

Building Maps of Large Environments Using Splines and Geometric Analysis

Luis Pedraza, Diego Rodriguez-Losada, Pablo San Segundo, and Fernando Matía

Abstract—Recently, a novel solution to the Simultaneous Localization and Map building (SLAM) problem for complex indoor environments was presented, using a set of splines for describing the geometries detected by a laser range finder mounted on a mobile platform. In this paper, a method for exploiting the geometric information underlying in these maps in the data association process is described. The proposed approach uses graphs of relations between simple features extracted from the environment, and a bit encoded implementation for obtaining a maximum clique that relates observations with previously visited areas. This information is used to update the relative positions of a collage of submaps of limited size.

I. INTRODUCTION

The Simultaneous Localization and Map Building problem (SLAM) has been the focus of attention of the mobile robotics research community for more than two decades. Today, very impressive implementations of a wide range of solutions to the problem can be found in the literature. Some of the most successful approaches are based on the use of an Extended Kalman Filter (EKF) for the concurrent estimation of the robot pose and the features locations in the environment.

The main disadvantage of EKF-SLAM implementations, along with the unavoidable appearance of inconsistency [1], is its computational cost, quadratic with the number of features in the map. Much work has been done to overcome this problem [2], [3], while the most straightforward solution seems to be the decomposition of the whole map into different submaps of limited size [4], [5].

However, the search of a correspondence between the different maps built during the modeling of a large scale environment is yet a poorly solved problem. Mechanisms for establishing correspondences between landmarks contained in different submaps that correspond to the same environment features are necessary for the obtaining of a consistent global map that accurately represents the reality as a whole. Considering every map as a set of landmarks positioned in a different frame for each of them, the goal is to find the correct set of associations between a set of observations and a set of reference landmarks. The cost of this search is exponential in time with the number of observations and landmarks considered. A suboptimal solution that deals with this computational cost can be obtained using an approximate representation of the environment and a voting algorithm [6].

This work was supported in part by the Spanish Ministry of Science under Project Robonauta: DPI2007-66846-C02-01

The authors are with the Intelligent Control Group, Universidad Politécnica de Madrid, C/ José Gutiérrez Abascal 2, E-28006 Madrid, Spain
luis.pedraza@ieee.org

The search of a correspondence between a set of observations and a different set of landmarks stored in a map can be formulated as the solution of a maximum clique problem (MCP) [7]. This is known to be a NP-Hard problem [8], that found a very efficient solution in [9] using a formulation based on encoding the information in bit strings and exploiting bit-level parallelism in hardware to boost the operations up. This bit-encoded solution to the MCP (BE-MCP) was successfully implemented in [10] for solving the global localization problem using maps composed of simple point features. Results outperformed existing algorithms such as branch and bound heuristic over the associations space [6] in up to two orders of magnitude. More importantly, the solution for a maximum clique search between a set of observations and a set of landmarks is exact, not relying on simplifications, and does not require of an initial relative position estimation.

In this paper it is proposed the use of BS-SLAM [11] for building submaps using data gathered by a mobile robot equipped with a laser range finder for sensing its environment. The computational cost of building these units is bounded by limiting their size in the number of control points. In this local approach, features are modeled as cubic splines, allowing the exploitation of much more laser data than traditional feature based SLAM algorithms can handle. Relative positions of these submaps are updated using a BE-MCP algorithm for establishing correspondences amongst the constellation of submaps in the form of relative spatial relationships. This way, global maps of large environments can be generated using a collage of accurate local representations.

The paper is organized as follows. Section II presents an overview of the fundamentals of using splines as a tool for describing complex indoor environments. Section III briefly explains how to address the computational problem of an EKF-SLAM algorithm by decomposing the full environment into more manageable submaps of limited size. Sections IV and V put forward the procedures for establishing correspondences between features contained in different submaps and some experimental results with real data, exemplifying the contributions of this article. Finally, in section VI some conclusions and future research lines are summarized.

II. SLAM WITH SPLINES

In [11], a novel method for Simultaneous Localization and Mapping was presented. In that work, it was proposed the use of splines for representing complex geometries in indoor environments. Up to that moment, feature based SLAM had

been undertaken using simple geometric entities such as points [12], segments [13], [14] or polylines [15].

A. B-splines Definition

Splines are powerful mathematical entities, capable of describing complex geometries in a powerful and computationally efficient manner. Letting $\mathbf{s}(t)$ be the position vector along the curve that describes an environment feature as a function of the parameter t , a spline curve of order κ (degree $\kappa - 1$), with control points $\mathbf{x}_i = (x_i, y_i)^T \in \mathbb{R}^2, i = 0 \dots n$ and knot vector $\Xi = \{\xi_0, \dots, \xi_{n+\kappa}\}$ can be expressed as:

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

being $\beta_{i,\kappa}(t)$ the normalized B-spline basis functions of order κ [16].

The knot vector is any nondecreasing sequence of real numbers ($\xi_i \leq \xi_{i+1}$ for $i = 0, \dots, n + \kappa - 1$). They define the locations of the parameter where two polynomial pieces join together, defining piecewisely the global shape of the curve. Spline curves are a computationally efficient way to handle complex geometries, and allow the mathematical analysis of the features, computing characteristics such as distances, lengths or curvatures. For example, the derivative of a B-spline of order κ is a B-spline of order $\kappa - 1$, whose coefficients can be obtained differencing the original ones [17].

$$\frac{d\mathbf{s}(t)}{dt} = \mathbf{s}'(t) = (\kappa - 1) \sum_{i=0}^{n-1} \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{\xi_{i+\kappa} - \xi_{i+1}} \beta_{i,\kappa-1}(t) \quad (2)$$

B. State vector in BS-SLAM

BS-SLAM addresses the task of describing the environments using the control points defining a set of splines. Considering the robot as the only non-static element, the state vector to be estimated is:

$$\mathbf{x} = [\mathbf{x}_r^T, \mathbf{x}_{s_1}^T, \dots, \mathbf{x}_{s_N}^T]^T \quad (3)$$

where the pose of the robot and the N features contained in the map are modeled as:

$$\begin{aligned} \mathbf{x}_r &= [x_r, y_r, \phi_r]^T \\ \mathbf{x}_{s_i} &= [x_{i,1}, \dots, x_{i,n_i}, y_{i,1}, \dots, y_{i,n_i}]^T, i = 1, \dots, N \end{aligned} \quad (4)$$

The main strength of BS-SLAM is that it does not rely on a specific geometry to be detected from the data provided by the sensors. The goal is to represent the environment as accurately as possible, exploiting the flexibility that B-spline curves provide for describing both segments and curves.

Our previous work in [11] provides very detailed descriptions and formulations for (a) the segmentation of the raw laser scan into separate pieces of information describing different objects, (b) initializing new objects and extending current objects when new parts of the environment are discovered, (c) formulating a suitable observation equation based on a ray-tracing algorithm between every single laser beam position and the parametric curves contained in the

map, and (d) evaluation of appropriate Jacobians for easy implementation of Kalman filter equations.

One major drawback of this solution is the computational cost of the update stage of the EKF, limiting its real time implementation to maps of up to a few hundreds of control points. As mentioned before, one successful way to alleviate this problem is the simple limitation in the number of control points contained in each submap, and maintain a spatial structure relating the positions of all of them.

III. DEALING WITH THE COMPUTATIONAL COST: BREAKING THE MAP UP INTO PIECES

One of the main drawbacks of EKF based approaches to the solution to the SLAM problem is its computational cost, quadratic with the number of features (control points in our case) present in the map. As the size of the state vector is a major problem for this formulation, many solutions have recently appeared addressing the task of building maps of large environments by simply decomposing the whole problem into pieces; i.e. building maps of limited size, corresponding to different visited areas [4], [5], and trying to merge or stitch them all together following some sort of joining process [18]. An alternative is to work with a graph of different submaps, whose poses are updated and corrected forming a constellation of local representations that, altogether, conform the whole environment.

In the work here presented, it is proposed the use of different small submaps, limited in size with the number of control points. This way, whenever a map reaches a certain maximum number of control points, a new map is initialized using the last sensor observation, and a new Kalman Filter algorithm is started for building the new map. The spatial relationship [19] of the newly born submap respect its predecessor (parent) is simply the robot pose at the map transition instant.

The simple approach proposed in this paper deals with the set of submaps as the elements of a state vector to be estimated by an extended Kalman filter. Hence, the state vector for $M + 1$ submaps is composed by the absolute position of every local frame in the global reference frame (which is coincident with the first submap generated):

$$\mathbf{x} = [\mathbf{x}_0^T, \mathbf{x}_1^T, \dots, \mathbf{x}_M^T]^T \quad (5)$$

where the pose of every single submap is defined by:

$$\mathbf{x}_i = [x_i, y_i, \phi_i]^T, i = 0, \dots, M \quad (6)$$

Now the problem is to obtain accurate relations between the elements of this set of submaps, providing information to be used for updating the estimation. This information is given in the form of spatial relationships that can be obtained whenever the robot revisits and area previously described by a map already contained in the state. Hence, the goal is to obtain correspondences between common features, present in both representations. Once this correspondence is obtained, relative position can be computed as a simple pose differences minimization problem. This scenario is schematically depicted in Fig. 1.

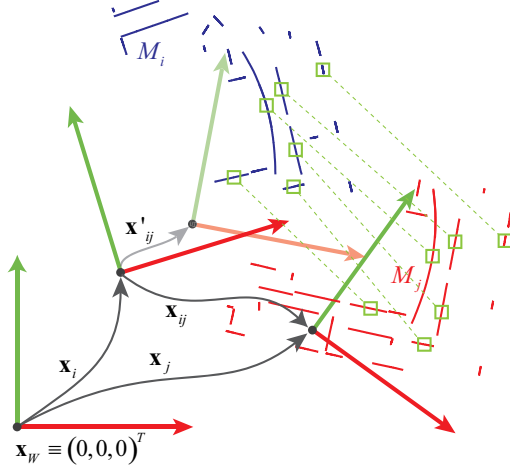


Fig. 1. Relative observation between two matched maps. Once a matching is obtained between features contained in different submaps, it is possible to obtain an observation \mathbf{x}'_{ij} which will differ from the expected relationship between both submaps \mathbf{x}_j . This information is used for updating the global map.

IV. MAP MATCHING AS A MAXIMUM CLIQUE PROBLEM

Obtaining a correspondence between two sets of features contained in two different submaps is equivalent to matching two sets of landmarks L_1 and L_2 . In this work, the procedure for obtaining the maximum correspondence between elements in both sets, is formulated as the obtaining of a maximum clique in the graph composed by the compatibility between generalized distances between every pair of features in both sets.

A. The Maximum Clique Problem

Given a undirected graph $G = \{V, E\}$, where V is a set of vertices and E is a set of edges, two vertices are said to be *adjacent* when they are connected by an edge. The graph is called *complete* if any two of its vertices are adjacent, and informally is also called a *clique*. A typical problem is the obtaining of the maximum subset of vertices in V conforming a clique.

Following our previous work on bitboard search models, we adapted our general purpose bit encoded algorithm [9] to the specific scenario of a graph generated in a global localization problem [10]. This same approach is adapted in this work for solving the problem of establishing correspondences between two submaps. Our approach is bit-encoded oriented, and exactly solves the MCP exploiting the parallelism between a computer architecture and the logic that arises when comparing two sets of landmarks.

The search of a correspondence between features contained in two different sets (submaps) can be formulated as the search of a maximum clique in a graph where:

- Each **vertex** is a **pair of landmarks**, each of them corresponding to a different submap.
- Each **edge** indicates the **compatibility** between the two pairs of associations considered.

More formally, let $|L_1| = m$, $|L_2| = n$ be the cardinalities of both sets of landmarks, with elements $L_1 = \{l_{1,1}, l_{1,2}, \dots, l_{1,m}\}$, and $L_2 = \{l_{2,1}, l_{2,2}, \dots, l_{2,n}\}$. A bit-graph model $G = \{V, E\}$ is generated with $V = \{v_1, v_2, \dots, v_{m \times n}\}$, where the elements are all possible combinations containing one element from each set.

Vertices in this graph are connected using a generalized distance D , independent from the global localization of the sets of landmarks, that considers only relative position and/or relative orientation. This distance is a function applied to every pair of vertices in V , returning 1 when distances between pair of observations are similar to distances between pairs of landmarks. Specifically, having two vertices $v_i = \{l_{1,p}, l_{2,q}\}$, $v_j = \{l_{1,r}, l_{2,s}\} \in V$, $i \neq j$, we define a function $D : \{V, V\} \rightarrow \{0, 1\}$ such that:

$$D(v_i, v_j) = \begin{cases} 1 & \text{if } d(l_{1,p}, l_{1,r}) \approx d(l_{2,q}, l_{2,s}) \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Every element of the adjacency matrix M of G encodes the potential matching of every pair of elements belonging to each of the considered submaps, with the rest of possible pairings. A maximum clique found in the graph this way generated finds an optimum solution to the correspondence search. Using a bit encoded formulation, exploiting the architecture of current computers, very impressive results can be obtained for the solution of this problem, outperforming classical solutions such as the GCBP [6], making possible real time implementations. A detailed description of the BE-MCP solution can be found in [10].

B. Features Used in Map Matching

What features will be used in the matching process? The direct use of control points does not seem to be a good option, as the same geometry can be described using different sets of control points, depending on the chosen parameterization. Presence of occlusions, or the simple fact that features are built using different sets of data for different submaps, makes the direct utilization of these points a risky option. Nevertheless, the availability of a parametric representation is once more an advantage more than a drawback, as the geometric analysis of the features can provide rich information, allowing the extraction of simpler features to be used in the matching process, as we will see. Moreover, the number of these simple entities is in general much lower than the total number of control points defining the original spline.

In complex indoor environments, no matter how curved the architectural elements are, it is always possible to approximate their appearance by sets of sampled points, or concatenations of simpler features:

- **Points:** A rough map representation can be obtained by sampling every spline with a given spacing along the parameter of the curve. This representation, though quite simple, implies an unnecessary growth in the number of features used in the matching process. However, points can be very useful where the complexity of the

environment makes unfeasible the extraction of more compact entities. They can also be used for representing elements such as corners, not considered in this work.

- **Points with curvature.** Points with curvature are simple points that have associated the curvature of the feature at that specific location. This obviously provides rich information for the matching process, as a point present in a concave location, for example, will never be associated with a point in a flat (segment) or convex location. This can eventually avoid ambiguities in the matching result.
- **Segments:** The analysis of the curvature can reveal the presence of simple straight elements. In our experiments, a curvature threshold of $0.01m^{-1}$ was used for segment detection.
- **Arc circles.** Points of a feature whose curvature is over the previously mentioned threshold, can be assumed as belonging to a curved span. Defining intervals of curvature, these points can be grouped into sets, and used for generating an arc circle which roughly describes the element in that particular area.

In [10] simple point features were used in the experiments. In this work, we rely on the extraction of segments and arcs as the basic feature to be used in the matching process.

C. Extracting Simple Features from Splines

All previous simple features can be easily obtained from the analysis of the curvature of every spline. An example of a simple hand drawn map, containing one long curve and two straight elements, and the corresponding arcs and segments obtained from the analysis of its geometry can be observed in Fig. 2. The computation of the curvature of a spline at a given point is a simple derivative process, using the well known formula for the curvature. For a spline curve given parametrically as $s(t) = [s_x(t), s_y(t)]$ the curvature at any given location t can be computed as:

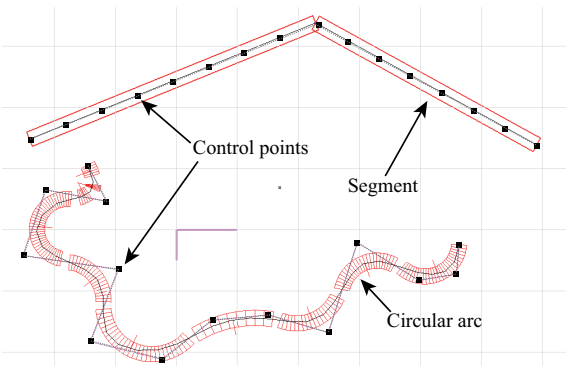


Fig. 2. Sample map composed by three splines defined by a total of 31 control points. The analysis of the map reveals the presence of 12 simple features (2 segments and 10 circular arcs) that accurately describe the map.

$$C(t) = \frac{s'_x s''_y - s'_y s''_x}{(s'^2_x + s'^2_y)^{3/2}} \quad (8)$$

This features extraction is done for each submap, right after its maximum size has been reached, using a fixed parameter step for sampling along each curve, and a binary search for fine location of the terminal points of segments and arcs.

D. Invariant relationships

The construction of an adjacency matrix requires of some rules to define whether two relations are compatible or not. The basic procedure for obtaining it is to compare the main metrics that define each relation, and to test whether they are below a certain threshold or not.

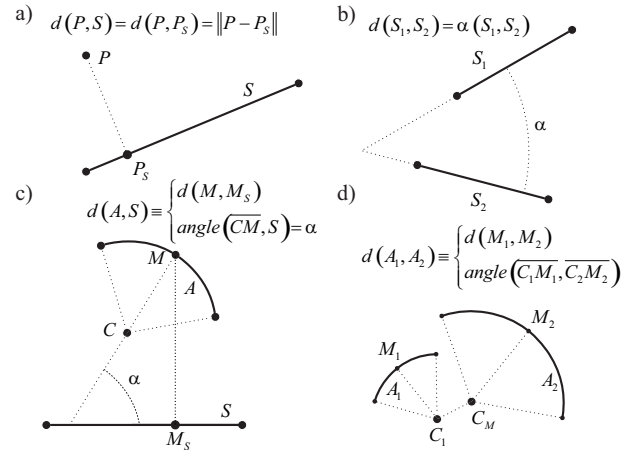


Fig. 3. Several examples of relations established between simple features.

An example with real data of the matching obtained between the features detected during an observation, and the features contained in a map of the environment, is presented in Fig. 4.

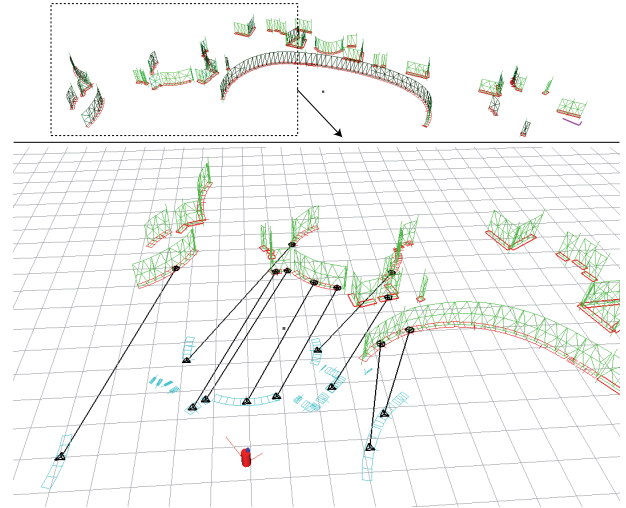


Fig. 4. Matching Example. A set of 26 observations (blue, flat representation) matched against a set of 68 landmarks (red, 3D representation). Most of the observations are successfully paired with their corresponding landmarks in just 0.06 seconds.

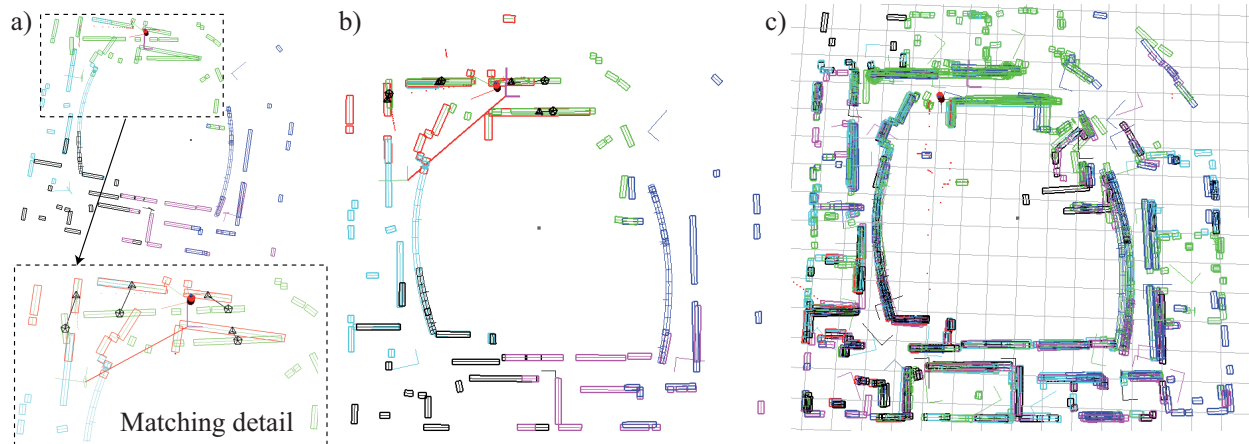


Fig. 5. Map of the Intel Research Lab in Seattle. Submaps are limited to 80 control points. a) Situation just before the first loop closure and features matching detail. Each submap is represented in a different color. b) Submaps positions are updated with the information obtained from the first loop closure matching. c) Final map configuration after three exploring loops.

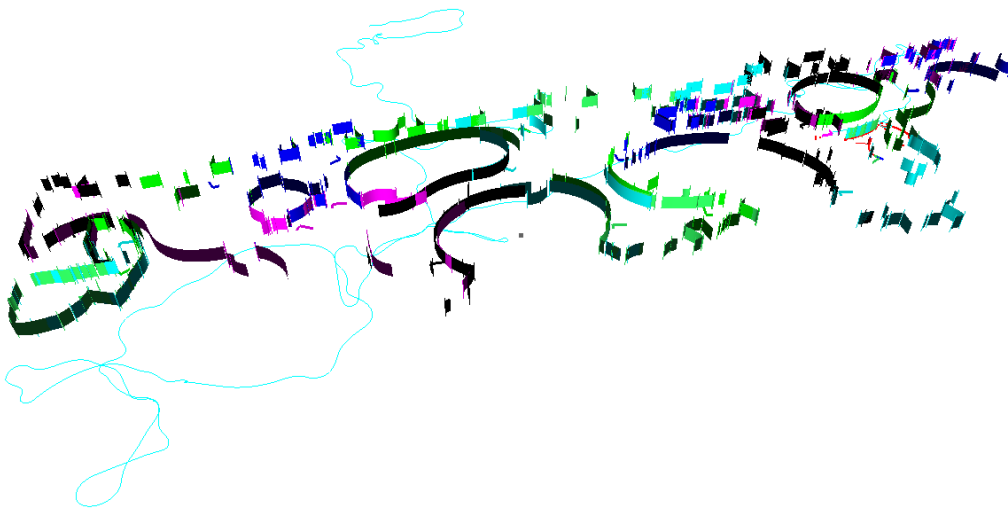


Fig. 6. Map of the Museum "Prince Felipe" in the City of Arts and Sciences, Valencia (Spain). Each submap contains 60 control points. Odometric trajectory is displayed in blue.

V. EXPERIMENTAL RESULTS

Concepts and methodologies presented in this paper have been tested with several real data sets, and very promising results have been obtained. Fig. 5 shows a map of the Intel research lab. (data set was obtained from [20], thanks go to D. Fox for providing this data) using submaps with 80 control points and updating their relative positions as described before. The quality is comparable to the map obtained using a single map, but the time required for its constructions is clearly much lower, allowing a real time implementation. Detail view in Fig. 5.a shows the results of the matching process right after the first loop has been completed in the upper-left corner of the map, and Fig. 5.b displays the results after updating the local maps with this observation. Fig. 5.c shows the global map after three exploring loops, updating the submaps positions whenever a good data association is obtained.

Fig. 6 shows a map of large size, covering an area of about $180 \times 50 \text{ m}^2$. The final result offers a structure topologically coherent with the real configuration of the environment. For using these maps the robot would simply transition from one submap to another, using the small pieces for its localization.

VI. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

This paper has presented a method for building maps of large environments using a *collage* of maps of limited size. This way, the computational cost of an extended Kalman filter used for building every single map is bounded. Individual maps are built using spline curves as a mathematical tool for describing complex geometries present in the environment. In this local representations, the state vector of each submap is composed by the robot pose and the set of control points

defining the splines that accurately represent the shapes detected by a laser range finder.

It has been shown the usefulness and importance of parametric representations, allowing the analysis of the shapes contained in each submap, making feasible the subsequent extraction of simpler features such as segments, or arc circles. The use of splines in the map building process avoids the need of relying on the search of specific geometries, while the obtaining of these latter features is always possible and efficient once the map is available.

Sets of points, segments, or circular arcs are extracted from every submap, and used for establishing correspondences between maps with similar appearance. This search is optimized using a bit-encoded formulation of the relations graph, outperforming current algorithms used with similar purposes, and making possible real time implementations. The search of a maximum correspondence between the associations in an observation, and the associations established between the elements of each the rest of submaps successfully enhances the solution to the difficult problem of data association. Solution obtained for the MCP is exact, and does not rely on previous assumptions or knowledge about the relative positions of landmarks and observations. However, this information could improve the results, specially when the amount of common features in two different submaps is reduced.

Correspondences between maps are used for generating spatial relations that, used as observations of the relative poses of the maps being matched, are used for updating the whole structure of the map. Hence, a real time implementation of a SLAM algorithm in environments of increasingly large dimensions is possible.

B. Future Works

The exploitation of the geometric possibilities that a map representation in parametric form offers has just begun. We believe that this reasoning over the maps built by an autonomous robot can greatly contribute to the knowledge and understanding the machine has of its environment. Further work will address the advantages of this representation for extracting more complex information from the world as the robot observes it.

Once the spatial information about the environment is available as a set of segments, arcs, corners or any other geometric entity easily parameterizable, it is possible to establish relations between them. Distances, relative positions, lengths... can be used to infer the existence of even higher level objects such as corridors, rooms, doors, windows or stairs. This could eventually contribute to the enhancement of a knowledge modeling of the environment and other areas of robotics research such as autonomous exploration.

Future research will address the problem of updating the relative positions of the constellation of submaps using novel methods recently appeared [21], [22]. Finally, these results are currently being extended to 3D scenarios.

VII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the supervision and collaboration of CACSA (*City of Arts and Sciences*, Valencia,

Spain) in this work.

REFERENCES

- [1] 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.
- [2] G. Dissanayake, H. Durrant-Whyte, and T. Bailey, "A computationally efficient solution to the simultaneous localization and map building (SLAM) problem," in *IEEE International Conference on Robotics and Automation, 2000*, vol. 2, 2000, pp. 1009–1014.
- [3] J. Leonard and H. Feder, "A computationally efficient method for large-scale concurrent mapping and localization," in *International Symposium on Robotics Research*, D. K. J. Hollerbach, Ed., 1999.
- [4] M. Bosse, P. M. Newman, J. J. Leonard, M. Soika, W. Feiten, and S. J. Teller, "An Atlas framework for scalable mapping," in *ICRA, 2003*, pp. 1899–1906.
- [5] C. Estrada, J. Neira, and J. D. Tardós, "Hierarchical SLAM: Real-time accurate mapping of large environments," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 588–596, 2005.
- [6] J. Neira, J. D. Tardós, and J. A. Castellanos, "Linear time vehicle relocation in SLAM," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation, ICRA 2003*, Taipei, Taiwan, 2003, pp. 427–433.
- [7] T. Bailey, E. Nebot, J. Rosenblatt, and H. Durrant-Whyte, "Data association for mobile robot navigation: a graph theoretic approach," in *IEEE Int. Conf. Robotics and Automation*, vol. 3, San Francisco, CA, USA, April 2000, pp. 2512–2517.
- [8] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979.
- [9] P. San Segundo, R. Galán, F. Matía, D. Rodríguez-Losada, and A. Jiménez, "Efficient search using bitboard models," in *ICTAI '06: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 132–138.
- [10] P. San Segundo, D. Rodríguez-Losada, R. Galán, F. Matía, and A. Jiménez, "Efficient global localization by searching a bit-encoded graph," in *IAV'07*, 2007.
- [11] L. Pedraza, G. Dissanayake, J. Valls Miró, D. Rodríguez-Losada, and F. Matía, "BS-SLAM: Shaping the world," in *Proc. Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.
- [12] 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.
- [13] 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.
- [14] D. Rodríguez-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.
- [15] 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.
- [16] L. Piegl and W. Tiller, *The NURBS Book (2nd ed.)*. New York, USA: Springer-Verlag New York, Inc., 1997.
- [17] C. de Boor, *A Practical Guide to Splines*, J. Marsden and L. Sirovich, Eds. Springer, 1978.
- [18] L. Paz, J. Guivant, J. Tardós, and J. Neira, "Data association in O(n) for divide and conquer SLAM," in *Proc. Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.
- [19] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I. J. Cox and G. T. Wilfon, Eds. New York, NY, USA: Springer-Verlag New York, Inc., 1990, pp. 167–193.
- [20] A. Howard and N. Roy. (2003) The robotics data set repository (Radish). [Online]. Available: <http://radish.sourceforge.net>
- [21] E. Olson, J. Leonard, and S. Teller, "Fast iterative optimization of pose graphs with poor initial estimates," in *IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2006, pp. 2262–2269.
- [22] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.