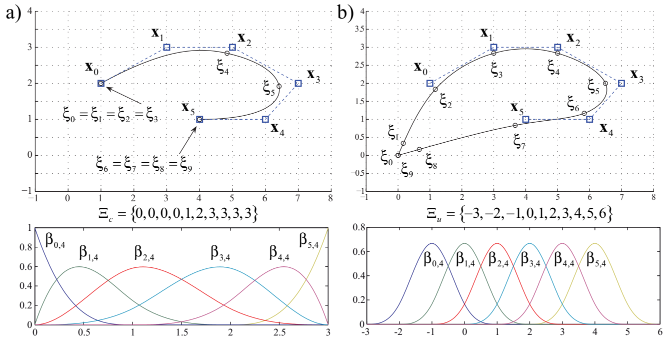


Fig. 1. Examples of cubic splines ($\kappa = 4$) and their corresponding basis functions using **a)** a *clamped* knot vector ($\Xi_c : \xi_0 = \ldots = \xi_3 \le \ldots \le \xi_6 = \ldots = \xi_9$), **b)** an *unclamped* knot vector ($\Xi_u : \xi_0 \le \ldots \le \xi_9$). Knots locations are represented as circles and control points as squares. The dashed blue line is the spline control polygon, which joins all the control points.

B. Properties of Spline Curves

B-splines exhibit many interesting mathematical and geometrical properties. For their special significance in following discussions, some of them are enumerated here:

1) The maximum order of the curve equals the number of control points. Equivalently, the minimum number of control points equals the order of the curve. This means, for example, that a cubic B-spline requires at least 4 control points to be defined.
2) The curve generally follows the shape of the control polygon.
3) Any affine transformation is applied to the curve by applying it to the control polygon.
4) Each basis function $\beta_{i,\kappa}(t)$ is a piecewise polynomial of order κ with breaks $\xi_i, \ldots, \xi_{i+\kappa}$ that vanishes outside the interval $[\xi_i, \xi_{i+\kappa})$ and is positive on the interior of that interval:

$$\beta_{i,\kappa}(t) > 0 \quad \Leftrightarrow \quad \xi_i \le t < \xi_{i+\kappa} \tag{4}$$

5) As a consequence of property 4, the value of $\mathbf{s}(t)$ at a given parameter location $\xi_j \le t \le \xi_{j+1}$ for some $j \in \{\kappa - 1, \ldots, n\}$ depends only on κ of the coefficients:

$$\mathbf{s}(t) = \sum_{i=j-\kappa+1}^{j} \mathbf{x}_i \beta_{i,\kappa}(t) \tag{5}$$

6) When clamped knot vectors are used, the sum of all the B-spline basis functions for any value of the parameter t is 1:

$$\sum_{i=0}^{n} \beta_{i,\kappa}(t) = 1 \tag{6}$$

7) The derivative of a spline of order κ is a spline of one order $\kappa - 1$. The control points of the derived function can be obtained by differencing the original ones [18].

$$\frac{d\mathbf{s}(t)}{dt} = \mathbf{s}'(t) = (\kappa - 1) \sum_{i=0}^{n} \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{\xi_{i+\kappa-1} - \xi_i} \beta_{i,\kappa-1}(t) \tag{7}$$

For further information and justification of these properties, please see [17], [18], [19] and [20].

C. Curve Fitting

One of the greatest appeals of splines curves, is their ability for approximating noise-contaminated data. In this section we consider the problem of obtaining a spline curve that fits a set of data points $\mathbf{d}_j$, $j = 0 \ldots m$. If a data point lies on the spline curve, then (1) must be satisfied:

$$\mathbf{d}_j = \beta_{0,\kappa}(t_j)\,\mathbf{x}_0 + \ldots + \beta_{n,\kappa}(t_j)\,\mathbf{x}_n, \quad j = 0 \ldots m$$

This system of equations can be more compactly written as

$$\mathbf{d} = \mathbf{B}\mathbf{x} \left\{ \begin{array}{lcl} \mathbf{d} & = & \begin{bmatrix} \mathbf{d}_0 & \mathbf{d}_1 & \ldots & \mathbf{d}_m \end{bmatrix}^T \\ \mathbf{x} & = & \begin{bmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \ldots & \mathbf{x}_n \end{bmatrix}^T \\ \mathbf{B} & = & \begin{bmatrix} \beta_{0,\kappa}(t_0) & \ldots & \beta_{n,\kappa}(t_0) \\ \vdots & \ddots & \vdots \\ \beta_{0,\kappa}(t_m) & \ldots & \beta_{n,\kappa}(t_m) \end{bmatrix} \end{array} \right. \tag{8}$$

Matrix $\mathbf{B}$ is usually referred to as the *collocation matrix*. For each of its rows has at most κ non-null values (recall property 4). The parameter value t_j defines the position of each data point $\mathbf{d}_j$ along the curve, and can be approximated by the cumulated chord length between consecutive data points:

$$\left.\begin{array}{lcl} t_0 & = & 0 \\ t_j & = & \sum_{s=1}^{j} \left\| \mathbf{d}_s - \mathbf{d}_{s-1} \right\|, \quad j \geq 1 \end{array}\right\} \tag{9}$$

being $\|\cdot\|$ the euclidean norm. The total length of the curve is

$$\ell = \sum_{s=1}^{m} \left\| \mathbf{d}_s - \mathbf{d}_{s-1} \right\| \tag{10}$$

which is taken as the maximum value of the knot vector.

When fitting noisy data acquired by a laser range finder, the most general case occurs for $2 \leq \kappa \leq n+1 < m+1$; the problem is overspecified and a least squares solution can be obtained using the pseudo inverse matrix of $\mathbf{B}$:

$$\mathbf{x} = \left[\mathbf{B}^T \mathbf{B} \right]^{-1} \mathbf{B}^T \mathbf{d} = \mathbf{\Phi} \mathbf{d} \tag{11}$$

If the order of the spline curve is predefined, the number of control points $n+1$ (or equivalently the number of knots $n+\kappa+1$), and the parameter values along the curve are known as indicated in (9), then the basis functions $\beta_{i,\kappa}(t_j)$ and hence the matrix $\mathbf{B}$ can be obtained.

In BS-SLAM, clamped knot vectors are generated taking the total length of the curve ℓ, and defining a knot spacing ρ which depends on the complexity of the environment. Recall that knots are the joints of the individual polynomial pieces a spline is composed of, so complex environments containing objects with high curvatures need a small spacing (high knot density), while straight features can be described properly using longer polynomial pieces (lower knot density).

To show the effects of both the curve order and the knot spacing on the quality of the fitting, we performed experiments with a simulated static robot in front of a curved wall. This unique feature was described by the polar equation:

$$z = 5 + 0.5 \cdot sin\left(5\varphi\right) \tag{12}$$

where z is the distance in meters from the curve to the origin, and φ is the polar angle. The simulated range-bearing sensor had a forward-facing 180° field-of-view and maximum range of 8 meters. For each measure across the angular range of the sensor, with a resolution of 1, synthetic noise with normal distribution $N\left(0, \sigma_L = 5\text{mm}\right)$ was added to the ideal measure computed using (12).

Fig. 2 shows the effect of varying the knot spacing used in the data fitting with a cubic spline. For each experiment, the average value of the mean squared error (MSE) over 50 Monte Carlo experiments was computed. The residuals are the differences between the real distance from the sensor to the wall and the distance from the sensor to the obtained spline for the same laser beam. It is clear that, as the number of polynomial pieces increases, the approximation is better.

Fig. 3 shows similar experiments for different orders of the spline used in the fitting process. Here the knot spacing is constant ($\rho = 2.92$ m). The average MSE over 50 runs of a Monte Carlo simulation is clearly smaller for the cubic spline given a fixed knot density. Better results could be obtained for the quadric spline by decreasing the knot density. However, for each additional knot a new control point should be introduced, increasing the number of necessary elements for describing this shape as compared to the cubic spline.

In general, cubic splines offer a fairly good compromise between mathematical complexity and geometric flexibility, and are the most widely used in technical and scientific applications. Most of the experimental results presented in this paper will use cubic splines for modeling the environment. However, it will be shown how easily linear splines (order 2) can be used for constructing maps of environments with flat features. The interested reader is referred to [21], [20], [19], where more information about curve fitting methods can be found.

III. Solving the SLAM Problem with B-Splines

In this section, all the procedures and formulae, necessary to make the splines theory fit into the EKF-SLAM framework are described in detail. Firstly, a simple though effective segmentation mechanism is presented, followed by the description of a robust method to perform the always delicate process of data association. Finally, suitable state and observation models are developed, and the necessary Jacobians for applying an EKF based SLAM algorithm, and building the map as new areas of the environment are explored, are presented.

A. Laser Scan Segmentation

The most commonly exterioceptive sensor used in mobile robotics, for its properties of accuracy, speed and resolution, is the laser range-finder. This sensor provides for each observation a set of m data points $\mathbf{d}_i \in \Re^2$ (bounding the problem to a 2D scenario). In this section, the necessary procedures for extracting a set of splines representing the detected physical objects are presented.

The main difference between the SLAM methodology presented in this paper, and traditional feature-based algorithms, is that we are not relying on a specific geometry to be detected. We attempt to describe the environment as accurately as possible, making no assumptions or dangerous simplifications. This segmentation methodology is based on the analysis of the relative positions of two consecutive laser data points. We define a set of $m-1$ vectors connecting the raw data (see Fig. 4):

$$\mathbf{p}_i = \mathbf{d}_i - \mathbf{d_{i-1}} \tag{13}$$

Then, the following comparisons are performed:

$$|\alpha_i| \leq \alpha_{max} \Leftrightarrow cos\left(\alpha_i\right) \geq cos\left(\alpha_{max}\right) \tag{14}$$

$$max\left(\left\|\mathbf{p}_i\right\|, \left\|\mathbf{p}_{i+1}\right\|\right) \leq \eta \cdot min\left(\left\|\mathbf{p}_i\right\|, \left\|\mathbf{p}_{i+1}\right\|\right) \tag{15}$$

Typically, $\alpha_{max} \in [0, \pi/4]$ and $\eta \in [1.5, 2]$ are fairly good values. When a set of m_F consecutive data points accomplish with both previous relationships, they are assumed as belonging to the same feature, and a fitting process is performed as described in section II-C. Additional restrictions can eventually be imposed, as demanding a minimum number of data points to be fitted, or a minimum total length for the obtained curve. This two last additional restrictions usually become indispensable for rejecting dynamic objects, as people, avoiding their inclusion in the map.

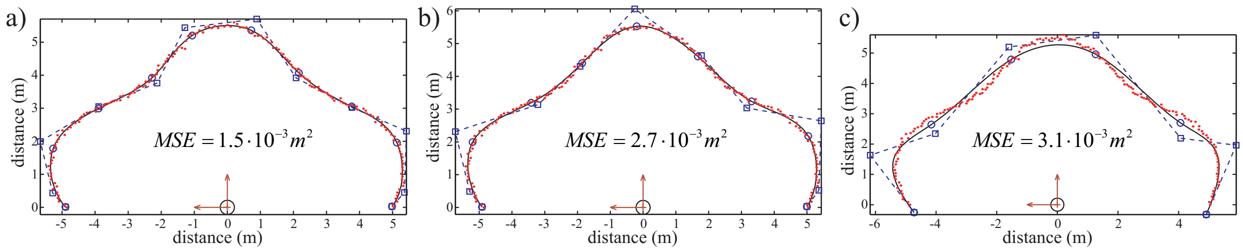


Fig. 2. Fitting of noisy data with cubic cubic splines using different spacings for the knot vector generation. The total length of the curved wall is 20,42 m. In each case the B-spline is composed by a) 9 ($\rho_a = 2,27m$), b) 8 ($\rho_b = 2,55m$), and c) 5 ($\rho_c = 4,1m$) polynomial pieces.

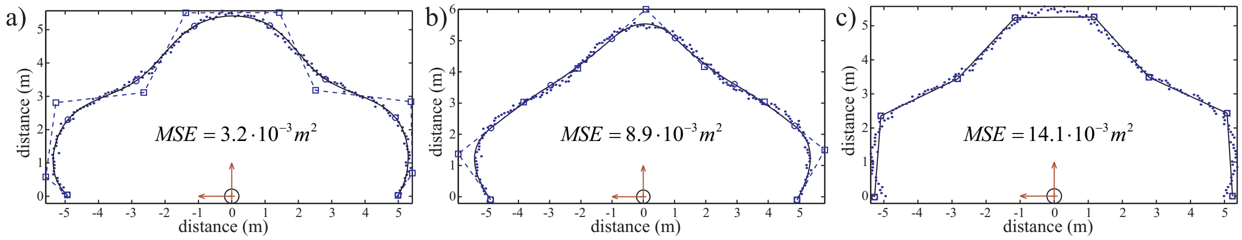


Fig. 3. Fitting of noisy data with a) cubic, b) quadratic and c) linear B-splines.

Fig. 4.a illustrates the process with a laser sample obtained from a real data set. In this case, a total number of 4 features have been detected. Note that F_2 and F_4 would be certainly difficult to describe with traditional feature based map representations, as segments or even circles. Fig. 4.b and Fig. 4.c show the effects of choosing incorrect values for α_{max}. High values of this angular threshold produce the incorrect identification of important locations, such as corners, whereas too small values produce excessive segmentation, and the wasting of many data points that are not assigned to any feature.

B. Data Association With Splines

At each sampling time, a new set of splines is obtained as described before, being necessary to establish a correspondence between the N features contained in the map ($\mathbf{s}_{m,1}, \ldots, \mathbf{s}_{m,N}$), and the N_o features detected by the robot ($\mathbf{s}_{o,1}, \ldots, \mathbf{s}_{o,N_o}$).

Possibly the main drawback of using splines as modeling tool, is that control points are not observable. For any given geometry, there are infinite ways of describing its shape depending on the chosen knot vector and the particular area of the feature detected by the sensors.

The proposed matching process is described with the help of Fig. 5 and the pseudocode in Fig. 6. First, the distances from the control points of each of the detected splines to the control points of the splines contained in the map are calculated (map can be simplified choosing only features that are close to the robot position). This way, couples of matching candidates are identified by simple calculation of the euclidean distance between their control points. A distance threshold of half the knot spacing ($d_{mincp} = \rho/2$) is enough for this preliminary association.

If a spline is close enough to a map feature, then it is necessary to obtain a matching between their points, as depicted in Fig. 5.b. The process is as follows:

- One of the end points of the observed spline is considered (point $\mathbf{a}$)
- The closest point on the map spline to the point $\mathbf{a}$ is calculated (point $\mathbf{b}$)
- If $\mathbf{b}$ is one of the end points of the map spline, then, the closest point to $\mathbf{b}$ on the observed spline is calculated (point $\mathbf{c}$).
- The process is repeated using as starting point the other extreme of the observed spline (point $\mathbf{d}$ in the picture, which is associated with point $\mathbf{e}$).

Once checked the last final conditions in Fig. 6 (avoiding, for example, matching an observation with the hidden side of a wall), the association is established. At the end of this process, not only correspondent pairs of points $(\mathbf{c} \equiv \mathbf{s}_{o,1}(u_{ini}), \mathbf{b} \equiv \mathbf{s}_{m,1}(t_{ini}))$ and $(\mathbf{d} \equiv \mathbf{s}_{o,1}(u_{fin}), \mathbf{e} \equiv \mathbf{s}_{m,1}(t_{fin}))$ are obtained, but also a correspondence between the map spline parameter t and the observed spline parameter u. This information is very useful when a spline extension is required, and for the observation model we propose.

The described data association process, though quite simple and based only upon euclidean distance metric, has performed very robustly in our experiments.

C. The State Model

The state of the system is composed by the robot pose (the only non-static element) and all the map features, which are modeled as B-spline curves. When a spline curve is expressed as linear combination of B-splines, its state can be described by the positions of its control points. Remember from section II that a spline is defined by both the control points and the knot vector, but this last element is considered fixed, defined during the feature initialization stage, and only modified when the element is enlarged, as will be described in section III-F.

Referring all the positions and orientations to a global reference frame $\{\mathbf{u}_W, \mathbf{v}_W\}$, and letting the robot be the

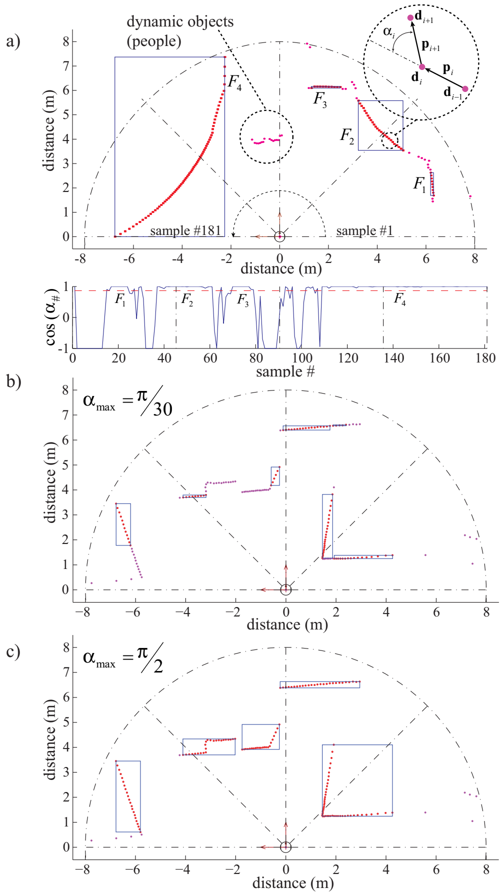


Fig. 4. Segmentation of raw laser scan data. a) Most of the information can be used in the estimation process $\alpha_{max} = \pi/4$. b) and c) Segmentation errors due to excessively small or large values for α_{max}

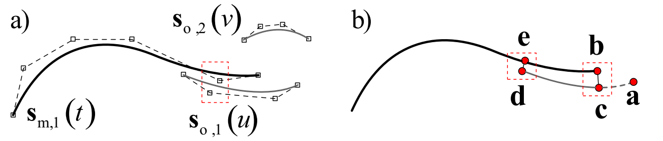


Fig. 5. Example of data association process. **a)** Comparison of control points positions and **b)** parameters correspondence.

first feature in the map (F_0) the following expressions fully describe the state of the system at a certain time k:

$$\mathbf{x}_{F_0} = \mathbf{x}_r = [x_r, y_r, \phi_r]^T \tag{16}$$

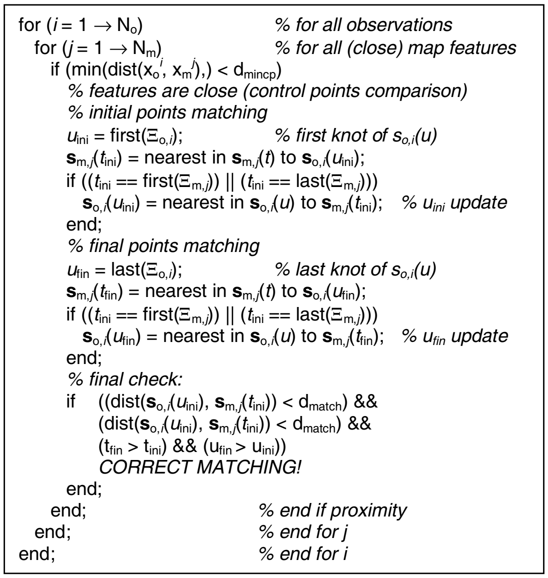
$$\mathbf{x}_{F_i} = \mathbf{x}_{s_i} = [x_{i,0}, \ldots, x_{i,n_i}, y_{i,0}, \ldots, y_{i,n_i}]^T \tag{17}$$

$$i = 1, \ldots, N$$

and finally

$$\mathbf{x} = \left[\mathbf{x}_r^T, \mathbf{x}_{s_1}^T, \ldots, \mathbf{x}_{s_N}^T\right]^T \tag{18}$$

In the previous equations, N is the number of map static elements, defined each of them by $n_i + 1$ control points. Note that the number of control points for each of the splines contained in the map is not bound to be fixed. Splines can be prolonged, their knot vectors can be extended and, therefore, new control points can eventually be inserted in the map when new areas of the environment are explored, as we will see further on. The state of the system is assumed to follow a normal random distribution with mean value



```
for ($i$ = 1 → N$_o$)                          % for all observations
  for ($j$ = 1 → N$_m$)                        % for all (close) map features
    if (min(dist($x_o^i$, $x_m^j$),) < d$_{mincp}$)
      % features are close (control points comparison)
      % initial points matching
      $u_{ini}$ = first($\Xi_{o,i}$);                % first knot of $s_{o,i}(u)$
      $\mathbf{s}_{m,j}(t_{ini})$ = nearest in $\mathbf{s}_{m,j}(t)$ to $\mathbf{s}_{o,i}(u_{ini})$;
      if (($t_{ini}$ == first($\Xi_{m,j}$)) || ($t_{ini}$ == last($\Xi_{m,j}$)))
        $\mathbf{s}_{o,i}(u_{ini})$ = nearest in $\mathbf{s}_{o,i}(u)$ to $\mathbf{s}_{m,j}(t_{ini})$;   % $u_{ini}$ update
      end;
      % final points matching
      $u_{fin}$ = last($\Xi_{o,i}$);                 % last knot of $s_{o,i}(u)$
      $\mathbf{s}_{m,j}(t_{fin})$ = nearest in $\mathbf{s}_{m,j}(t)$ to $\mathbf{s}_{o,i}(u_{fin})$;
      if (($t_{ini}$ == first($\Xi_{m,j}$)) || ($t_{ini}$ == last($\Xi_{m,j}$)))
        $\mathbf{s}_{o,i}(u_{fin})$ = nearest in $\mathbf{s}_{o,i}(u)$ to $\mathbf{s}_{m,j}(t_{fin})$;   % $u_{fin}$ update
      end;
      % final check:
      if    ((dist($\mathbf{s}_{o,i}(u_{ini})$, $\mathbf{s}_{m,j}(t_{ini})$)) < d$_{match}$) &&
            (dist($\mathbf{s}_{o,i}(u_{ini})$, $\mathbf{s}_{m,j}(t_{ini})$)) < d$_{match}$) &&
            ($t_{fin} > t_{ini}$) && ($u_{fin} > u_{ini}$))
            CORRECT MATCHING!
      end;
    end;                                     % end if proximity
  end;                                       % end for j
end;                                         % end for i
```

Fig. 6. Pseudocode for data association.

$$\hat{\mathbf{x}}(k|k) = \left[\begin{array}{cccc} \hat{\mathbf{x}}_r(k|k) & \hat{\mathbf{x}}_{s_1}(k|k) & \ldots & \hat{\mathbf{x}}_{s_N}(k|k) \end{array}\right] \tag{19}$$

and coavariance matrix

$$\mathbf{P}(k|k) = \begin{bmatrix} \mathbf{P}_{rr}(k|k) & \mathbf{P}_{rs_1}(k|k) & \ldots & \mathbf{P}_{rs_N}(k|k) \\ \mathbf{P}_{s_1r}(k|k) & \mathbf{P}_{s_1s_2}(k|k) & \ldots & \mathbf{P}_{s_1s_N}(k|k) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{s_Ns_1}(k|k) & \mathbf{P}_{s_Ns_2}(k|k) & \ldots & \mathbf{P}_{s_Ns_N}(k|k) \end{bmatrix} \tag{20}$$

D. The Observation Model

Using an Extended Kalman Filter for solving the SLAM problem requires an observation model; i.e. some expression which allows to predict the measurements that are likely to be obtained by the robot sensors given the robot pose and the current knowledge of the environment. As mentioned before, control points of map splines are not observable.

However, it is still possible to predict every single laser measurement for each position of the laser beam across its angular range. This way, the problem is reduced to the calculation of the intersection of the straight line defined by a laser beam (for each angular position p) with the splines contained in the map.

Unfortunately, calculating the intersection of a straight line with a parametric curve, in the form $\mathbf{s}(t) = [s_x(t), s_y(t)]^T$ is not suitable for an explicit mathematical formulation. This problem is known in the literature as *ray tracing* [22], and is schematically depicted in Fig. 7.

In BS-SLAM, the predicted measurement is calculated making use of the following two elements:

- Property 3 in section II-B, which states that any affine transformation can be applied to a spline curve by applying it to its control points.

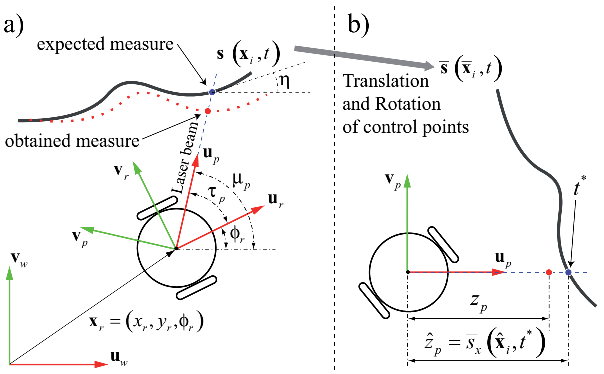


Fig. 7. Observation model. The expected intersection of each laser beam across the angular range of the sensor with the map spline is computed expressing the map spline (a) in the $\{\mathbf{u}_p,\mathbf{v}_p\}$ reference frame (b).

- The Newton-Raphson method, for calculating the roots of a function.

The first step is to define an orthonormal reference frame $\{\mathbf{u}_p,\mathbf{v}_p\}$, centered in the robot reference frame $\{\mathbf{u}_r,\mathbf{v}_r\}$ and with $\mathbf{u}_p$ defined by the laser beam direction and orientation (see Fig. 7depicting an arbitrary laser beam and its intersection with a spline curve that needs to be computed). Let $\bar{\mathbf{s}}\left(\bar{\mathbf{x}}_i\left(\mathbf{x}_i,\mathbf{x}_r\right),t\right)$ be the position vector along a spline curve expressed in such a reference system (we are making here explicit the functional dependency between a spline and its control points). The relationship between control points $\mathbf{x}_i=\left[x_i,y_i\right]^T$ and $\bar{\mathbf{x}}_i=\left[\bar{x}_i,\bar{y}_i\right]^T$, $i=0\ldots n$, is given by:

$$\begin{bmatrix}\bar{x}_i\\ \bar{y}_i\end{bmatrix}=\begin{bmatrix}cos\mu_p & sin\mu_p\\ -sin\mu_p & cos\mu_p\end{bmatrix}\begin{bmatrix}x_i-x_r\\ y_i-y_r\end{bmatrix}\qquad(21)$$

where μ_p is the angle of the considered laser beam in the global reference frame.

In this context, the measurement prediction $\hat{z}_p=h\left(\mathbf{x}_i,\mathbf{x}_r\right)$ is given by $\bar{s}_x\left(\bar{x}_i\left(\mathbf{x}_i,\mathbf{x}_r\right),t^*\right)$, where t^* is the value of the parameter t that makes $\bar{s}_y\left(\bar{y}_i\left(\mathbf{x}_i,\mathbf{x}_r\right),t^*\right)=0$.

Despite the lack of an explicit observation model, it is possible to compute its derivatives with respect to the state in an approximate way. Once calculated the value t^* which makes $\bar{s}_y\left(t^*\right)=0$, the expected measurement in the nearness of this parameter location, assuming small perturbations in the state vector, can be approximated by (subindex p is omitted in the following equations):

$$h\left(\mathbf{x}_i,\mathbf{x}_r\right)=\bar{s}_x\left(\bar{x}_i\left(\mathbf{x}_i,\mathbf{x}_r\right),t^*-\frac{\bar{s}_y\left(\bar{y}_i\left(\mathbf{x}_i,\mathbf{x}_r\right),t^*\right)}{\bar{s}'_y\left(\bar{y}_i\left(\mathbf{x}_i,\mathbf{x}_r\right),t^*\right)}\right)\qquad(22)$$

Derivating with respect to the control points positions, and making use of (1) and (21) we can write [15]:

$$\frac{\partial h}{\partial x_i}=\beta_{i,k}\left(t^*\right)\left[cos\mu+\frac{sin\mu}{\tan\left(\eta-\mu\right)}\right]\qquad(23)$$

$$\frac{\partial h}{\partial y_i}=\beta_{i,k}\left(t^*\right)\left[sin\mu-\frac{cos\mu}{\tan\left(\eta-\mu\right)}\right]\qquad(24)$$

Similarly, making use of property 6, (1) and (21), we obtain:

$$\frac{\partial h}{\partial x_r}=-cos\mu-\frac{sin\mu}{\tan\left(\eta-\mu\right)}\qquad(25)$$

$$\frac{\partial h}{\partial y_r}=-sin\mu+\frac{cos\mu}{\tan\left(\eta-\mu\right)}\qquad(26)$$

$$\frac{\partial h}{\partial \phi_r}=\frac{\hat{z}}{\tan\left(\eta-\mu\right)}\qquad(27)$$

These formulas will allow the efficient calculation of the relevant Jacobians in the following sections.

E. Applying the EKF

In this section, all previously obtained results are combined in the working frame of the Extended Kalman Filter [2], allowing the incremental building of the map of an environment, where all features are modeled using cubic splines.

1) Kalman Filter Prediction: Between the times k and $k+1$ the robot makes a relative movement, given by the stochastic variable

$$\mathbf{u}\left(k+1\right)\sim N\left(\hat{\mathbf{u}}\left(k+1\right),\mathbf{Q}\left(k+1\right)\right)\qquad(28)$$

Under the hypothesis that the only moving object in the map is the robot, the a priori estimation of the state at time $k+1$ is given by:

$$\hat{\mathbf{x}}_r\left(k+1|k\right)=\mathbf{f}_r\left(\hat{\mathbf{x}}_r\left(k|k\right),\hat{\mathbf{u}}\left(k+1\right)\right)\qquad(29)$$

$$\hat{\mathbf{x}}_{s_i}\left(k+1|k\right)=\hat{\mathbf{x}}_{s_i}\left(k|k\right)\qquad(30)$$

where $\mathbf{f}_r$ is the motion model, which depends on the mobile platform being used, and its covariance:

$$\begin{aligned}\mathbf{P}\left(k+1|k\right)=\mathbf{F}_x\left(k+1\right)\mathbf{P}\left(k|k\right)\mathbf{F}_x^T\left(k+1\right)+\\+\mathbf{F}_u\left(k+1\right)\mathbf{Q}\left(k+1\right)\mathbf{F}_u^T\left(k+1\right)\end{aligned}\qquad(31)$$

The Jacobian matrices are

$$\mathbf{F}_x\left(k+1\right)=\begin{bmatrix}\left.\frac{\partial\mathbf{f}_r}{\partial\mathbf{x}_r}\right|_{\hat{\mathbf{x}}_r(k|k),\hat{\mathbf{u}}(k+1)} & \mathbf{0} & \ldots & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_{n_1} & \ldots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{I}_{n_N}\end{bmatrix}\qquad(32)$$

$$\mathbf{F}_u\left(k+1\right)=\begin{bmatrix}\left.\frac{\partial\mathbf{f}_r}{\partial\mathbf{u}}\right|_{\hat{\mathbf{x}}_r(k|k),\hat{\mathbf{u}}(k+1)}\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\end{bmatrix}\qquad(33)$$

2) Kalman Filter Update: Once obtained the expected measurements for each of the laser beams positions of an observation associated with a map spline, the innovation covariance matrix is given by [2]:

$$\mathbf{S}\left(k+1\right)=\mathbf{H}_\mathbf{x}\left(k+1\right)\mathbf{P}\left(k+1|k\right)\mathbf{H}_\mathbf{x}^T\left(k+1\right)+\mathbf{R}\left(k+1\right)\qquad(34)$$

where $\mathbf{R}$ is the sensor covariance matrix, and the Jacobian is:

$$\mathbf{H}_\mathbf{x}\left(k+1\right)=\begin{bmatrix}\frac{\partial\mathbf{h}}{\partial\mathbf{x}_r} & \mathbf{0} & \ldots & \mathbf{0} & \frac{\partial\mathbf{h}}{\partial\mathbf{x}_{s_i}} & \mathbf{0} & \ldots & \mathbf{0}\end{bmatrix}\qquad(35)$$

In the previous equation, the term $\frac{\partial \mathbf{h}}{\partial \mathbf{x}_r}$ is calculated making use of (25), (26) and (27), and $\frac{\partial \mathbf{h}}{\partial \mathbf{x}_{s_i}}$ is calculated from (23) and (24). The gain matrix is calculated as follows:

$$\mathbf{W}\left(k+1\right) = \mathbf{P}\left(k+1|k\right)\mathbf{H}_{\mathbf{x}}^{T}\left(k+1\right)\mathbf{S}^{-1}\left(k+1\right) \qquad (36)$$

Finally, the state estimation and its covariance are updated according to:

$$\hat{\mathbf{x}}\left(k{+}1|k{+}1\right){=}\hat{\mathbf{x}}\left(k{+}1|k\right){+}\mathbf{W}\left(k{+}1\right)\nu\left(k{+}1\right) \qquad (37)$$

$$\mathbf{P}\left(k{+}1|k{+}1\right){=}\left[\mathbf{I}{-}\mathbf{W}\left(k{+}1\right)\mathbf{H}_x\left(k{+}1\right)\right]\mathbf{P}\left(k{+}1|k\right) (38)$$

with the innovation calculated as:

$$\nu\left(k+1\right) = \mathbf{z}\left(k+1\right) - \hat{\mathbf{z}}\left(k+1|k\right) \qquad (39)$$

F. Map Enlargement

The stochastic map is incrementally built in two different ways: adding new objects, and extending objects already contained in the map. In this section, algorithms for both initializing new objects, and prolonging existing features as new areas are explored are described in detail.

1) Adding New Objects to the Map: Observations that cannot satisfy the association procedure described in section III-B are considered as a newly discovered features, and the splines defining their shapes must be added to the map. Given a map containing N static features, and a set of measurements $\mathbf{z} = \{z_i, i = p \dots p+q\}$ obtained for the laser angular positions in the robot reference frame (see Fig. 7) corresponding to a new feature F_{N+1}, the augmented state vector can be computed as:

$$\mathbf{x}^a = \mathbf{g}\left(\mathbf{x},\mathbf{z}\right) \Leftrightarrow \begin{cases} \mathbf{x}_r^a & = \mathbf{x}_r \\ \mathbf{x}_{s_i}^a & = \mathbf{x}_{s_i} \quad ,i = 1,\dots,N \\ \mathbf{x}_{s_{N+1}}^a & = \mathbf{g}_{s_{N+1}}\left(\mathbf{x}_r,\mathbf{z}\right) \end{cases} \qquad (40)$$

This means that the fact of adding a new object does not change the current map structure (the robot pose and the control points of the N existing features). Function $\mathbf{g}_{s_{N+1}}\left(\mathbf{x}_r,\mathbf{z}\right)$ is the fitting function of the $q+1$ new data points as described in section II-C. This way, we can obtain the control points of the new feature as a linear functioin of the data points and the robot pose.

$$\begin{bmatrix} x_{N+1,0} \\ \vdots \\ x_{N+1,n_N} \end{bmatrix} = \mathbf{\Phi}\begin{bmatrix} x_r + z_p\cos\left(\phi_r+\tau_p\right) \\ \vdots \\ x_r + z_{p+q}\cos\left(\phi_r+\tau_{p+q}\right) \end{bmatrix} \qquad (41)$$

$$\begin{bmatrix} y_{N+1,0} \\ \vdots \\ y_{N+1,n_N} \end{bmatrix} = \mathbf{\Phi}\begin{bmatrix} y_r + z_p\sin\left(\phi_r+\tau_p\right) \\ \vdots \\ y_r + z_{p+q}\sin\left(\phi_r+\tau_{p+q}\right) \end{bmatrix} \qquad (42)$$

The new covariance matrix for the augmented state vector is:

$$\mathbf{P}^a = \mathbf{G_x}\mathbf{P}\mathbf{G_x}^T + \mathbf{G_z}\mathbf{R}\mathbf{G_z}^T \qquad (43)$$

and the Jacobians $\mathbf{G_x} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}$ and $\mathbf{G_z} = \frac{\partial \mathbf{g}}{\partial \mathbf{z}}$:

$$\mathbf{G_x} = \left[\begin{array}{cccc} \mathbf{I}_r & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I}_{n_N} \\ \hline \frac{\partial \mathbf{g}_{s_{N+1}}}{\partial \mathbf{x}_r} & \mathbf{0} & \dots & \mathbf{0} \end{array}\right], \mathbf{G_z} = \left[\begin{array}{c} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \hline \frac{\partial \mathbf{g}_{S_{N+1}}}{\partial \mathbf{z}} \end{array}\right] \qquad (44)$$

with

$$\frac{\partial \mathbf{g}_{s_{N+1}}}{\partial \mathbf{x}_r} = \left[\begin{array}{c} \mathbf{\Phi}\begin{bmatrix} 1 & 0 & -z_p\sin\mu_p \\ \vdots & \vdots & \vdots \\ 1 & 0 & -z_{p+q}\sin\mu_{p+q} \end{bmatrix} \\ \hline \mathbf{\Phi}\begin{bmatrix} 0 & 1 & z_p\cos\mu_p \\ \vdots & \vdots & \vdots \\ 0 & 1 & z_{p+q}\sin\mu_{p+q} \end{bmatrix} \end{array}\right] \qquad (45)$$

$$\frac{\partial \mathbf{g}_{s_{N+1}}}{\partial \mathbf{z}} = \left[\begin{array}{c} \mathbf{\Phi}\begin{bmatrix} \cos\mu_p & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \cos\mu_{p+q} \end{bmatrix} \\ \hline \mathbf{\Phi}\begin{bmatrix} \sin\mu_p & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sin\mu_{p+q} \end{bmatrix} \end{array}\right] \qquad (46)$$

2) Extending Map Objects: Frequently, observations are only partially associated with a map feature (as in Fig. 5.b). This means that a new unexplored part of a map object is being detected and, consequently, this spline must be extended. Take for instance the situation displayed in Fig. 8, where the j-th map spline has been partially associated with an observation, and the information contained in a new set of $m+1$ data points

$$\mathbf{d_i} = \begin{bmatrix} d_i^x \\ d_i^y \end{bmatrix} = \begin{bmatrix} x_r + z_i\cos\mu_i \\ y_r + z_i\sin\mu_i \end{bmatrix}, \quad i = q,\dots,q+m \quad (47)$$

must be integrated into the map feature. The extended state vector will be:

$$\mathbf{x}^e = \mathbf{g_e}\left(\mathbf{x},\mathbf{z}\right) \Leftrightarrow \begin{cases} \mathbf{x}_r^e & = \mathbf{x}_r \\ \mathbf{x}_{s_i}^e & = \mathbf{x}_{s_i}, \quad i \neq j \\ \mathbf{x}_{s_j}^e & = \mathbf{g}_{s_j}^e\left(\mathbf{x}_r,\mathbf{x}_j,\mathbf{z}\right) \end{cases} \qquad (48)$$

Function $\mathbf{g}_{s_j}^e\left(\mathbf{x}_r,\mathbf{x}_j,\mathbf{z}\right)$ is constructed following a similar scheme to the one used in the data fitting process.

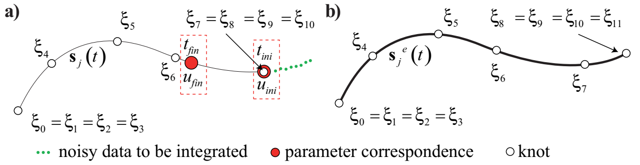


Fig. 8. Map spline extension with new data after Kalman filter update. a) The original spline $\mathbf{s}_j\left(t\right)$ needs to be unclamped and last 3 control points are displaced, allowing the insertion of the new data in the extended knot vector. b) A new knot $\xi_1 1$ is inserted at the end of the extended curve $\mathbf{s}_j^e\left(t\right)$, which is defined using a clamped knot vector.

In [19] an iterative unclamping algorithm is proposed, which is successfully applied in [23] with the goal of extending a B-spline curve to a single target points, inserting one additional control point and one additional knot for each extension. Our problem is to extend a map spline given a new set of measured data points, and maintaining the strictly necessary number of new control points bounded. The combination of the unclamping algorithm with the approximation scheme previously proposed, makes possible to extend the map features as new measurements are obtained. Given a spline curve, defined by a set of control points $\mathbf{x}_i$ and clamped knot vector in the form:

$$\Xi : \underbrace{\xi_0 = \ldots = \xi_{\kappa-1}}_{\kappa} \leq \xi_\kappa \leq \ldots \leq \xi_n \leq \underbrace{\xi_{n+1} = \ldots = \xi_{\kappa+n}}_{\kappa}$$

the unclamping algorithm proposed in [19] calculates the new control points corresponding to the unclamped knot vectors:

$$\Xi_r\text{:}\ \underbrace{\xi_0 = \ldots = \xi_{\kappa-1}}_{\kappa} \leq \xi_\kappa \leq \ldots \leq \xi_n \leq \bar{\xi}_{n+1} \leq \ldots \leq \bar{\xi}_{\kappa+n}$$

$$\Xi_l\text{:}\ \bar{\xi}_0 \leq \ldots \leq \bar{\xi}_{\kappa-1} \leq \xi_\kappa \leq \ldots \leq \xi_n \leq \underbrace{\xi_{n+1} = \ldots = \xi_{\kappa+n}}_{\kappa}$$

The particularization of the iterative algorithm for a given spline order, permits to obtain linear relationships between the old and new control points values. The trivial case occurs for linear splines (order $\kappa = 2$). Here, the control points of the unclamped curved remain unchanged.

Here we show the adaptation of the algorithm for cubic splines, but similar expressions can be obtain for other orders, if desired. The right-unclamping algorithm (on the side corresponding to higher values for the parameter t) of a cubic spline (order $\kappa = 4$, converting the knot vector Ξ into Ξ_r, and obtaining the new control points

$$\left.\begin{aligned} \mathbf{x}_i^r &= \mathbf{x}_i, \quad i = 0, \ldots, n-2 \\ \mathbf{x}_{n-1}^r &= -\Gamma_{n-1}^2 \mathbf{x}_{n-2} + \tfrac{1}{\gamma_{n-1}^2} \mathbf{x}_{n-1} \\ \mathbf{x}_n^r &= \Gamma_n^2 \Gamma_{n-1}^2 \mathbf{x}_{n-2} - \left(\tfrac{\Gamma_n^2}{\gamma_{n-1}^2} + \tfrac{\Gamma_n^1}{\gamma_n^2} \right) \mathbf{x}_{n-1} + \tfrac{1}{\gamma_n^1 \gamma_n^2} \mathbf{x}_n \end{aligned}\right\} \qquad (49)$$

being

$$\gamma_i^j = \tfrac{\bar{\xi}_{n+1} - \bar{\xi}_i}{\bar{\xi}_{i+j+1} - \bar{\xi}_i} \quad \text{and} \quad \Gamma_i^j = \tfrac{1 - \gamma_i^j}{\gamma_i^j} \qquad (50)$$

Similar results can be obtained when converting a clamped knot vector Ξ into a left-unclamped one Ξ_l:

$$\left.\begin{aligned} \mathbf{x}_0^l &= \tfrac{1}{\omega_0^1 \omega_0^2} \mathbf{x}_0 - \left(\tfrac{\Omega_0^2}{\omega_1^2} + \tfrac{\Omega_0^1}{\omega_0^2} \right) \mathbf{x}_1 + \Omega_0^2 \Omega_0^1 \mathbf{x}_2 \\ \mathbf{x}_1^l &= \tfrac{1}{\omega_1^2} \mathbf{x}_1 - \Omega_1^2 \mathbf{x}_2 \\ \mathbf{x}_i^l &= \mathbf{x}_i, \quad i = 2, \ldots, n \end{aligned}\right\} \qquad (51)$$

with

$$\omega_i^j = \tfrac{\bar{\xi}_{\kappa-1} - \bar{\xi}_{i+\kappa}}{\bar{\xi}_{i+\kappa-j-1} - \bar{\xi}_{i+\kappa}} \text{ and } \Omega_i^j = \tfrac{1 - \omega_i^j}{\omega_i^j} \qquad (52)$$

These results can be combined with the methodology proposed in section II-C for obtaining new splines as new data is acquired, being aware of the following considerations:

- New data points must be associated with the existing parameterization of the map spline. This relation can be obtained from the data association stage.
- The knot vector needs to be unclamped and might need to be extended with additional knots in order to make room for the new span being added. The number of new knots is chosen taking into account the specified knot spacing, and the final length of the curve.

This way, system (8) is written for the new data points, extended with the unclamping equations 49, 50, and/or 51, 52, and its least-squares solution provides a matrix-form linear relationship between the old control points $\mathbf{x}_j$ and the sampled data $\mathbf{d}_i$, and the new control points $\mathbf{x}_j^e$. For example, for a right-extension we can obtain:

$$\begin{bmatrix} x_{j,0}^e \\ \vdots \\ x_{j,n_j+p}^e \end{bmatrix} = \mathbf{\Phi}^{\mathbf{e}} \begin{bmatrix} d_q^x \\ \vdots \\ d_{q+m}^x \\ \hline x_{j,0} \\ \vdots \\ x_{j,n_j} \end{bmatrix}, \begin{bmatrix} y_{j,0}^e \\ \vdots \\ y_{j,n_j+p}^e \end{bmatrix} = \mathbf{\Phi}^{\mathbf{e}} \begin{bmatrix} d_q^y \\ \vdots \\ d_{q+m}^y \\ \hline y_{j,0} \\ \vdots \\ y_{j,n_j} \end{bmatrix} \qquad (53)$$

where Φ^e is a constant matrix, which depends only on the configuration of the new knot vector. The new covariance matrix after extending the j-th spline is:

$$\mathbf{P}^e = \mathbf{G}_{\mathbf{x}}^e \mathbf{P} \mathbf{G}_{\mathbf{x}}^{e\,T} + \mathbf{G}_{\mathbf{z}}^e \mathbf{R} \mathbf{G}_{\mathbf{z}}^{e\,T} \qquad (54)$$

where the involved Jacobians $\mathbf{G}_{\mathbf{x}}^e = \frac{\partial \mathbf{g}^e}{\partial \mathbf{x}}$ and $\mathbf{G}_{\mathbf{z}}^e = \frac{\partial \mathbf{g}^e}{\partial \mathbf{z}}$ have the following appearance:

$$\mathbf{G}_{\mathbf{x}}^e = \begin{bmatrix} \mathbf{I}_r & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_1} & \ldots & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ \frac{\mathbf{g}_{s_j}^e}{\partial \mathbf{x}_r} & \mathbf{0} & \ldots & \frac{\partial \mathbf{g}_{s_j}^e}{\partial \mathbf{x}_{s_j}} & \ldots & \mathbf{0} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{I}_{n_N} \end{bmatrix}, \mathbf{G}_{\mathbf{z}}^e = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \frac{\partial \mathbf{g}_{s_j}^e}{\partial \mathbf{z}} \\ \vdots \\ \mathbf{0} \end{bmatrix} \qquad (55)$$

being

$$\frac{\partial \mathbf{g}_{s_j}^e}{\partial \mathbf{x}_r} = \begin{bmatrix} \mathbf{\Phi}^e \begin{bmatrix} 1 & 0 & -z_q \sin \mu_q \\ \vdots & \vdots & \vdots \\ 1 & 0 & -z_{q+m} \sin \mu_{q+m} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\ \hline \mathbf{\Phi}^e \begin{bmatrix} 0 & 1 & z_p \cos \mu_p \\ \vdots & \vdots & \vdots \\ 0 & 1 & z_{p+q} \sin \mu_{p+q} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \end{bmatrix}, \frac{\partial \mathbf{g}_{s_j}^e}{\partial \mathbf{x}_{s_j}} = \begin{bmatrix} \mathbf{\Phi}^e \left[\begin{array}{c|c} \mathbf{0} & \mathbf{0} \\ \hline \mathbf{I}_{n_j} & \mathbf{0} \end{array}\right] \\ \hline \mathbf{\Phi}^e \left[\begin{array}{c|c} \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I}_{n_j} \end{array}\right] \end{bmatrix}$$

and

$$\frac{\partial \mathbf{g}_{s_j}^e}{\partial \mathbf{z}} = \begin{bmatrix} \mathbf{\Phi}^e \begin{bmatrix} \cos \mu_q & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \cos \mu_{q+m} \\ \hline \mathbf{0} & \ldots & \mathbf{0} \end{bmatrix} \\ \hline \mathbf{\Phi}^e \begin{bmatrix} \sin \mu_q & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \sin \mu_{q+m} \\ \hline \mathbf{0} & \ldots & \mathbf{0} \end{bmatrix} \end{bmatrix}$$

IV. Experimental Results

Several experiments have been performed with both real and simulated data in order to validate the methodologies and algorithms presented in this paper. In all the experiments, the following motion model has been used:

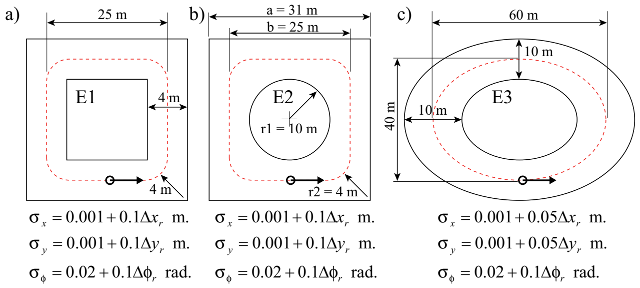

a)
25 m
E1
4 m
4 m
b)
a = 31 m
b = 25 m
E2
r1 = 10 m
r2 = 4 m
c)
60 m
10 m
E3
10 m
40 m
$\sigma_x = 0.001 + 0.1\Delta x_r$ m.
$\sigma_y = 0.001 + 0.1\Delta y_r$ m.
$\sigma_\phi = 0.02 + 0.1\Delta\phi_r$ rad.
$\sigma_x = 0.001 + 0.1\Delta x_r$ m.
$\sigma_y = 0.001 + 0.1\Delta y_r$ m.
$\sigma_\phi = 0.02 + 0.1\Delta\phi_r$ rad.
$\sigma_x = 0.001 + 0.05\Delta x_r$ m.
$\sigma_y = 0.001 + 0.05\Delta y_r$ m.
$\sigma_\phi = 0.02 + 0.1\Delta\phi_r$ rad.

Fig. 9. Synthetic environments for consistency experiments.

$$\begin{aligned}
x_{r,k+1|k} &= x_{r,k|k} + \Delta x_r cos\left(\phi_{r,k|k}\right) - \Delta y_r sin\left(\phi_{r,k|k}\right)\\
y_{r,k+1|k} &= y_{r,k|k} + \Delta x_r sin\left(\phi_{r,k|k}\right) + \Delta y_r cos\left(\phi_{r,k|k}\right)\\
\phi_{r,k+1|k} &= \phi_{r,k|k} + \Delta\phi_r
\end{aligned}$$

A. Simulation Experiments

It is well known that one of the main limitations of the EKF solution to the SLAM problem is the inconsistency of the algorithm due to linearization errors [24], that can lead to catastrophic failure when the true uncertainty of the robot's orientation exceeds a limit [16]. The source and factors of inconsistency when landmarks convey angular information have also been thoroughly studied, as it is the case when indoor walls are modeled as segments [25].

In this section, several experiments are performed showing the limitations of BS-SLAM from the point of view of the filter consistency.

1) Consistency of BS-SLAM: For consistency experiments, three synthetic environments have been generated combining both straight and curved features (Fig. 9). A simulated mobile robot performs a double loop in each of these environments, and odometry and laser measurements are contaminated with gaussian noise with mean 0 and covariances detailed in Fig. 9. In all cases laser sensor is simulated with a maximum range of 8 m.

Results of these simulations are displayed in Fig. 10. In experiments E1.A, E2.A and E3.A map splines are progressively extended as new areas of the environment are explored, while in experiments E1.B, E2.B and E3.B splines are not extended, and new features are added to the stochastic map when no matching is successful. It can be clearly seen how extra simplifications introduced during the extension process produce a much earlier appearance of inconsistency, being the filter optimistic; i.e. the real location uncertainty is greater than the estimated.

2) One symptom of inconsistency: excessive information gain: As Bailey *et al.* pointed out in [16], one symptom of inconsistency is the excessive information gain, which makes any EKF-SLAM implementation to become optimistic after a certain period of time (the estimated covariance is less than the true covariance). To analyze this effect in BS-SLAM, a static robot has been placed in the curved environment defined by (12) that can be seen in Figs. 2 and 3. The observation noise is $\sigma_L = 5$mm for the laser distance readings. Having a static vehicle making successive observations of this environment, its true uncertainty may never decrease.

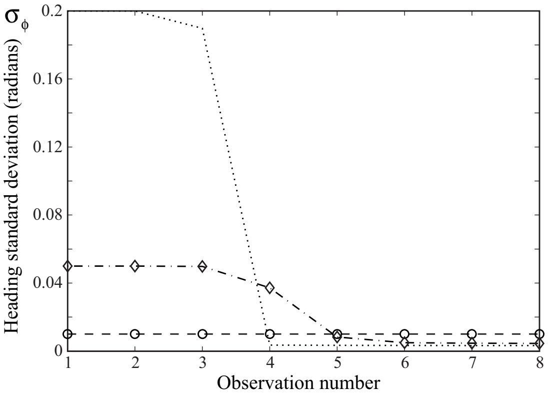


Fig. 11. Information gain in heading for a stationary vehicle. The estimated standard deviation σ_ϕ decreases after several updates. The results show the average values for 50 Monte Carlo simulations, for three different inicial values.

Fig. 11 shows the results of this experiment, using different initial values for the heading typical deviation of the robot ($\sigma_\phi(0|0) = 0.01$rad., $\sigma_\phi(0|0) = 0.05$rad., and $\sigma_\phi(0|0) = 0.2$rad.). The graphic shows the mean result for 50 runs of a Monte Carlo simulation for each initial value. For each case, the heading standard deviation shows an immediate abrupt decrease, and the larger initial uncertainties tend to produce the smallest final estimations.

3) Motecarlo tests of filter consitency: The experimental results in Fig. 10 suggest that the splines extension has some impact on the consistency properties of the algorithm. This section analyzes this effect using an environment of the shape depicted in Fig. 9.b, for $a = 14$ m, $b = 10$ m, $r1 = 4$ m, and $r2 = 2$ m. The process noise standard deviations are: $\sigma_x = 5 + 0.1\Delta x$ mm, $\sigma_y = 5 + 0.1\Delta y$ mm, and $\sigma_\phi = 0.02 + 0.1\Delta\phi$ rad., and the typical deviation of a laser range finder with a forward $180°$ field of view and maximum range of 8 meters is $\sigma_L = 5$ mm.

When the true state of the vehicle $\mathbf{x}_r(k)$ is known, the normalized estimation error squared (NEES) can be used to characterize the filter performance [16]:

$$\epsilon(k) = \left(\mathbf{x}_r(k) - \hat{\mathbf{x}}_r(k|k)\right)^T \mathbf{P}_r^{-1}(k|k)\left(\mathbf{x}_r(k) - \hat{\mathbf{x}}_r(k|k)\right) \tag{56}$$

The average NEES over N Monte Carlo simulations, under the hypotheses of a consistent and approximately linear-Gaussian filter, is a χ^2 distribution with $dim\left(\mathbf{x}_r(k)\right)$ degrees of freedom. Hence, the average value of $\epsilon(k)$ tends towards the dimension of the state (3, considering the robot pose). Fig. 12 shows the average results for these experiments after 50 Monte Carlo runs. The 95% probability concentration region is bounded by the interval [2.36, 3.72].

Fig. 12.a shows the results when splines are not extended. Fig. 12.a shows the results when splines are not extended and an inflated sensor covariance is used, multiplying by a factor of 3 matrix $\mathbf{R}$ in (34), and Fig. 12.c shows the results when splines are extended and the true matrix $\mathbf{R}$ is used. In all the cases the filter become optimistic when 50 samples

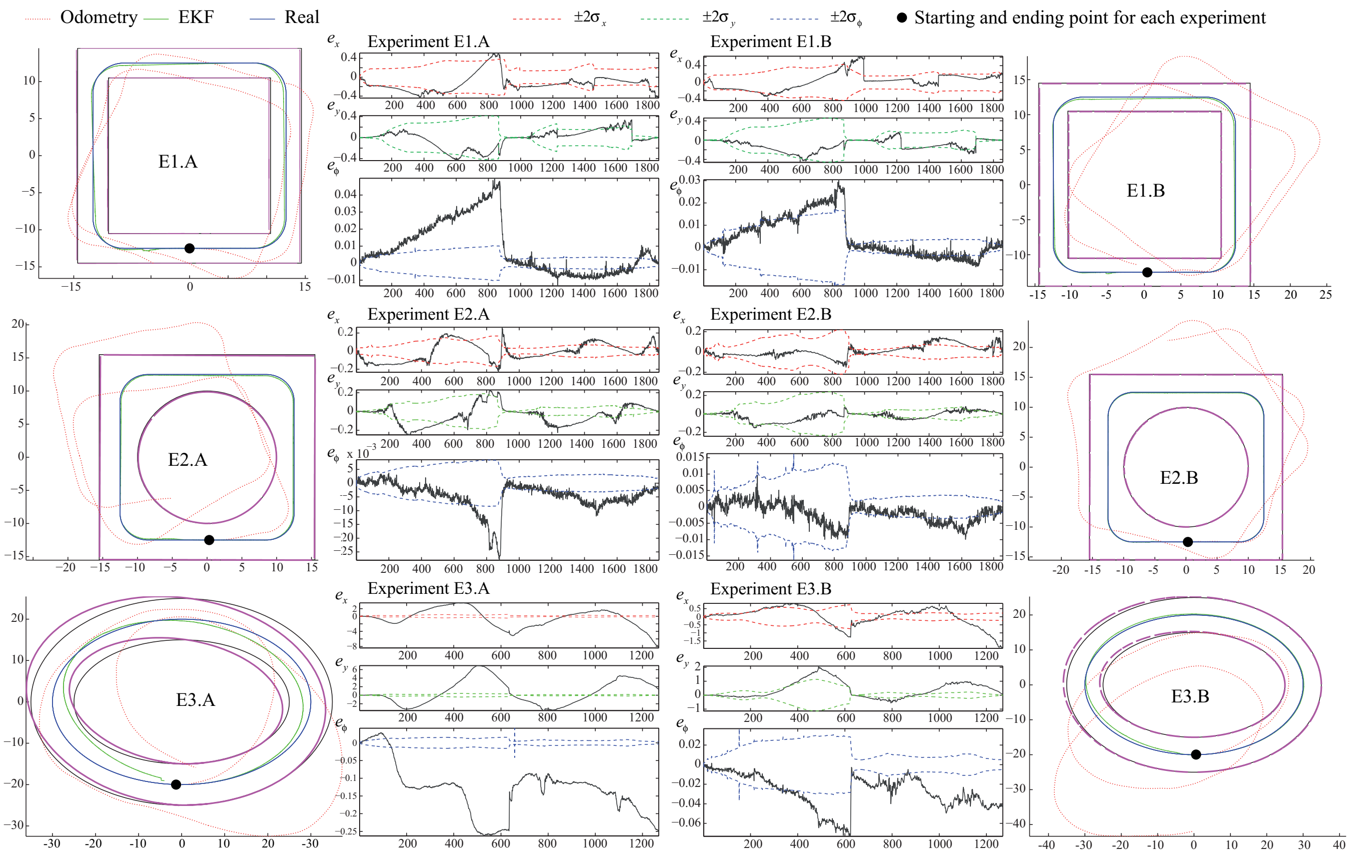


Fig. 10. Consistency experiments results. Left column shows the simulated environments (black line) and the obtained maps (magenta line), along with the odometry, real and filtered robot trajectories. For each experiment, the central graphics show consistency results for robot position and orientation with $\pm 2\sigma$ bounding limits. All linear dimensions are in meters, and angular dimensions in radians.

have been processed, and the fact of raising the covariance of the observation model does not ameliorate the result. The appearance of inconsistency is even more abrupt when splines are extended.

B. Experiments With Real Data

Several experiments have been carried out with data obtained from real environments. Fig. 13 shows a map representing an environment with predominantly flat features (segments), which are modeled using linear splines (order $\kappa = 2$). The map contains 83 linear splines (166 control points). An alternative map, using cubic splines can be found in [15] (in that case 81 splines defined by 332 control points were necessary to describe the environment).

Fig. 14 depicts the map of a bigger and more complex environment with a mixture of both straight and curved features. In this case 461 control points defining a total of 96 cubic splines are necessary. In experiments of Fig. 13 and Fig. 14, a B21r robot with a SICK laser was used for the data acquisition with a sampling frequency of 5 Hz. A knot spacing of $\rho = 2$ m. was used for knot vectors generation. Please, note that no geometric constraints have been used in the map building process.

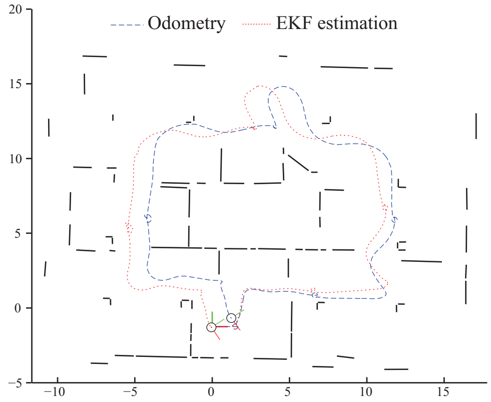


Fig. 13. Map of a fair held at the School of Industrial Engineering of the Universidad Politécnica de Madrid.

When building large maps, computational cost (quadratic with the size of the environment) arises as an unavoidable problem. With the aim of demonstrating the accuracy of the proposed methodology, which allows the exploitation of the majority of the rich information provided by a laser range finder, the map in Fig. 15 is provided. This map shows the interior of the Intel Research Lab in Seattle. The data set was obtained from the Robotics Data Set Repository [26] (thanks go to Dieter Fox for providing this data). It is built using only the localization method; i.e. the only feature in the stochastic map is the robot, and features are added to the map with zero uncertainty. The low cumulated localization error allows the good results obtained. An alternative representation using

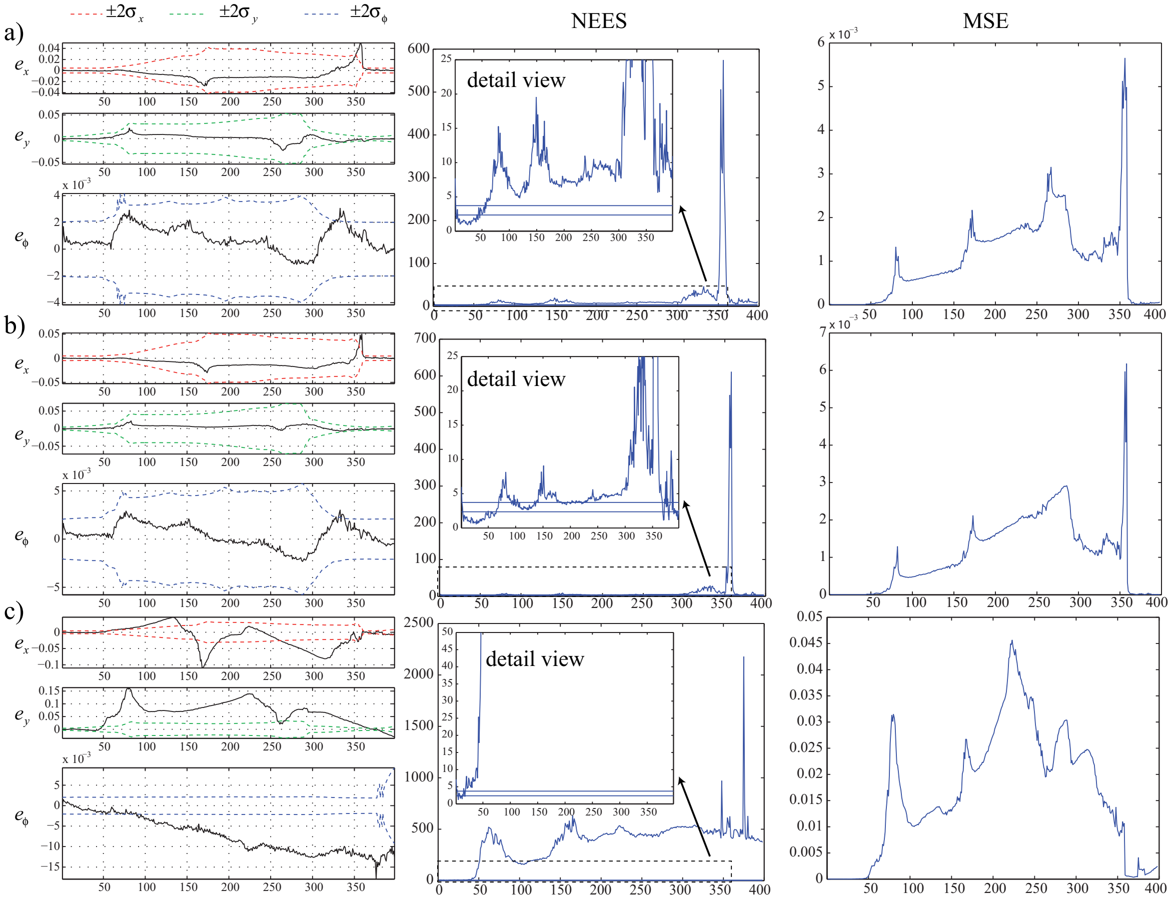


Fig. 12. Monte Carlo experiments for consistency (50 runs). The first column displays the mean localization error vs. $2\sigma_x$, $2\sigma_y$ and $2\sigma_\phi$ (linear dimensions are in meters and angular dimensions in radians). The second column is the NEES and the third column the Mean Squared Error. a) No extensions are performed. b) No extensions are performed and sensor noise is inflated by a factor of 3 in the filter. d) Extensions are performed.

occupancy grid maps can be found in [7].

We want to call the attention of the reader to the detail view in Fig. 15. The fact of using a fixed degree for the spline curves, reduces the performance of the method increasing the strictly necessary density of control points for describing simple geometries as segments. The minimum number of control points for a spline of degree κ is precisely κ. We are using in this map only cubic splines, but it is clear that for the correct description of flat features a lower degree ($\kappa = 2$) should suffice, as occurs in the map in Fig. 10. Videos of the experiments can be downloaded from [27].

V. Conclusion

A new methodology for simultaneous localization and mapping in complex environments with a mixture of flat and curved geometries has been described and experimentally tested. The power and computational efficiency of spline curves, has been used in an EKF-SLAM framework allowing the representation of complex structures in a parametric way. BS-SLAM provides a set of simple and easily programmable matrix-form expressions, allowing a successful symbiosis between EKF-SLAM and B-splines theory. When simple descriptions of the environment are insufficient or unfeasible, any other SLAM algorithm could benefit from this new representation.

Control points contained in the stochastic map incrementally encapsulate all the uncertainties conveyed in the odometry and laser sensor readings. Finally, this new representation constitutes a big step forward compared to current SLAM techniques based on geometric maps: where no other feature based algorithm could the mathematical interpretation and reasoning over the map features, regardless of their shape, is now possible once they are described as parametric functions.

Our current research involves further extension of the ideas presented in this paper on the following topics:

- Improvement of segmentation strategies, developing more sophisticated clustering techniques, and enhancing the detection and isolation of features, previously to the curve fitting process.
- Intelligent control points selection, making the most of spline management techniques: degree elevation, degree reduction, knot insertion and knot deletion. The aim is to use only the strictly necessary degree for correct features description, deleting unnecessary knots and introducing new ones when a shape refinement is necessary.
- Exploitation of the parametric feature representation in the data association process.
- Finally, we believe that techniques and concepts presented here could find a natural extension to a 3D

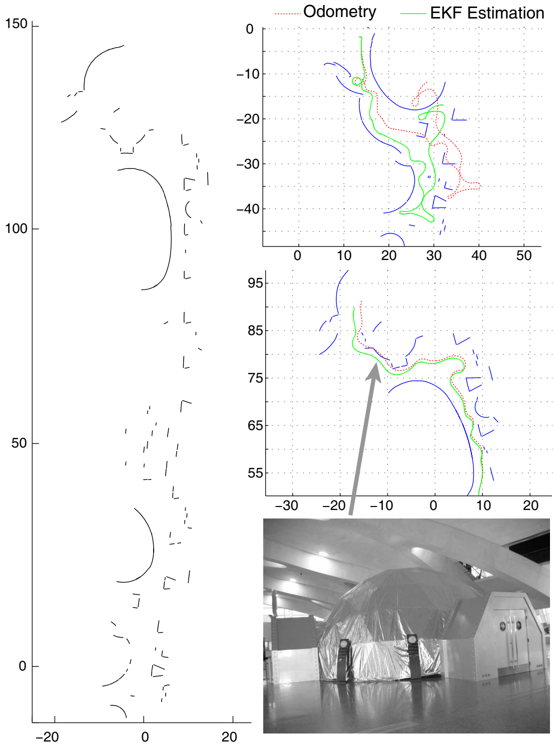


Fig. 14. Left: Map of the Museum of Science "Príncipe Felipe" (Valencia, Spain). Right: Two detail views and picture of the museum.

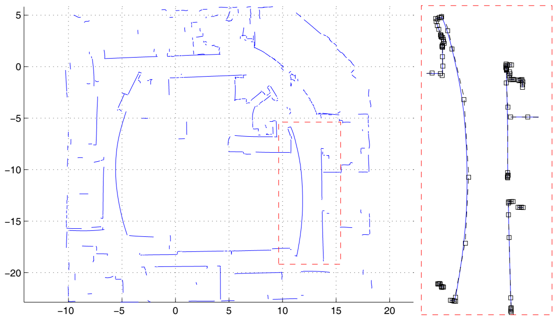


Fig. 15. Intel Laboratories map and detail view showing control points.

scenario, given the similiarities in the formulations for spline curves and spline surfaces.

Acknowledgement

The first author thanks to the "Consejería de Educación de la Comunidad de Madrid" for the PhD scholarship being granted to him, and to the "ARC Centre of Excellence for Autonomous Systems", UTS node, for having him as visiting postgraduate research student during some months in 2006.

References

[1] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, March 1985, pp. 116–121.

[2] 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 Trans. Robot. Autom.*, vol. 17, no. 3, pp. 229–241, 2001.

[3] J. Guivant, E. Nebot, and H. Durrant-Whyte, "Simultaneous localization and map building using natural features in outdoor environments," in *Proc. Intelligent Autonomous Syst. 6*, vol. 1, Venice, Italy, July 2000, pp. 581–588.

[4] J. J. Leonard, H. F. Durrant-Whyte, and I. J. Cox, "Dynamic map building for an autonomous mobile robot," *Int. J. Rob. Res.*, vol. 11, no. 4, pp. 286–298, August 1992.

[5] R. Madhavan, G. Dissanayake, and H. F. Durrant-Whyte, "Autonomous underground navigation of an LHD using a combined ICP–EKF approach." in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 4, 1998, pp. 3703–3708.

[6] R. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1100–1114, December 2006.

[7] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, Las Vegas, NV, USA, October 2003, pp. 206–211.

[8] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based SLAM with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 2443–2448.

[9] J. A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardós, "The SPmap: A probabilistic framework for simultaneous localization and map building," *IEEE Trans. Robot. Autom.*, vol. 15, no. 5, pp. 948–952, October 1999.

[10] D. Rodriguez-Losada, F. Matía, and R. Galán, "Building geometric feature based maps for indoor service robots," *Robotics and Autonomous Systems*, vol. 54, no. 7, pp. 546–558, July 2006.

[11] P. MacKenzie and G. Dudek, "Precise positioning using model-based maps," in *Proc. IEEE Int. Conf. Robot. Autom*, 1994, pp. 1615–1621.

[12] M. Veeck and W. Burgard, "Learning polyline maps from range scan data acquired with mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 2, Sendai, Japan, 2004, pp. 1065–1070.

[13] S. Zhang, L. Xie, M. Adams, and F. Tang, "Geometrical feature extraction using 2D range scanner," in *Proc. Fourth Int. Conf. Contr. Automat.*, Montreal, Canada, June 2003, pp. 901–905.

[14] J. Nieto, T. Bailey, and E. Nebot, "Scan-SLAM: Combining EKF-SLAM and scan correlation," in *Proc. Int. Conf. on Field and Service Robotics*, 2005, pp. 129–140.

[15] L. Pedraza, G. Dissanayake, J. Valls Miró, D. Rodriguez-Losada, and F. Matía, "BS-SLAM: Shaping the world," in *Proc. Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[16] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[17] D. F. Rogers, *An introduction to NURBS: With historical perspective*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.

[18] C. de Boor, *A Practical Guide to Splines*, J. Marsden and L. Sirovich, Eds. Springer, 1978.

[19] L. Piegl and W. Tiller, *The NURBS Book (2nd ed.)*. New York, USA: Springer-Verlag New York, Inc., 1997.

[20] J. H. Ahlberg, E. N. Nilson, and J. L. Walsh, *The Theory of Splines and their Applications*. New York, USA: Academic Press, 1967.

[21] A. Atieg and G. A. Watson, "A class of methods for fitting a curve or surface to data by minimizing the sum of squares of orthogonal distances," *J. Comput. Appl. Math.*, vol. 158, no. 2, pp. 277–296, 2003.

[22] A. J. Sweeney and R. H. Barrels, "Ray tracing free-form B-spline surfaces," *IEEE Comput. Graph. Appl.*, vol. 6, no. 2, pp. 41–49, February 1986.

[23] S.-M. Hu, C.-L. Tai, and S.-H. Zhang, "An extension algorithm for B-Splines by curve unclamping," *Computer-Aided Design*, vol. 34, no. 5, pp. 415–419, April 2002.

[24] J. A. Castellanos, J. Neira, and J. D. Tardós, "Limits to the consistency of EKF-based SLAM," in *5th IFAC Symposium on Intelligent Autonomous Vehicles, IAV'04*, Lisbon, Portugal, July 2004.

[25] D. Rodriguez-Losada, F. Matía, L. Pedraza, A. Jiménez, and R. Galán, "Consistency of SLAM-EKF algorithms for indoor environments," *Journal of Intelligent and Robotic Systems*, vol. 50, no. 4, pp. 375–397, December 2007.

[26] A. Howard and N. Roy. (2003) The robotics data set repository (Radish). [Online]. Available: http://radish.sourceforge.net

[27] L. Pedraza. Experiments of SLAM with B-splines. [Online]. Available: http://www.luispedraza.es/research.php