Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments

Carlos Estrada, José Neira, and Juan D. Tardós

Abstract—In this paper, we present a hierarchical mapping method that allows us to obtain accurate metric maps of large environments in real time. The lower (or local) map level is composed of a set of local maps that are guaranteed to be statistically independent. The upper (or global) level is an adjacency graph whose arcs are labeled with the relative location between local maps. An estimation of these relative locations is maintained at this level in a relative stochastic map. We propose a close to optimal loop closing method that, while maintaining independence at the local level, imposes consistency at the global level at a computational cost that is *linear* with the size of the loop. Experimental results demonstrate the efficiency and precision of the proposed method by mapping the Ada Byron building at our campus. We also analyze, using simulations, the precision and convergence of our method for larger loops.

Index Terms—Large maps, local maps, loop closing, stochastic mapping.

I. INTRODUCTION

OVER THE past few years, there has been an increasing interest in reducing the computational time and memory requirements when performing simultaneous localization and mapping (SLAM) in large areas (see [1] and [2] and the references therein). The method presented here, hierarchical SLAM, addresses the complexity of mapping large areas by building a set of independent local stochastic maps. By limiting the maximum local map size, this can be carried out in constant time per step. The system is completed with an upper global level containing a graph whose nodes correspond to the local maps and whose arcs represent adjacency relations. The metric information about the relative location between adjacent maps is also stored at the global level using a stochastic map representation. An additional advantage of this approach is its natural extension to multirobot map building: several robots can contribute new local maps or refine previously built local maps.

Several current methods address the computational complexity problem by working on a limited region of the map. Postponement [3] and the compressed filter [4] significantly reduce the computational cost without sacrificing precision, although they require an $O(n^2)$ step on the total number of landmarks to obtain the full map. The split covariance intersection method [5] limits the computational burden but sacrifices

precision: it obtains a conservative estimate. The sparse extended information filter [6] is able to obtain an approximate map in constant time per step, except during loop closing. All cited methods work on a single absolute map representation and confront divergence due to nonlinearities as uncertainty increases when mapping large areas [7].

In contrast, sequential map joining [8] and the constrained local submap filter [9] propose to build stochastic maps relative to a local reference, guaranteed to be statistically independent. By limiting the size of the local map, this operation is constant time per step. Local maps are joined periodically into a global absolute map, in an $O(n^2)$ step. Given that most of the updates are carried out on a local map, these techniques reduce the harmful effects of linearization [7].

The constrained relative submap filter (CRSF) [10] proposes to maintain the local map structure. Each map contains links to other neighboring maps, forming a tree structure (where loops cannot be represented). The method converges by revisiting the local maps and updating the links through correlations. In the Atlas system [1], the constant-time SLAM (CTS) algorithm [11], and our approach, the links between local maps form an adjacency graph. The main difference is that neither the Atlas system nor CTS impose loop consistency in the graph, sacrificing the optimality of the resulting global map. Instead, the location of a local map relative to a given base is computed by composing transformations along the most precise path throughout the graph. Bailey [12] proposes a similar approach called the network coupled feature maps (NCFM). The coupling (relative location) between two adjacent local maps is estimated using the constraints imposed by common features found by a batch data association algorithm. However, the method ignores the correlations appearing between the coupling estimates. The author conjectures that this still produces consistent results, but no proof or experimental validation is provided.

It is well known that, even in small environments, imposing loop consistency increases the precision of the resulting map [13]. However, closing large loops is an especially difficult task. The first key problem is the reliable detection of loops. This requires data association techniques that do not rely on a precise estimation of the vehicle location. In this study, we use the RS linear time relocation algorithm of [14]. The reader may consult this paper for additional details and further references on the subject.

The second key problem is that linearized methods like the extended Kalman filter (EKF) fail to obtain accurate map estimations due to the large uncertainties appearing in large loops. Several researchers have addressed the problem of loop closing while building metric maps. In [15], the authors make use of the expectation-maximization (EM) algorithm to simultaneously

The authors are with the Department of Computer Science and Systems Engineering, University of Zaragoza, 50018 Zaragoza, Spain (e-mail: cestrada@unizar.es; jneira@unizar.es; tardos@unizar.es).

estimate the map and the pose of the robot. This methodology works with a global map and has a high computational cost. The work in [16] proposes an incremental mapping method that uses scan matching and global correlation to build a graph of adjacent robot poses. For closing loops, they use consistent pose estimation [17], whose time complexity is $O(n^3)$ on the number of robot poses, making the method unsuitable for real-time execution in large environments. More recently, a similar approach to build consistent maps with many cycles has been proposed in [18]. This method obtains correspondences between vehicle poses using the iterative closest-point algorithm. Using a quadratic penalty function, correspondences are incorporated into an optimization algorithm that recomputes the whole trajectory. This process is iterated until convergence. However, the method requires the estimated vehicle poses before loop closing to be nearby, limiting the size of the loops that can be successfully closed. Neither computing time not theoretical computational complexity are reported.

In this paper, we present a new method to impose loop consistency at the global map level that makes careful use of stochastic map techniques. Due to the large uncertainties appearing in big loops, nonlinearities become important, making the classical EKF technique obtain inaccurate estimations. Nonlinearities have been successfully addressed using standard *nonlinear least-squares optimization* techniques in the context of bearing-only SLAM [19] and range-only SLAM [20]. We show that employing similar techniques with an appropriate representation for the problem structure produces close to optimal results at the global map level at a low computational cost. The relative representation of the locations at this upper level introduces highly sparse matrices that can be exploited in the calculation process by means of specialized sparse methods. Moreover, this efficient local parametrization contributes to make the method more robust and stable. We are able to obtain the optimal solution to the loop-consistency problem in $O(n)$ with n being the loop length (number of local maps in the loop). We also show that our scheme can be derived from a general formulation of the problem using the iterated EKF (IEKF) detailed in [21]. Experimental results validate our method: we obtain a close to optimal global map of a building with several big loops (in the 200–350 m range) in less than 1 s. We have also carried out simulations that assert the accuracy of hierarchical SLAM for loops up to 3.6 km long.

This paper is organized as follows. After the introduction in Section I, the fundamental aspects of the hierarchical SLAM model are discussed in Section II. The process to impose loop consistency at the global level is presented in Section III. Section IV describes the experiments conducted to validate our approach. In Section V, we discuss the advantages and limitations of hierarchical SLAM with respect to other methods. Finally, in Section VI, we draw the main conclusions of this study and outline future research directions.

II. HIERARCHICAL SLAM

The hierarchical map proposed in this paper is an environment representation at two levels. In the following, each level in the hierarchy, as well as the process of map maintenance that assures consistency, are described in detail.
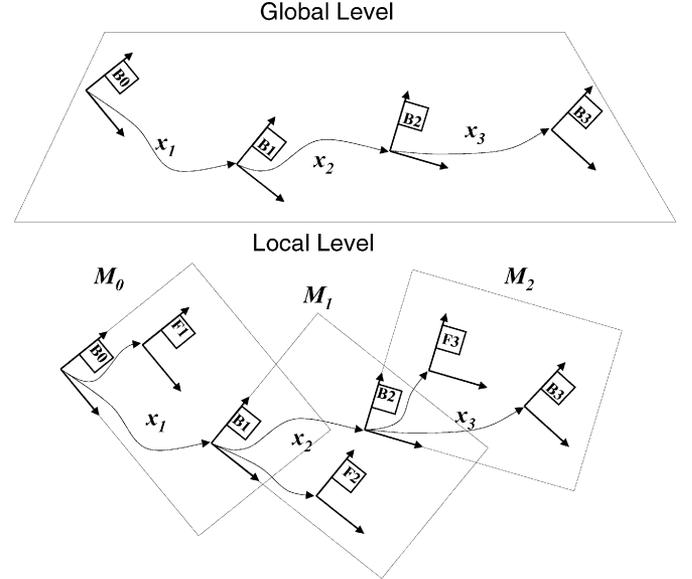
Fig. 1. Two-level hierarchical SLAM model.

A. Local Level

The fundamental building blocks of our map representation at the local level are feature-based *local* stochastic maps of different regions of the environment (Fig. 1, bottom), guaranteed to be of limited size and mutually independent in the statistical sense at all times. A local map $\mathcal{M} = (\hat{\mathbf{x}}_{\mathcal{F}}^B, \mathbf{P}_{\mathcal{F}}^B)$ contains information related to a set of n elements in the environment, where

$$\hat{\mathbf{x}}_{\mathcal{F}}^B = \begin{bmatrix} \hat{\mathbf{x}}_R^B \\ \vdots \\ \hat{\mathbf{x}}_{F_n}^B \end{bmatrix}; \qquad \mathbf{P}_{\mathcal{F}} = \begin{bmatrix} \mathbf{P}_R^B & \cdots & \mathbf{P}_{RF_n}^B \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{F_n R}^B & \cdots & \mathbf{P}_{F_n F_n}^B \end{bmatrix}. \quad (1)$$

The state vector $\hat{\mathbf{x}}_{\mathcal{F}}^B$ contains the estimated pose of the vehicle R, $\mathbf{x}_R^B = (x, y, \theta)^T$, and the location of the environment features $\mathcal{F} = \{F_1 \dots F_n\}$, all with respect to a base reference B. The parameters that represent this location depend on the feature type [8]. The base reference may be the initial current vehicle location when the map is initialized or it may be associated with a set of local features, such as a corner or a pair of points.

Furthermore, each local map may contain the position and orientation of the base reference of neighboring maps, relative to the local base. For simplicity of notation, we will denote the relative transformation between two consecutive maps as $\mathbf{x}_j = \mathbf{x}_{B_j}^{B_j - 1}$ and its corresponding covariance matrix as $\mathbf{P}_j$. For example, in Fig. 1, bottom, local map $\mathcal{M}_1$ contains the estimation of $\mathbf{x}_2 = \mathbf{x}_{B_2}^{B_1}$ and its corresponding covariance.

Local maps are required to be *mutually independent* at all times for this representation to be consistent. This condition is guaranteed during the different stages of the map building process described in the following subsections.

B. Global Level

At the upper (or global) level, the topology of the environment is represented by a graph in which each node i corresponds to a local map $\mathcal{M}_i$ of the local level. An arc ij in the graph represents a known topological relation between local maps $\mathcal{M}_i$ and $\mathcal{M}_j$ detected during the mapping process. The

properties of these topological relations are the relative transformations between the base reference of both maps, $\mathbf{x}_{ij} = \mathbf{x}_{B_j}^{B_i} = (x_{ij}, y_{ij}, \theta_{ij})^T$. At this global level, these relative transformations are maintained in a *relative* stochastic map $\mathcal{X}_u = (\hat{\mathbf{x}}_u, \mathbf{P}_u)$ (the u subscript stands for *unconstrained*, since no loop closing constraints are imposed in this map), where

$$\hat{\mathbf{x}}_u = \begin{bmatrix} \vdots \\ \hat{\mathbf{x}}_{ij} \\ \vdots \end{bmatrix}; \qquad \mathbf{P}_u = \begin{bmatrix} \ddots & 0 & 0 \\ 0 & \mathbf{P}_{ij} & 0 \\ 0 & 0 & \ddots \end{bmatrix}. \qquad (2)$$

Every time a local map is created or modified, the new values are included in this relative stochastic map. Since all local maps are independent, matrix $\mathbf{P}_u$ is block diagonal by construction. More accurate estimates are calculated at the global level when imposing loop constraints (Section III). These new estimates are maintained at the global level in a separate relative stochastic map $\mathcal{X} = (\hat{\mathbf{x}}, \mathbf{P})$. Every time a loop is detected, the constrained stochastic map $\mathcal{X}$ is recomputed by minimizing the difference with respect to the unconstrained map $\mathcal{X}_u = (\hat{\mathbf{x}}_u, \mathbf{P}_u)$, using the last value of $\hat{\mathbf{x}}$ as seed for the optimization process. As will be proved in Section III-A, this allows us to maintain the computational cost of loop closing linear with the size of the loop.

C. Building Local Maps

During continuous SLAM, local maps are sequentially built in the following way (see [8] for a more detailed explanation of the process): at a given instant t_j, a new map is initialized using the current vehicle location as base reference B_j. Then, the vehicle continues moving and acquiring sensor information about the environment features $\mathcal{F}_j$. At each step, data association is robustly solved using the joint compatibility test [22], and the local map is updated with standard stochastic mapping techniques [23].

The resulting local map $\mathcal{M}_j$ is independent of any prior map $\mathcal{M}_i$ because it is built relative to the initial vehicle location at t_j and depends only on the sequence of odometry readings and sensor data obtained during the involved steps. Under the common assumption that process and measurement noise are *white* random sequences, two local maps built with the same robot from *disjoint* sequences of steps are functions of independent stochastic variables. Therefore, the two maps will be *statistically independent* and *uncorrelated*.

The very important consequence is that, during local map building, we do not need to compute the correlations between features in the current local map and features in any other local map, because they are known to be zero by construction. Thus, if the size of local maps is bounded, the cost of local map building is constant at each step, independent of the size of the global map.

The decision to close map $\mathcal{M}_j$ and start a new local map is made once the number of features in the current local map reaches a maximum, or the uncertainty of the vehicle location with respect to the base reference of the current map reaches a limit, or no matchings were found by the data association process for the last sensor measurements (a separate region of the environment is observed). Note that the new local map
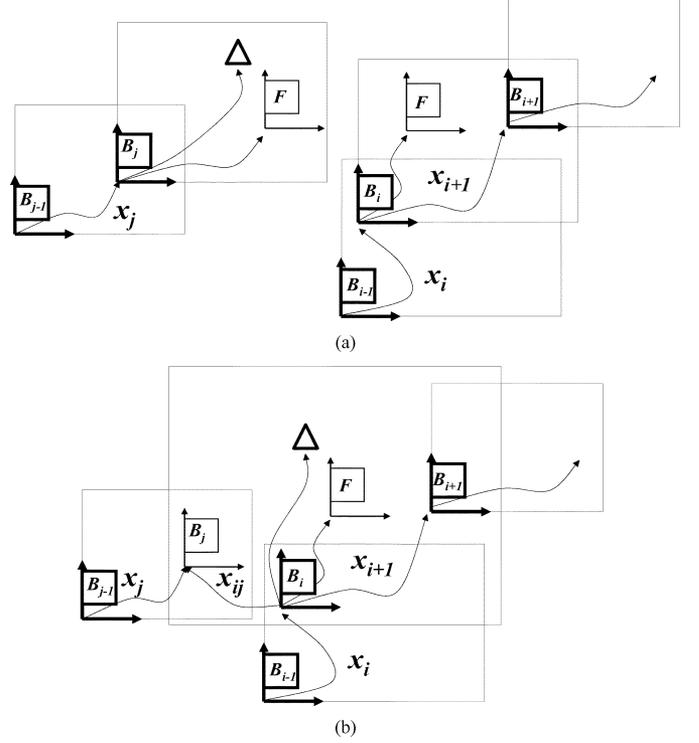
Fig. 2. (a) Map building before closing a loop on maps $\mathcal{M}_i$ and $\mathcal{M}_j$. (b) Map building after closing a loop by joining and fusing map $\mathcal{M}_j$ into $\mathcal{M}_i$.

$\mathcal{M}_{j+1}$ will have the current vehicle position as a base reference, which corresponds to the last vehicle position in map $\mathcal{M}_j$. This provides the relative transformation $\mathbf{x}_{j+1} = \mathbf{x}_{B_{j+1}}^{B_j}$, corresponding to the topological relation between the adjacent maps $\mathcal{M}_j$ and $\mathcal{M}_{j+1}$. This relation can be safely copied to the global relative stochastic map $\mathcal{X}_u$ because it is independent from any other relative location already present at this level.

D. Loop Closing

The sequential mapping process allows us to have an estimation of the current vehicle position with respect to any previous local map base reference. This allows to hypothesize that a previously mapped region of the environment, say corresponding to map $\mathcal{M}_i$, is being revisited, and thus a loop is being closed. To confirm this hypothesis and discard alternative ones with neighboring maps, we use the RS relocation algorithm proposed in [14]. It has a very low false positive rate, except for environments with a high degree of symmetries. If considerable overlap is found with more than one local map, loop closing should be delayed until enough information is gathered to resolve the ambiguity.

Once a loop-closing hypothesis has been confirmed, a common reference F corresponding to a feature (for example, a corner) observed in both maps is selected [Fig. 2(a)]. Given that by construction both maps are independent, local map joining [8] is used to join and fuse them [Fig. 2(b)]. Map $\mathcal{M}_i$ will still have B_i as the base reference and will include an estimation of reference B_j with respect to B_i, $\mathbf{x}_{ij} = \mathbf{x}_{B_j}^{B_i}$. Map $\mathcal{M}_{j-1}$ also contains an estimation of reference B_j, $\mathbf{x}_j = \mathbf{x}_{B_j}^{B_{j-1}}$. This gives us a new topological relation between maps i and $j-1$ that

allows us to add a new link at the global level of the hierarchical map. This provides the information necessary to close the loop.

It may, of course, be the case that map $\mathcal{M}_i$ already has topological relations with other maps, in particular with map $\mathcal{M}_{i+1}$. In this case, the stochastic map in the global level before closing the loop would be

$$\hat{\mathbf{x}}_u = \begin{bmatrix} \vdots \\ \hat{\mathbf{x}}_{i+1} \\ \vdots \\ \hat{\mathbf{x}}_j \end{bmatrix}; \qquad \mathbf{P}_u = \begin{bmatrix} . & 0 & 0 & 0 \\ 0 & \mathbf{P}_{i+1} & 0 & 0 \\ 0 & 0 & . & 0 \\ 0 & 0 & 0 & \mathbf{P}_j \end{bmatrix}.$$

As a result of joining and fusing map $\mathcal{M}_j$ into map $\mathcal{M}_i$, the estimated relative location $\mathbf{x}_{B_{i+1}}^{B_i}$ will be updated in map $\mathcal{M}_i$, and its correlation with $\mathbf{x}_{B_j}^{B_i}$ will also be computed. This allows us to update the estimation of $\mathbf{x}_{i+1} = \mathbf{x}_{B_{i+1}}^{B_i}$ in the stochastic map at the global level, and simultaneously copy the new link $\mathbf{x}_{ij} = \mathbf{x}_{B_j}^{B_i}$ and its correlation $\mathbf{P}_{ij}$, with $\mathbf{x}_{B_{i+1}}^{B_i}$, $\mathbf{P}_{i+1,ij} = \mathbf{P}_{B_{i+1}B_j}^{B_i}$ (and any other relative location already in both levels for that matter). The result will be

$$\hat{\mathbf{x}}_u = \begin{bmatrix} \vdots \\ \hat{\mathbf{x}}_{i+1} \\ \hat{\mathbf{x}}_{ij} \\ \vdots \\ \hat{\mathbf{x}}_j \end{bmatrix}; \qquad \mathbf{P}_u = \begin{bmatrix} . & 0 & 0 & 0 & 0 \\ 0 & \mathbf{P}_{i+1} & \mathbf{P}_{i+1,ij} & 0 & 0 \\ 0 & \mathbf{P}_{i+1,ij}^T & \mathbf{P}_{ij} & 0 & 0 \\ 0 & 0 & 0 & . & 0 \\ 0 & 0 & 0 & 0 & \mathbf{P}_j \end{bmatrix}.$$

Note that map $\mathcal{M}_i$ remains independent from any other local map, guaranteeing mutual independence at the local level.

Once the map fusion is performed at the local level, a loop is obtained at the global level. However, due to the large uncertainties appearing in big loops, the composition of the successive relative transformations between maps may be inconsistent. In Section III, a method for imposing consistency at the global level will be detailed.

E. Revisiting a Previous Local Map

Sometimes the vehicle arrives at a previous local map through an existing topological relation. This happens either when the vehicle is turning back, or when traversing a loop for the second time. In this case, the relocation algorithm is used to confirm the revisiting of the map, and to determine the current location for the vehicle in that map. The vehicle can then proceed to update the map as it is being traversed.

During this process, the local estimation of the relative transformation between the map and its neighbors is also improved through correlations [10]. In our system, when possible, the base reference of a local map is associated to an observable feature instead of the initial vehicle location when the map was created. This allows to further improve the link estimate through direct observation. The procedure for changing the base reference of the map to a feature shared with the previous map is described in [8]. Once the vehicle leaves the map, the relative transformations are updated in map $\mathcal{X}_u$. If the update is important, loop closing constraints can be applied again (Section III) to obtain a more accurate estimation of $\mathcal{X}$.
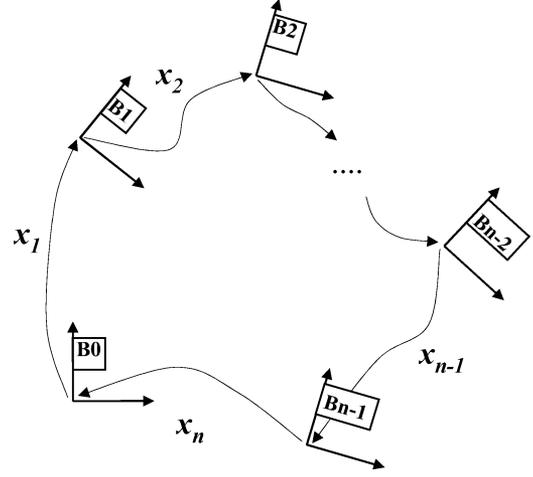
Fig. 3. Loop of relative transformations in the global level.

III. IMPOSING LOOP CONSTRAINTS

Suppose that a sequence of n local maps have been built, and thus we have obtained $\mathcal{X}_u = (\hat{\mathbf{x}}_u, \mathbf{P}_u)$ where $\mathbf{P}_u$ is block diagonal. Suppose an overlap has been detected between the first and the last maps. To obtain a consistent estimation at the global level $\mathcal{X} = (\hat{\mathbf{x}}, \mathbf{P})$, we impose the condition that the composition of transformations along the loop be zero (Fig. 3) as

$$\mathbf{h}(\mathbf{x}) \equiv \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_{n-1} \oplus \mathbf{x}_n = \mathbf{0} \qquad (3)$$

where $\mathbf{x}$ is the state vector of the relative transformations between the local maps that form the loop

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}. \qquad (4)$$

Given that $\mathbf{h}(\mathbf{x})$ is a composition of transformations, it is nonlinear due to the angular terms. When the loop is big, linearization errors become important, and the linear techniques employed at the local map level are no longer valid to compute an accurate estimation of $\mathbf{x}$.

A. Nonlinear Constrained Least-Squares Optimization for One Loop

In order to confront these nonlinearities, a different approach is necessary. For this purpose, we formulate the problem as the obtention of the maximum *a posteriori* likelihood estimation of the relative locations at the global level, given the loop constraint. This can be expressed by means of the following constrained optimization problem:

$$\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}} \frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}_u)^T \mathbf{P}_u^{-1}(\mathbf{x} - \hat{\mathbf{x}}_u)$$
$$\mathbf{h}(\mathbf{x}) = \mathbf{0}. \qquad (5)$$

Several methodologies can be used to find a solution to this problem. Early methods, considered now relatively inefficient, translate the constrained problem to a basic unconstrained problem by using penalty functions. These methods have now

been replaced by methods that focus on the solution of the Kuhn–Tucker equations. Sequential quadratic programming (SQP), derived from the Kuhn–Tucker equations, is one of the most employed (the general SQP expression for a new estimation can be found in Triggs *et al.* [24]). Adapting it for the particular case of the optimization problem (5), the SQP estimation for $\mathbf{x}$ and its associated covariance matrix $\mathbf{P}$ can be obtained by iterating these two equations until convergence

$$\mathbf{P}_i = \mathbf{P}_u - \mathbf{P}_u \mathbf{H}_i^T \left(\mathbf{H}_i \mathbf{P}_u \mathbf{H}_i^T \right)^{-1} \mathbf{H}_i \mathbf{P}_u \qquad (6)$$

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i - \mathbf{P}_i \mathbf{P}_u^{-1} (\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_u) - \mathbf{P}_u \mathbf{H}_i^T \left(\mathbf{H}_i \mathbf{P}_u \mathbf{H}_i^T \right)^{-1} \hat{\mathbf{h}}_i. \qquad (7)$$

As seed $\hat{\mathbf{x}}_0$ for these iterations, we use the best values computed for each link so far. For links that have already been involved in a previous loop constraint, the corresponding values are taken from the constrained map transformations $\hat{\mathbf{x}}$; otherwise, from the unconstrained transformations $\hat{\mathbf{x}}_u$.

Vector $\hat{\mathbf{h}}_i$ is the estimated value of $\mathbf{h}$

$$\hat{\mathbf{h}}_i = \mathbf{h}\left(\hat{\mathbf{x}}_i\right) \qquad (8)$$

and matrix $\mathbf{H}_i$ is the Jacobian of $\mathbf{h}$ evaluated in $\hat{\mathbf{x}}_i$

$$\mathbf{H}_i = \left[\left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}_1} \right|_{\hat{\mathbf{x}}_i} \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}_2} \right|_{\hat{\mathbf{x}}_i} \cdots \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}_{n-1}} \right|_{\hat{\mathbf{x}}_i} \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}_n} \right|_{\hat{\mathbf{x}}_i} \right]. \qquad (9)$$

As $\mathbf{h}(\mathbf{x})$ is a composition of transformations, $\mathbf{H}_i$ is a $3 \times 3n$ matrix that must be recomputed for every iteration. The Jacobian terms are calculated using the expression

$$\begin{aligned} \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}_k} \right|_{\hat{\mathbf{x}}_i} &= \left. \frac{\partial \mathbf{h}}{\partial(\mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_{k-1} \oplus \mathbf{x}_k)} \right|_{\hat{\mathbf{x}}_i} \\ &\quad \cdot \left. \frac{\partial(\mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_{k-1} \oplus \mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{\hat{\mathbf{x}}_i} \\ &= \mathbf{J}_{1\oplus} \left\{ \mathbf{h}_k, \ominus \mathbf{h}_k \oplus \mathbf{h} \right\} |_{\hat{\mathbf{x}}_i} \cdot \mathbf{J}_{2\oplus} \left\{ \mathbf{h}_k \ominus \mathbf{x}_k, \mathbf{x}_k \right\} |_{\hat{\mathbf{x}}_i} \end{aligned} \qquad (10)$$

where $\mathbf{J}_{1\oplus}$ and $\mathbf{J}_{2\oplus}$ represent the Jacobians of the composition of transformations [25], and $\mathbf{h}_k$, used in the last expression to simplify the notation, is defined as

$$\mathbf{h}_k = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_{k-1} \oplus \mathbf{x}_k. \qquad (11)$$

It should be noticed that $\mathbf{P}_i$ is a full matrix, singular due to the loop constraint. A more efficient implementation of the method can be deduced after the substitution of (6) into (7). Rearranging the terms, we obtain a new expression for calculating $\hat{\mathbf{x}}_{i+1}$

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_u + \mathbf{P}_u \mathbf{H}_i^T \left(\mathbf{H}_i \mathbf{P}_u \mathbf{H}_i^T \right)^{-1} \left(\mathbf{H}_i \left(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_u \right) - \hat{\mathbf{h}}_i \right). \qquad (12)$$

Note that there is no need to compute the full matrix $\mathbf{P}_i$ in each iteration. Instead, the block diagonal matrix $\mathbf{P}_u$ is employed. As $\mathbf{H}_i$ is a $3 \times 3n$ matrix, each iteration is *linear* with n, the number of local maps in the loop. Since we are using relative transformations between maps, local maps that do not belong to the loop do not take part in the optimization, making its complexity independent of the total map size.

Should you need to determine the uncertainty of each base reference in a global frame, the full covariance $\mathbf{P}_i$ may be computed at the end of the iterations using (6). However, this involves an $O(n^2)$ computation.

In complex environments with a high degree of symmetry, it is possible that the map-matching algorithm gives a spurious loop closing. In this case, the loop constraint of (3) would force the optimization to converge to a solution far from its true value. In order to detect this event, and thus add robustness to the system, a hypothesis test based on the squared Mahalanobis distance is performed

$$D_{\hat{\mathbf{x}}_i}^2 = (\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_u) \mathbf{P}_u^{-1} (\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_u)^T < \chi_{r,\alpha}^2 \qquad (13)$$

where $\chi_{r,\alpha}^2$ is a threshold value obtained from the χ^2 distribution with $r = \text{rank}(\hat{\mathbf{x}}_i)$, and such that α is the probability of rejecting a good loop closing.

B. Nonlinear Constrained Least-Squares Optimization for Several Loops

The method explained above can be easily generalized to deal with the simultaneous closing of l loops by adding as many constraints as loops to be closed. The loop constraint for the jth loop is then expressed as

$$\mathbf{h}_j(\mathbf{x}) = \mathbf{x}_{j_1} \oplus \mathbf{x}_{j_2} \oplus \cdots \oplus \mathbf{x}_{j_{n_j-1}} \oplus \mathbf{x}_{j_{n_j}} = 0 \qquad (14)$$

where n_j is the number of relative transformations that constitute loop j. The vector $\mathbf{h}$ of composition of transformations along the loops is formed by all the terms $\mathbf{h}_j$ from $j = 1$ to l

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_l \end{bmatrix}. \qquad (15)$$

The state vector $\mathbf{x}$ of relative transformations maintains the expression of (4) where n is now the total number of relative transformations belonging to the whole of the l loops. Any relative transformation that does not form part of a loop in the upper level takes no part in the optimization process.

The solution is obtained by iterating (12), where in this case matrix $\mathbf{H}_i$ is the Jacobian of $\mathbf{h}$ evaluated in $\hat{\mathbf{x}}_i$ as

$$\mathbf{H}_i = \begin{bmatrix} \left. \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}_1} \right|_{\hat{\mathbf{x}}_i} & \left. \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}_2} \right|_{\hat{\mathbf{x}}_i} & \cdots & \left. \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}_n} \right|_{\hat{\mathbf{x}}_i} \\ \left. \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}_1} \right|_{\hat{\mathbf{x}}_i} & \left. \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}_2} \right|_{\hat{\mathbf{x}}_i} & \cdots & \left. \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}_n} \right|_{\hat{\mathbf{x}}_i} \\ \vdots & \vdots & \ddots & \vdots \\ \left. \frac{\partial \mathbf{h}_l}{\partial \mathbf{x}_1} \right|_{\hat{\mathbf{x}}_i} & \left. \frac{\partial \mathbf{h}_l}{\partial \mathbf{x}_2} \right|_{\hat{\mathbf{x}}_i} & \cdots & \left. \frac{\partial \mathbf{h}_l}{\partial \mathbf{x}_n} \right|_{\hat{\mathbf{x}}_i} \end{bmatrix}. \qquad (16)$$

It must be noticed that this matrix becomes sparse as l increases because all the terms $\partial \mathbf{h}_j / \partial \mathbf{x}_k$ where $\mathbf{x}_k$ does not belong to the loop j are null. When implementing (12), the sparsity of both $\mathbf{P}_u$ and $\mathbf{H}_i$ will be considered to optimize the algorithm.

C. Iterated Extended Kalman Filter

Equation (7), and therefore (12), can also be derived from the IEKF algorithm found in Bar-Shalom *et al.* [21]. For adapting this algorithm to our problem, we can represent the detection of a new loop $\mathbf{h}(\mathbf{x})$ as an imprecise measurement $\mathbf{z}$ affected by a small Gaussian noise $\mathbf{w}$

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{w}; \quad \mathbf{w} \sim \mathcal{N}(0, \mathbf{P}_z). \qquad (17)$$
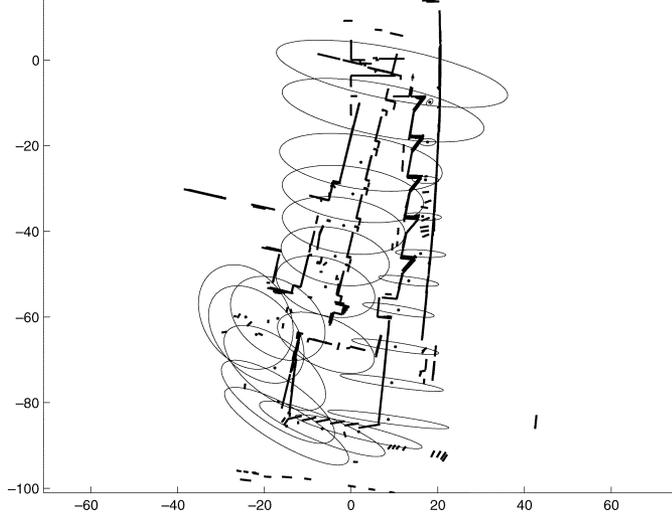
Fig. 4.   Map before closing the first loop.

Fig. 5.   Map after closing the first loop.

The new estimation for $\mathbf{x}$ at iteration $i+1$ and its associated covariance matrix $\mathbf{P}_i$ at iteration i obtained from this algorithm is given by

$$\mathbf{P}_i = \mathbf{P}_u - \mathbf{P}_u \mathbf{H}_i^T \left(\mathbf{H}_i \mathbf{P}_u \mathbf{H}_i^T + \mathbf{P}_z\right)^{-1} \mathbf{H}_i \mathbf{P}_u \qquad (18)$$

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i + \mathbf{P}_i \mathbf{H}_i^T \mathbf{P}_z^{-1} \left(\mathbf{z} - \hat{\mathbf{h}}_i\right) - \mathbf{P}_i \mathbf{P}_u^{-1} \left(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_u\right). \quad (19)$$

Equation (19) can be rearranged, using (18), to produce the following expression, which avoids inverting $\mathbf{P}_z$:

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i - \mathbf{P}_i \mathbf{P}_u^{-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_u)$$
$$+ \mathbf{P}_u \mathbf{H}_i^T \left(\mathbf{H}_i \mathbf{P}_u \mathbf{H}_i^T + \mathbf{P}_z\right)^{-1} \left(\mathbf{z} - \hat{\mathbf{h}}_i\right). \quad (20)$$

Substituting the imprecise measurement $\mathbf{z}$ for the exact loop constraints ($\mathbf{z} = 0$ and $\mathbf{P}_z = 0$) in (18) and (20) will lead to (6) and (7). Thus, using the IEKF with an exact measurement function is equivalent to solving the nonlinear constrained least squares optimization problem (5) using SQP.

IV. EXPERIMENTAL RESULTS

To validate the hierarchical SLAM method, we have conducted an experiment in the Ada Byron building at our campus using a robotized wheelchair equipped with a SICK laser scanner. The vehicle was handdriven along a mixed indoor/outdoor path of 735 m with three loops of about 250, 160, and 325 m, at a mean speed of 0.45 m/s.

First, the scans were processed to obtain line features using a robust segmentation algorithm, and a classical EKF SLAM procedure was used to build each local map. For this demonstration, new local maps are initialized at fixed intervals of about 10 m. This results in a large amount of local maps (84), which allows us to better analyze the performance of the techniques used at the global map level. Fig. 4 shows the map obtained along the first 250 m. Ellipses represent the accumulated uncertainty in the position of the base reference of each local map, computed by composing the relative transformations stored in the global map level.

At this point the vehicle closes a loop for the first time. The loop consistency constraint explained in the previous section was applied at the global map level, obtaining a much more
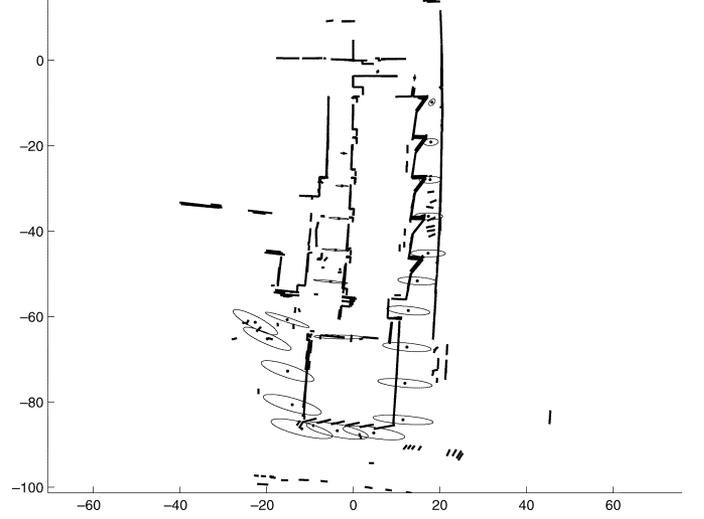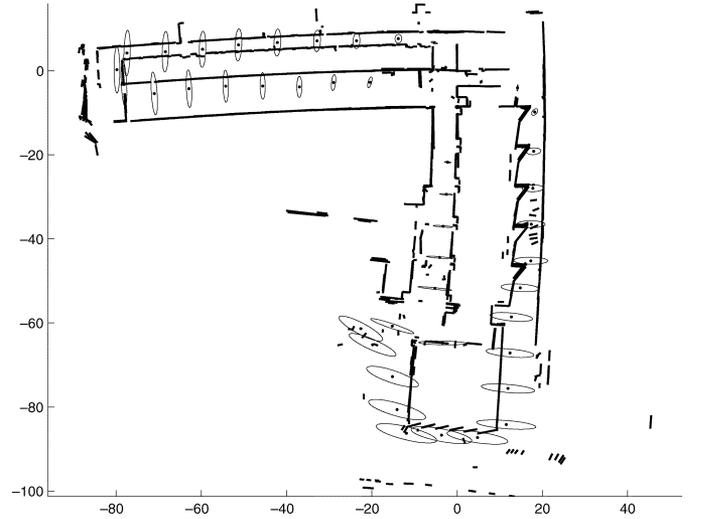
Fig. 6.   Map after closing the second loop.

precise map, shown in Fig. 5. The error is distributed among all relative transformations, and the uncertainty of each location is clearly reduced.

The same procedure was applied after closing the second and third loops, obtaining the maps shown in Figs. 6 and 7. The first two loops have little overlap, and thus the angle between the corresponding two wings in the building is somewhat loosely estimated. The third loop constraint involves submaps already present in the first two, and therefore it enforces rigidity in the map and corrects the deformation. As the third loop involves an outdoor experiment, small inaccuracies are introduced when segmenting vegetation and slopes that can be appreciated as superposing lines in the left part of the map. In the three cases, the method converged in four iterations with computing times of 0.21, 0.46, and 1.12 s using (6) and (7) and 0.15, 0.32, and 0.68 s using (6) and (12) on a Pentium IV 1.7-GHz processor.

To provide a deeper insight into the accuracy and robustness of the hierarchical SLAM for larger loops, we have performed a set of simulations. In each run, the vehicle traverses a square loop, ranging from 40 up to 3.5 km long, and revisits the loop
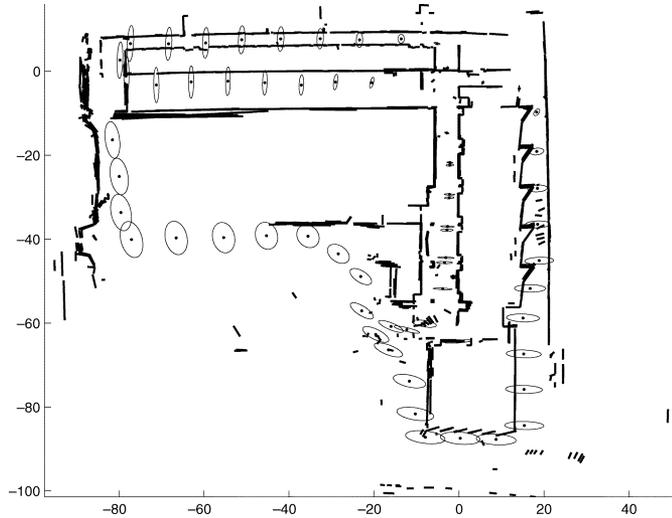
Fig. 7. Map after closing the third loop.

twice. Each local map is generated by assuming a 10-m displacement of the robot, with an associated covariance matrix obtained from the experimental results with the real robot. The relative map $\mathcal{X}_u = (\hat{\mathbf{x}}_u, \mathbf{P}_u)$ at the global level is obtained by adding Gaussian noise to the ground true solution. We run 100 simulations for every loop of a given size. The loop constraint is imposed at the end of each of the three completions of the loop. In all cases, the optimization converged to a local minimum, improving the precision of the initial estimation.

Fig. 8(a) and (b) shows the mean and maximum error (distance from the ground truth) of the position of the corner opposite to the start point for different loop sizes. The four curves represent, from top to bottom, the initial error, the error after the first closing of the loop, the error after the second pass and the error after the third pass. We can see that the accuracy of the estimation is roughly doubled by imposing the first loop constraint. The same effect can be observed in the real world experiments (Fig. 5). For the following passes, the error decreases in proportion to $1/\sqrt{n}$, where n is the number of passes. Error increase is worse than linear with the length of the loop: $e \sim l^{3/2}$.

V. DISCUSSION

In hierarchical SLAM, as local maps are being revisited, their estimates are improved using the standard EKF algorithm. This is known to provide a suboptimal solution, due to linearization errors. Given that the local maps are of limited size, linearization errors have a small effect [7]. At this local level, hierarchical SLAM is equivalent to all other methods based on local maps in the sense that local maps and their relative transformations will converge to the true solution. Roughly, the error will be reduced at a rate of $1/\sqrt{n}$, where n is the number of times a local map has been traversed.

The main difference appears when a loop has been detected at the global level. CRSF [10] simply discards this information. The absolute location of a local map is computed by composing relative transformations along a fixed tree. In Atlas [1] and NCFM [12], the loop is used to decide the shortest path from

the absolute reference to each local map. This only improves the accuracy of local maps whose distance to the absolute reference becomes shorter thanks to the loop detection: roughly the second half of the maps in a single loop. In CTS [11], in order to preserve constant time operation, only the local map currently visited gets its global position improved. In hierarchical SLAM, imposing loop consistency improves the accuracy in the absolute location of *all* local maps in the loop. Furthermore, as seen in the experiments, the accuracy of the map farther from the origin is doubled. All other methods would require at least four passes along the loop to attain this precision.

In hierarchical SLAM, as in other two level mapping systems, the separation between the local and global levels introduces suboptimality in the solution. In order to maintain independence between the local maps, corrections performed at the global level as a result of loop closing are not propagated back to the local level. However, these corrections are very small and would have little incidence in local map precision. The absolute location of a local map can be consistently computed, but if this information is composed with the location of a feature within the local map, the correlations between the local level and the global level are ignored. However, this inconsistency is likely to be small. If necessary, this inconsistency can be avoided by including this feature at the global level and recomputing the loop constraint.

Another cause of suboptimality appears at the frontier between local maps. If the same feature is observed in two neighboring maps, the consistency in its absolute location is not imposed. Imposing this constraint would increase the precision of the local maps and the link between them, but the maps will become correlated. Bailey [12] conjectures that ignoring these correlations still produces consistent results, but gives no proof.

The method used to impose the loop closing in hierarchical SLAM makes optimal use of the information available at the global level. Nonetheless, SQP is a *local* optimization technique that assures convergence to a local minimum, but does not guarantee obtaining the best local minimum, i.e., the *global* solution [26]. In our system, the use of a local parametrization facilitates convergence to the global optimum. Linearization errors are not introduced in the optimization objective function $f(\mathbf{x})$. The nonlinear aspect of the problem is represented by the constraint function $\mathbf{h}(\mathbf{x})$, which is relinearized in each iteration. As long as the initial solution is close enough to the true solution (in particular, if angular errors in the closing of the loop are small), the linearization errors will also be small and the process will converge as shown by our simulations. Using an absolute parametrization leads to introducing linearizations in the objective function. When dealing with big loops, absolute estimation errors due to nonlinearities become important, and convergence is less likely to be attained.

In [18], an absolute parametrization is employed to build consistent maps with many cycles. In order to compute consistent poses in closed form, the relative pose information is transformed into quadratic constraints using a first order approximation. The loop closing matchings are also incorporated with an *ad hoc* quadratic penalty function. The convergence of the optimization is likely to be very sensitive to the initial absolute errors. The success of the experiments presented seems to rely
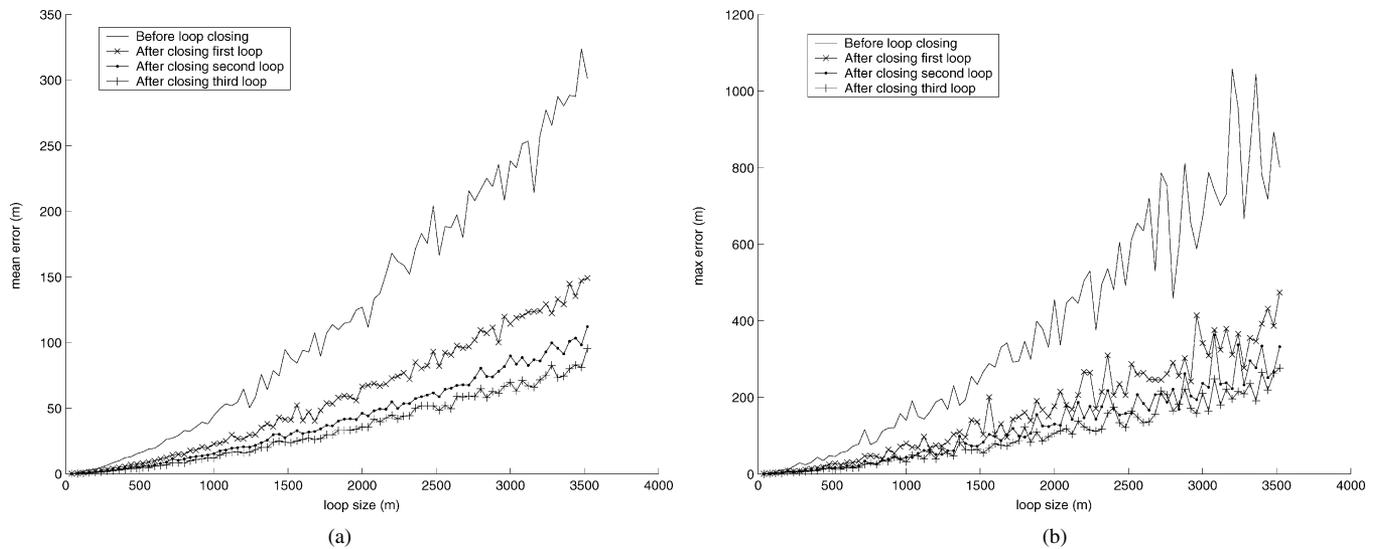
Fig. 8. (a) Mean errors and (b) max errors obtained at the global level in 100 runs for each loop size.

on the accuracy of the scan matching technique that gives accumulated heading errors of less than $1°$.

VI. CONCLUSION

In this paper, we propose a hierarchical mapping method for large-scale SLAM. Its main advantage over previous proposals is the efficient maintenance of loop consistency when calculating the optimal estimation at the global level, which allows us to improve map precision. A robust, stable and local parametrization is established by the use of relative locations, giving the method a good numerical performance near the current state estimate. The algorithm developed is based on a nonlinear constrained least-squares optimization using the SQP method. This problem formulation gives the maximum *a posteriori* likelihood estimation of the relative locations in the global level given the loop constraints. Convergence to the accurate solution will be achieved as association errors remain small. Experimental results show that environments of around $100 \text{ m} \times 100 \text{ m}$, including large loops, can be mapped in real time. Simulations also verify the robustness and accuracy of the method for larger loops, and gives insight into its proprieties. In future work, we will investigate the applicability of this method to larger environments and its extension by adding higher levels to the hierarchy.

REFERENCES

[1] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An atlas framework for scalable mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, R.O.C., 2003, pp. 1899–1906.

[2] J. E. Guivant and E. M. Nebot, "Solving computational and memory requirements of feature-based simultaneous localization and mapping algorithm," *IEEE Trans. Robot. Autom.*, vol. 19, no. 4, pp. 749–755, Aug. 2003.

[3] J. Knight, A. Davison, and I. Reid, "Towards constant time SLAM using postponement," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Maui, HI, 2001, pp. 406–412.

[4] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map building algorithm for real time implementation," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 242–257, Jun. 2001.

[5] S. J. Julier and J. K. Uhlmann, "Building a million beacon map," *Proc. SPIE*, vol. 4571, pp. 10–21, 2001.

[6] Y. Liu and S. Thrun, "Results for outdoor-SLAM using sparse extended information filters," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, R.O.C., 2003, pp. 1227–1233.

[7] J. A. Castellanos, J. Neira, and J. D. Tardós, "Limits to the consistency of EKF-based SLAM," in *Proc. 5th IFAC Symp. Intell. Auton. Veh.*, Lisbon, Portugal, 2004, Paper WA-1-2.

[8] J. D. Tardós, J. Neira, P. Newman, and J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *Int. J. Robot. Res.*, vol. 21, no. 4, pp. 311–330, 2002.

[9] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte, "An efficient approach to the simultaneous localization and mapping problem," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, Washington, DC, 2002, pp. 406–411.

[10] S. B. Williams, "Efficient solutions to autonomous mapping and navigation problems," Ph.D. dissertation, Australian Centre for Field Robotics, Univ. Sydney, Sydney, Australia, 2002.

[11] P. Newman, J. Leonard, and R. Rikoski, "Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar," in *Proc. 11th Int. Symp. Robot. Res.*, Sienna, Italy, 2003.

[12] T. Bailey, "Mobile robot localization and mapping in extensive outdoor environments," Ph.D. dissertation, Australian Centre for Field Robotics, Univ. Sydney, Sydney, Australia, 2002.

[13] J. A. Castellanos and J. D. Tardós, *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Boston, MA: Kluwer, 1999.

[14] J. Neira, J. D. Tardós, and J. A. Castellanos, "Linear time vehicle relocation in SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, R.O.C., Sep. 2003, pp. 427–433.

[15] S. Thrun, W. Burgard, and D. Fox, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Mach. Learn.*, vol. 31, no. 1–3, pp. 29–53, 1998.

[16] J. S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. IEEE Int. Symp. Computat. Intell. Robot. Autom.*, 1999, pp. 318–325.

[17] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robots*, vol. 4, pp. 333–349, 1997.

[18] S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker, "A system for volumetric robotic mapping of abandoned mines," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, R.O.C., 2003, pp. 4270–4275.

[19] M. Deans and M. Hebert, "Experimental comparison of techniques for localization and mapping using a bearing-only sensor," in *Proc. Int. Symp. Experimental Robot.*, vol. 271, Lecture Notes in Control and Information Science, S. S. D. Rus, Ed., Waikiki, HI, 2000, pp. 395–404.

[20] P. Newman and J. Leonard, "Pure range-only sub-sea SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, R.O.C., Sep. 2003, pp. 1921–1926.

[21] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: Wiley, 2001.

[22] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 890–897, Dec. 2001.

[23] 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. 948–953, Oct. 1999.

[24] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – A modern synthesis," in *Vision Algorithms: Theory and Practice*, W. Triggs, A. Zisserman, and R. Szeliski, Eds.   Berlin, Germany: Springer-Verlag, 2000, LNCS, pp. 298–375.

[25] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 56–68, 1986.

[26] J. Nocedal and S. J. Wright, *Numerical Optimization*.   New York: Springer-Verlag, 1999.

José Neira was born in Bogotá, Colombia, in 1963. He received the M.S. degree from the Universidad de los Andes, Bogotá, Colombia, in 1986, and the Ph.D. degree from the University of Zaragoza, Zaragoza, Spain, in 1993, both in computer science.

He is an Associate Professor with the Department of Computer Science and Systems Engineering, University of Zaragoza, where he teaches courses in compiler theory, computer vision, and mobile robotics. His current research interests include autonomous robots, data association, and environment modeling.

Carlos Estrada was born in Ferrol, Spain, in 1970. He received the M.S. and Ph.D. degrees in industrial-electrical engineering from the University of Zaragoza, Zaragoza, Spain, in 1995 and 2001, respectively.

He is a Lecturer with the Department of Computer Science and Systems Engineering, University of Zaragoza, where he teaches courses in automatic control theory, PLCs, and mobile robotics. His current research interests include data association and mobile robotics.

Juan D. Tardós was born in Huesca, Spain, in 1961. He received the M.S. and Ph.D. degrees in industrial-electrical engineering from the University of Zaragoza, Zaragoza, Spain, in 1985 and 1991, respectively.

He is an Associate Professor with the Department of Computer Science and Systems Engineering, University of Zaragoza, where he teaches courses in real-time systems, computer vision, and artificial intelligence. His current research interests include perception and mobile robotics.