Mobile Robot SPLAM for Robust Navigation

Abraham Sánchez[1], Alfredo Toriz[2], Rene Zapata[2], and Maria Osorio[1]

[1] Benemérita Universidad Autónoma de Puebla, Computer Science Department, Puebla, Mexico
[2] Montpellier Laboratory of Informatics, Robotics, and Microelectronics (LIRMM), UMR 5506 - CC 477
161 rue Ada, Montpellier, France
alfredo.torizpalacios@lirmm.fr, zapata@lirmm.fr, asanchez, aosorio@cs.buap.mx

Abstract. This paper describes a simultaneous planning localization and mapping (SPLAM) methodology, where a mobile robot explores the environment efficiently and also considers the requisites of the simultaneous localization and mapping algorithm. The method is based on the randomized incremental generation of a data structure called Sensor-based Random Tree, which represents a roadmap of the explored area with an associated safe region. A continuous localization procedure based on B-Splines features of the safe region is integrated in the scheme.

1 Introduction

SLAM (Simultaneous Localization And Mapping) is a challenging problem in mobile robotics. SLAM approaches are used simultaneously with classic exploration algorithms [1]; however, results obtained with SLAM algorithms strongly depend on the trajectories performed by the robots, while classic exploration algorithms do not take into account the uncertainty about the localization of the robot when it travels through unknown environments, making harder the construction of the map, when the robot's position is unknown, generating useless and inaccurate maps. With the integrated exploration or SPLAM (simultaneous planning localization and mapping), the robot explores the environment efficiently and also considers the requisites of the SLAM algorithm [2], [3].

An integrated exploration method is introduced in [3] to achieve the balance of speed of exploration and accuracy of the map using a single robot [3]. Freda et al. [2] use a sensor-based random tree (SRT). Recently, a novel laser data based SLAM algorithm using B-Spline as features has been developed in [4]. Extended Kalman filter (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 observation model used for the EKF update is the intersections of the laser beams with the splines contained in the map. In our proposal, we did not use the EKF but and integrated exploration based approach, called SRT-B-Splines. In this method, the tree is expanded, while the configurations for the coverage near the frontiers of the robot, that is a new candidate, are selected.

These configurations belonging to the new candidates are evaluated considering the reliability of the expected observable features in those points.

The basics of the B-splines are briefly presented in Section II. The proposed approach to solve the simultaneous planning localization and mapping problem is detailed in Section III. Simulation results are discussed in Section IV. Finally, conclusion and future work are detailed in Section V.

2 B-Splines

Most shapes are simply too complicated to define using a single Bézier curve. A spline curve is a sequence of curve segments that are connected together to form a single continuous curve. A knot vector is a list of parameter values, or knots, that specify the parameter intervals for the individual Bézier curves that make up a B-spline. The purpose of the knot vector is to describe the range of influence for each of the control points [6]. Let a list $t_0 \leq t_1 \leq t_2 \leq \ldots \leq t_{m-1} \leq t_m$ of $m+1$ non-decreasing numbers, such that the same value should not appear more than k times, $k =$ order of the B-spline. We define the i-th B-spline function $N_{ik}(t)$ of order $k(= k-1$ degree) as:

$$N_{i1}(t) = \begin{cases} 1 & \text{if } t_i \leq t \leq t_{i+1}, \\ 0 & \text{otherwise.} \end{cases}, \ k = 1 \tag{1}$$

$$N_{ik}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t), \ k > 1 \tag{2}$$

Let the next properties:

1. $N_{ik}(t) > 0$ for $t_i < t < t_{i+k}$
2. $N_{ik}(t) = 0$ for $t_0 \leq t \leq t_i, t_{i+k} \leq t \leq t_{n+k}$
3. $\sum_{i=0}^{n} N_{ik}(t) = 1 \quad t \in [t_{k-1}, t_{n+1}]$ normalizing property

Given a set of $n+1$ control points $d_i (i = 0, ..., n)$ and a knot vector $T = [t_0, t_1, ..., t_{m-1}, t_m]$ one can define a B-spline $X(t)$ of order k as:

$$X(t) = \sum_{i=0}^{n} d_i N_{ik}(t) \tag{3}$$

where $N_{ik}(t)$ describes the blending B-spline function of degree $k-1$ associated with the knot vector T.

The simplest method of fitting a set of data points with a B-splines curve is the global interpolation method [6]. The spline fitting problem is, given a set of data points $D_0, D_1, \ldots, D_n$ which correspond to an unknown curve, find the B-spline function to approximate the data points.

3 The SPLAM Approach

Several techniques have been proposed so far to tackle the SLAM problem. The main difference between them concerns basically with the environment representation and the uncertainty description [5]. A wide variety of localization and mapping techniques relies on environment representations consisting of a set of characteristics elements detectable by the robot's sensory system (feature-based maps). Lines and segments are commonly used as features. They can be effectively extracted from range scans and then exploited for localization and/or mapping purposes.

In the integrated exploration approach, the robot simultaneously creates a map of its environment and finds a location in such environment (i.e., the robot takes local decisions on how to move in order to minimize the error of its estimated positions and the positions of the marks). The strategy adopted for the exploration process is called SRT (Sensor Based Random Tree) [5], [8], and is based on the construction of a data structure that represents the roadmap of the explored area with an associated security region (SR); each node tree ($\mathcal{T}$) consists of a robot's position and its associated local security region (LSR) that is constructed through the perception of the robot system. It carries out a continuous localization process based on the extraction of environmental characteristics (curves or lines), these features are compared with the new curves that are extracted from the LSR of the current position. The algorithm implemented for the integrated exploration is described in [9].

The exploration of unknown environments requires an additional functionality because the odometric information reported by the robot, in most cases is not accurate, resulting in inaccurate maps useless for future navigations. The localization function implemented, uses B-spline curves to represent the frontier between the free regions and the obstacles in a complex environment. The proposed algorithm assumes that the robot's initial position is well located and, consequently, the first observation of the environment has a perfect location. Once the robot has moved from a position q_{last} to a position q_{curr}, the new position of the robot is obtained by adding to the last located position, the increment $\triangle x, \triangle y$ and $\triangle \theta$ reported by the robot's odometric system. After this position is estimated, the robot will collect the information of the surrounding environment for the localization process.

The raw data collected by the sensor can not be directly used in the localization process, due to errors inherent to the measurement system used (laser sensor). In our case, we got an equivalent error of $\pm 1\%$ of the measured distance, that will be optimized using the method of least median square (LMS Least Median Square). The decision to use this method is based on a comparative study of different methods proposed in the literature, including RANSAC and its variants (MSAC and NAPSAC), and Least Squares. The minimum is defined as $\min M = med(r_i^2)$, where $r_i (i = 1, ..., n)$ are the residuals of the control points d_i and their corresponding points on the curve: $r_i = |d_i - d_i N_{i,p}(t_k)|$. The treatment of the laser readings will be described in the data segmentation process. Before the processed data can be used by the localization algorithm, they need to undergo several processes, see Fig. 1:

– FIRST SEGMENTATION. An analysis of the relative position of consecutive data points is performed. The aim is to detect points close enough that belong to the same obstacle.
– SECOND SEGMENTATION. The obtained segments in the first segmentation are again subjected to testing for consecutive points whose angle is below a certain threshold. The objective of this segmentation is to detect corners and curves with high curvatures.
– SETTING. Each of the obstacles of the second segment are adjusted to the B-Spline grade 3 that form its control polygons.

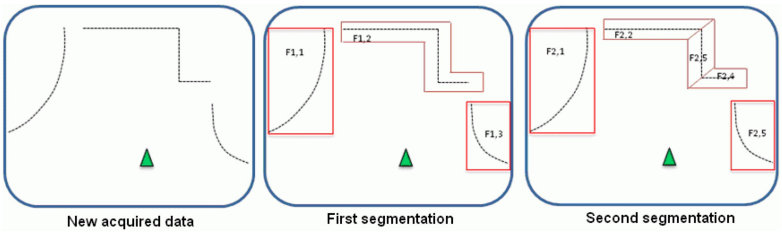

Fig. 1. The segmentation process.

The first segmentation is performed using the adaptive clustering concept, which consists of dividing a dataset into subsets (clusters) such that the data from the same subset share some common characteristic. In adaptive clustering, the membership to a measured point in the subset depends on the distance between the objects and the laser and on the calculation of different values of the discriminant $D_{threshold}$. The clustering process used in this part is based on the classic criteria of Dietmayer [10], whose operation can be explained in the Fig. 2. Pa and Pb represent two consecutive points detected by the laser, while ra and rb are the distances of these points to the coordinates origin. Given the triangle $OPaPb$, where ra and rb are known and α is the angular resolution of the laser, we can apply the cosines theorem to calculate the distance between Pa and Pb:

$$rab = \sqrt{ra^2 + rb^2 - 2rarb\cos(\alpha)}$$

Because the scanner used in our experiments have an angular resolution $\alpha = 0.061$, a very small value according to Dietmayer, it is possible to simplify the calculation of rab assuming that $rab \approx |ra - rb|$. The criteria used to form the clusters is that, if the distance between Pa and Pb is less than $rab \leq C_0 + C_1 \cdot \min\{ra - rb\}$, where $C_1 = \sqrt{2(1 - \cos(\alpha))}$, then Pb belongs to the same cluster than Pa. Otherwise, the points Pa and Pb belong to different clusters. The constant C_0 represents a noise adjustment in the laser measures.

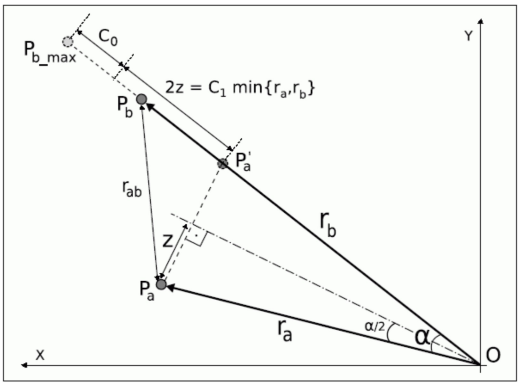

Fig. 2. Illustration of the Dietmayer's clustering criterion.

The other constant, C_1, takes a value not explained by Dietmayer, but, that can be explained using the Fig. 2, where it can be appreciated that $\min\{ra, rb\} = ra$, therefore:

$$C_1 \cdot \min\{ra, rb\} = ra \cdot \sqrt{2(1 - \cos(\alpha))} = 2 \cdot ra \cdot \sqrt{\frac{1 - \cos(\alpha)}{2}}$$

On the other side, the variable named z in the Fig. 2 will take the value:

$$z = ra \cdot \sin(\frac{\alpha}{2}) = ra \cdot \sqrt{\frac{1 - \cos(\alpha)}{2}}$$

Finally, we get $2z = C_1 \cdot ra$; which means that C_1 or ra is the distance between the points Pa and Pa', being Pa' a point on the segment OB located at the same distance from the origin than Pa. On the other hand, knowing that $rb > ra$ is satisfied in the Fig. 2, then:

$$rab = rb - ra \leq C_0 + C_1 \cdot ra$$
$$rb \leq ra + C_0 + C_1 \cdot ra$$

This means that if we add the distances $C1 \cdot ra$ and C_0 to the point Pa', we obtain the point Pb_{max} that corresponds to the maximum distance that point Pb can be moved in order to form part of the same cluster that the point Pa. As mentioned above, the second segmentation has the objective of detecting straight lines that form corners and high curvature loops. To achieve this objective we used the work by Pavlidis and Horowitz named "Split and Merge" [11]. The algorithm has two phases. The first phase is recursive, and consists in dividing the available segments into smaller ones, while the second is used to merge segments that are almost collinear. In Fig. 3, one can see a practical example of the algorithm.

Due to the nature of the method, we can make particular segments of untreated data corresponding to a specific cluster, that is, the method allows us to

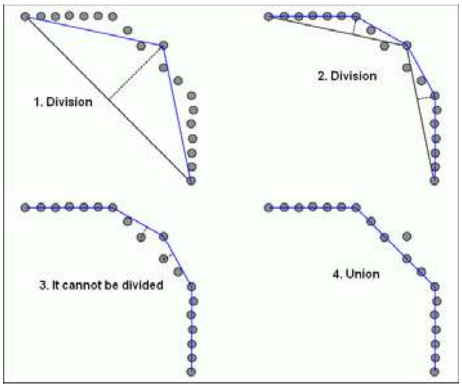

Fig. 3. The split and merge algorithm.

easily identify straight lines but at the same time if the result is a series λ_{min} of non-collinear segments smaller than D_{min} threshold, then we can safely say that the robot is approaching a curve. Thus, the treatment to be given to the data is as follows:

- **Segments of straight lines**. Let $\min M = med(r_i^2)$, where $r_i(i = 1, ..., n$ are the residual of the points to the line $r_i = |y_i - mx_i - b|$. To calculate M, one can use the algorithm proposed by David M. Mount et al., [12].
- **Curves**. Let $\min M = med(r_i^2)$, where $r_i(i = 1, ..., n$ are the residual of the points to the line $r_i = |d_i - d_i N_{i,p}(t_k)|$. This operation is performed by approximating a cubic B-Spline to the points d_i. Once obtained the residuals, there will be a correction, moving the points d_i with larger residuals, a distance heuristically selected, to the B-Spline.

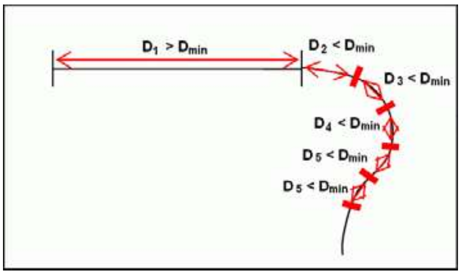

Fig. 4. Segmentation of lines and curves on a cluster by using the split and merge algorithm. D_1 is considered a straight line and the segments D_2, ..., D_5 curves.

Once the data from the sensor are segmented, a process of data association is performed. The first association is crude, and the control points of each segment obtained in the segmentation process are compared with the control points in the map, using the following criteria:

$$\min(dist(X_{m,i}, X_{o,j})) < d_{min}, i = 1, ..., n_m, j = 1, ..., n_0$$

Where $X_{m,i}$ and $X_{o,j}$ are the control points of the splines, on the map and observed, respectively, n_m and n_0 are the number of control points of the splines on the map and observed, $dist(X_{m,i}, X_{o,j})$ represents the Euclidean distance between the control points, and finally d_{min} is the parameter that will regulate if the points are or not related. If no spline in the map is close enough to a detected spline in order to be related, then this new object is added to the map, once the robot's position has been located. By contrast, if a spline is associated with a map's feature, it is necessary to obtain a concordance between its points, as follows:

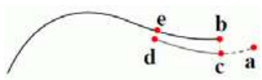

Fig. 5. Concordance between curves.

- One of the ends of the curve is considered point a.
- The closest point between the spline on the map and the point a is calculated (point b).
- If b is one of the endpoints of the spline on the map, then, the point nearest to b in the spline is calculated and named point c, if not, point a is associated with point b.
- The process is repeated starting in the other end of the spline (point d in the Fig. 5, that is associated with the point e on the spline in the map.
- Due to the B-splines property, the length of the curves can be known, and segments e-b and d-c can be adjusted to have the same length. If the difference of the lengths is greater than threshold l_{max}, the extreme elements of the larger curve are eliminated to adjust its size.

Once the curves of the estimated position and the curves of the environment are associated, it is necessary to conduct a final verification of the association by getting the distance from each end of each curve to the other as shown in the Fig. 6:

4 Experimental Results

A simulated robot and the real Pioneer P3DX robot equipped with front and rear bumper arrays, a ring of eight forward ultrasonic transducer sensors (range-finding sonar) and a Hokuyo URG-04Lx laser range finder were used in the experiment. The Pioneer P3DX robot is an unicycle robot. The LIRMM laboratory

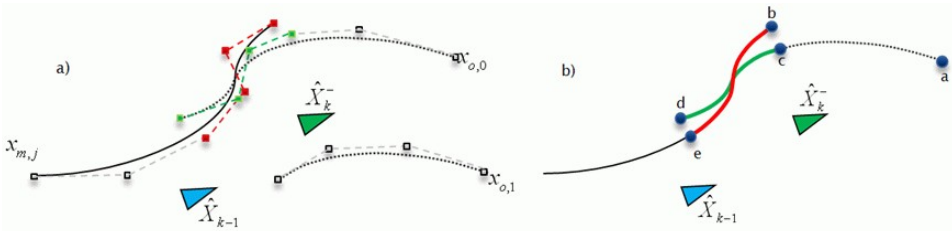

Fig. 6. Two association forms: a) association crude, b) association fine.

environment was used in the experimental and simulation tests (the environment had several corridors). Figure 7 shows the two final maps: in the leftside, the map obtained without a localization process and only with odometric estimates; in the rightside, the map obtained with the proposed approach. Comparing the two final maps, it can be said that the robot did not use the localization process, and collided frequently with the obstacles, as can be appreciated in the left image. Figure 8 shows the odometry errors versus the errors obtained with the proposed approach.

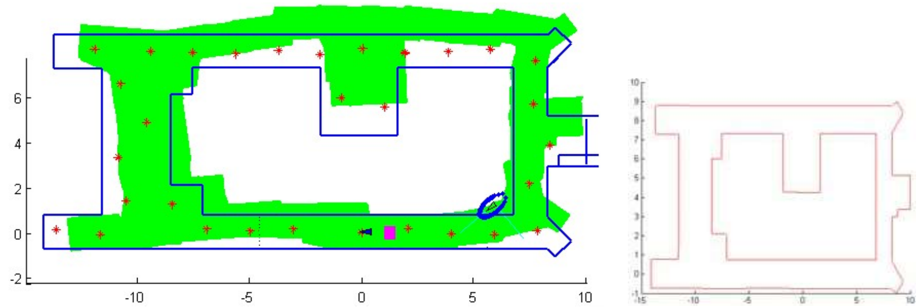

Fig. 7. Final map obtained only with odometric estimates and final map obtained with the SPLAM approach.

For the association process, the basic procedure proposed in [4] was retaken, but with a new functionality derived from the properties of the B-splines, i.e., we ensure that these new curves and those belonging to the environment have the same lengths. This new feature enabled better and more accurate association of the data collected with the sensor than the association obtained with the basic method originally proposed. The geometric properties of the final regions of the curves was considered to make a final association checking with the distances of the points a the end of the curves of the environment and the estimated position in order to verify their similitude. This exhaustive verification is necessary because the nature of the proposed localization method. It can be said that the

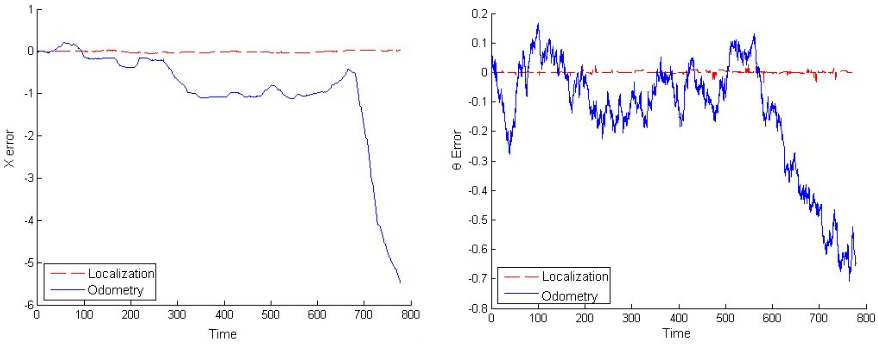

Fig. 8. X, θ errors for the odometry and the proposed approach.

approach presented in this paper, makes a good use of the parametric representation of the environment characteristics at the time of the data association.

The localization capability of a mobile robot is central to basic navigation and map building tasks. The two main instances of mobile robot localization problem are the continuous pose maintenance problem and the global localization also known as 'robot kidnapping' problem. Global position estimation is the ability to determine the robot's position in an a priori or previously learned map, given no information other than that the robot is somewhere in the region represented by the map. Fulfilling all these properties, our method can solve the kidnapping problem in a robust form, as can be seen in Fig. 9.

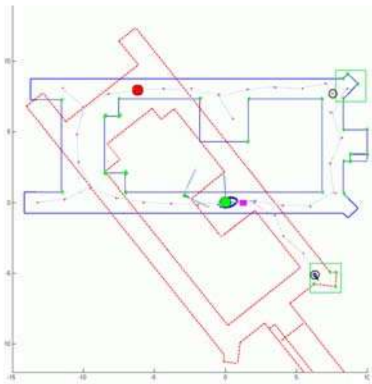

Fig. 9. Snapshot showing the execution of the proposed kidnapping strategy.

5 Conclusions and Future Work

Most of the early SLAM work was point-feature based. The main drawback with point-feature based SLAM is that measurements acquired from typical sensors did not correspond to the feature points 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. Furthermore, in some situation, the environment does not have enough significant structure to enable point features to be robustly extracted from them.

As a conclusion, we can mention that we have developed a robust SPLAM tool that is not limited to environments with linear features. The localization method is perfectly suited to the new curves that can be increasingly seen in everyday's life. The theory and implementation of the B-splines was a powerful tool in our approach, and can be adapted to environments where the previous methods considered only simple descriptions.

As future work, we have considered the challenge of working with an extension of our proposal to the case of integrated exploration with multiple robots, which will take to us to the search of a solution to the multi-robot localization problem.

References

1. S. Thrun, W. Burgard and D. Fox, "Probabilistic robotics", *The MIT Press*, (2005)
2. L. Freda, F. Loiudice and G. Oriolo., "A randomized method for integrated exploration", *IEEE Int. Conf. on Intelligent Robots and Systems*, (2006) 2457–2464
3. A. A. Makarenko, S. B. Williams, F. Bourgante and H. F. Durrant-Whyte., "An experiment in integrated exploration", *IEEE Int. Conf. on Intelligent Robots and Systems*, (2002) 534–539
4. 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*, Vol. 25, (2009) 353–366
5. A. Garulli, A. Giannitrapani, A. Rossi and A. Vicino, "Mobile robot SLAM for line-based environment representation", *IEEE European Control Conference. CDC-ECC*, (2005) 2041–2046
6. F. Yamaguchi, "Curves and surfaces in computer aided geometric design", *Springer-Verlag*, (1988)
7. G. Oriolo, M. Vendittelli, L. Freda and G. Troso, "The SRT method: Randomized strategies for exploration", *IEEE Int. Conf. on Robotics and Automation*, (2004) 4688–4694
8. J. Espinoza L., A. Sánchez L. and M. Osorio L., "Exploring unknown environments with mobile robots using SRT_Radial", *IEEE Int. Conf. on Intelligent Robots and Systems*, (2007) 2089–209
9. A. Toriz P., A. Sánchez L., R. Zapata and M. Osorio L., "Building feature-based maps with B-splines for integrated exploration". *LNAI 6433*, (2010) 562–571
10. K. C. Dietmayer, J. Sparbert and D. Streller, "Model based object classification and object tracking in traffic scenes from range images", *Proc. of the IV IEEE Intelligent Vehicles Symposium*, (2001) 25–30

11. T. Pavlidis and S. L. Horowitz, "Segmentation of plane curves", *IEEE Transactions on Computers*, Vol. C-23, No. 8, (1974) 860–870
12. D. M. Mount, N. S. Netanyahu, K. Romanik, R. Silverman and A. Y. Wu, "A practical approximation algorithm for the LMS line estimator", *Journal Computational Statistics & Data Analysis*, Vol. 51, Issue 5, (2007)