Independent Local Mapping for Large-Scale SLAM

Josep Aulinas* Xavier Lladó* Yvan Petillot† Joaquim Salvi*
*Computer Vision and Robotics group (ViCoRob), University of Girona, Girona, Spain
†Ocean Systems Lab, Heriot Watt University, Edinburgh, United Kingdom

Abstract— **SLAM algorithms do not perform consistent maps for large areas mainly due to the uncertainties that become prohibitive when the scenario becomes larger and to the increase of computational cost. The use of local maps has been demonstrated to be well suited for mapping large environments, reducing computational cost and improving map consistency. This paper proposes a technique based on using independent local maps. Every time a loop is detected, these local maps are corrected using the information from local maps that overlap with them. Meanwhile a global stochastic map is kept through loop detection and minimization as it is done in the classical Hierarchical SLAM approach. This global level contains the relative transformations between local maps, which are updated once a new loop is detected. In addition, the information within the local maps is also corrected, maintaining always each local map separately. This approach requires robust data association algorithms, for instance, an adapted version of the JCBB algorithm. Experimental results show that our approach is able to obtain large maps areas with high accuracy.**

Index Terms— **SLAM, submap, large scale, data association**

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) also known as Concurrent Mapping and Localization (CML) is one of the fundamental challenges of robotics. The goal of SLAM is to build a map of an unknown environment while simultaneously determining the location of the robot within this map [1]. In this scenario, the vehicle has a known kinematic model and it is moving through an unknown environment, which is populated with artificial or natural landmarks measured by the on-board sensors of the vehicle. The SLAM problem involves finding appropriate representation for both the observation and the motion models, which is generally performed by computing its prior and posterior distributions using probabilistic algorithms, for instance Kalman Filters (KF), Particle Filters (PF) [2] and Expectation Maximization (EM) [3]. The main reason for the increasing popularity of these probabilistic techniques is the fact that robot mapping is characterized by uncertainty and sensor noise. Therefore, probabilistic algorithms tackle the problem by modeling explicitly different sources of noise and their effects on the measurements [4].

The most known SLAM approach is the Extended Kalman Filter SLAM (EKF-SLAM) [5]. It is based on representing the vehicle's pose and the location of a set of environment features in a joint state vector estimated and updated by the Extended Kalman Filter. The EKF provides a suboptimal solution due to the approximations introduced when linearizing the models, which may result in inconsistencies [6], and also due to the assumption that the uncertainties associated to the motion and measurement processes are only additional white Gaussian noise. In addition, one of the main drawbacks of the EKF implementation is the fact that for long duration missions, the number of landmarks increases and, eventually, computer resources will not suffice to update the map in real-time. This scaling problem arises because each landmark is correlated to all other landmarks, giving a memory complexity of $O(n^2)$ and a time complexity of $O(n^2)$ per step, where n is the total number of features stored in the map. The correlation appears since the observation of a new landmark is obtained with a sensor mounted on the moving vehicle and thus the landmark location is correlated with the vehicle location and the other landmarks of the map. This correlation is a key point for the long-term convergence of the algorithm, and needs to be maintained during all the mission.

Using submaps both limitations can be addressed at the same time (i.e. linearization errors and rise on computational cost). Therefore, using small maps to later on build a bigger one improves the consistency of the EKF-SLAM [6]. Limiting the size of a submap, by bounding the total number of landmarks or by fixing the maximum distance traveled by a vehicle, maintains the uncertainties of the submap and the linearization errors small. Furthermore, having small uncertainty matrices improve the consistency of the data association methods. For instance, in the Joint Compatibility Branch and Bound (JCBB) algorithm, the smaller the covariance matrix values are, the better the performance is [7]. Another advantage of working with small maps is that the amount of data involved in the EKF-SLAM is kept small, thus computational cost is reduced.

The main contribution of this paper is a novel technique that uses submaps as in a Hierarchical SLAM [8], but keeps local maps independently. The general idea of our approach is as follows. The vehicle navigates and builds local maps. Meanwhile, a global stochastic map containing the relative transformations between submaps is built. Once a loop is closed, these relative transformations are corrected by using loop closure constraints, as it is done in the Hierarchical SLAM approach. At this point, those maps that compose the loop are also updated, but kept separately. Therefore, map joining is not performed. The vehicle continues its navigation building local maps until a new loop is closed. To find a loop closing situation, the information from the corrected local maps is used. In addition, the data association process is simplified since the feature's information was previously corrected and the uncertainty was reduced.

This new technique has been implemented, tested with simulated scenarios, and compared with other EKF based techniques. The experiments show a reduction of the effects of the linearization error and also a more precise reconstruction of the map since the drift suffered in shorter distances is smaller, and the data association can be more robustly solved.

This paper is organized as follows. Related work is briefly described in Section II, especially those approaches that deal with large scale scenarios. Section III concerns the proposed solution, describing the steps involved in the process. Section IV summarizes the experimental results obtained. The paper ends with the conclusions and future work in Section V.

II. RELATED WORK

In the last decades, several works have tackled the issues associated with SLAM in large areas, such as the computational complexity and the inconsistencies caused by the linearization errors. Regarding the computational complexity some techniques delay the global update stage after several observations, reducing significantly the cost. For instance, the Compressed Extended Kalman Filter algorithm (CEKF) [9] or the Postponement approach [10], in which the global map update cost is $O(n^2)$.

On the other hand, the problem of map consistency produced by linearization errors motivated algorithms such as the Unscented Kalman Filter (UKF) [11], which achieves better consistency addressing the approximation issues of the EKF, but increases the computational complexity.

Another set of approaches has been based on the use of the Information Filter. An efficient example of this group is the Treemap algorithm [12]. It requires $O(\log\ n)$ time per step to recover a part of the state and $O(n)$ to recover the whole map. However, these techniques based on information filters suffer from the difficulty to perform data association since no covaraince matrix is involved.

More recent techniques based on submapping face the problems of consistency and computational complexity, but they require complete independency between maps, or what is the same, no information is shared between neighboring maps. For instance, the Decoupled Stochastic Map [13] works with submaps but it has difficulties due to the fact that its absolute submaps are not statistically independent and it requires approximations to solve these dependencies, introducing inconsitency in the map. The Constrained Local Submap Filter (CLSF) [14] or Local Map Joining (LMJ) [15] produce efficient global maps by consistently combining completely independent local maps, with a total cost of $O(n^2)$. The Divide and Conquer SLAM (DC) [18] is capable to recover the global map in approximately $O(n)$ time. The main limitation of the DC approach is that the time increases considerably with the overlapping information between local maps. More efficient techniques, such as the Constant Time SLAM (CTS) [16], the Atlas approach [17], and the Hierarchical SLAM [8], store the link between local maps by means of an adjacency graph. The CTS and the Atlas do not impose loop consistency in the graph, obtaining a suboptimal global map. Instead, the most precise path along the graph is computed to find the location of local maps in a global reference frame. The Hierarchical SLAM approach can perform consistent global maps by imposing loop constraints. However, the common information shared by different maps is discarded, or only used when joining maps. This way only the joint map information is kept, instead of that from the individual maps. The method we propose in this paper takes advantage of the adjacency graph representation from the CTS and Atlas, but it uses the loop constraint optimization as in the Hierarchical SLAM. In addition, the independent local maps are kept at a small size, while being corrected with the local information from overlapping submaps.

III. PROPOSED SOLUTION

The approach presented in this paper is similar to that of the Hierarchical SLAM. Once a loop is detected, the information shared by more than one submap is used to correct each individual local map separately. The assumption that enough overlapping exists between those maps that close the loop is made. Taking advantage of the overlapping, features that are common in several submaps are updated. These common landmarks from other submaps are first referred with respect to the last position of the local map to be updated. Afterwards, they are considered as new observations, while the existing ones are used to perform a EKF prediction. Using the predicted features and the observed ones, the innovation vector is estimated and an EKF update is computed. Therefore, the location of the landmarks within the local map is corrected. This correction is conducted for each of the maps with common features. Each local map is kept as an independent local map, allowing to use its local information in further loops. With this approach, the information from various submaps is used when correcting local maps, and the detection of loops takes profit of the higher precision of the corrected local maps. A diagram of the complete process is shown in Fig. 1.

A. Local Map Building

A local map is built running a basic EKF SLAM algorithm [15]. Every local map is initialized with its state vector x_k at zero, and its associated uncertainty $P_k = \mathbf{0}$. The local map building process is stopped when the number of landmarks in the current local map exceeds a threshold, when the distance visited within this local map is long enough according to a threshold, or when the uncertainties become higher than a certain acceptance limit. The output of this algorithm is the local map $M_{F_i}^{B_i}$ (1) with respect to its base reference B_i, which contains the position of the vehicle and the location of the n features F_i observed within this map, and its associated unceratinty $P_{F_i}^{B_i}$.

$$M_{F_i}^{B_i} = (x_{F_i}^{B_i}, P_{F_i}^{B_i}) : F_i = \{B_i, F_{i1}, F_{i2}, ..., F_{in}\} \quad (1)$$

B. Global Stochastic Map Building

The relative transformation $x_{ij}= x_{B_j}^{B_i} = (x_{ij}, y_{ij}, \theta_{ij})^t$ between two consecutive submaps M_i and M_j is stored in a relative stochastic map $\chi_u = (\widehat{x}_u, \widehat{P}_u)$ (similar to the Hierarchical SLAM approach [8]), where the $\widehat{x}_u$ is a vector containing all relative transformations between local maps, and $\widehat{P}_u$ the uncertainties associated to these transformations. The subindex u stands for unconstrained, since at this point no

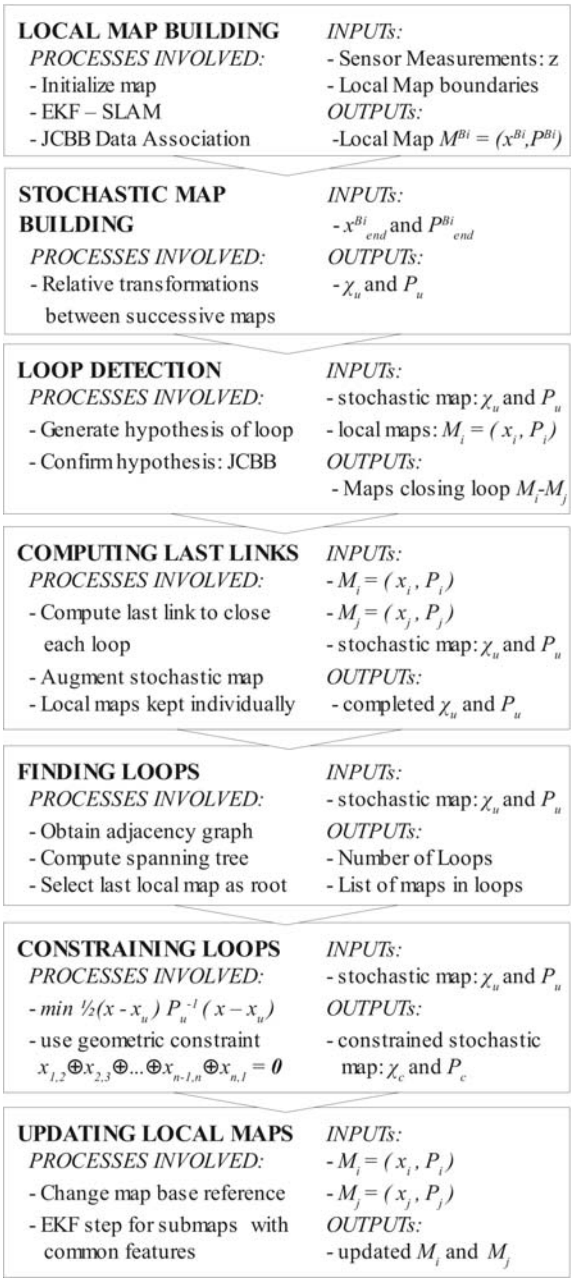

Fig. 1. Block diagram detailing the Hierarchical SLAM process adapted to the approach presented in this paper.

loop closing constraints have been applied. This will be done in further steps when imposing loop constraints (*Section III.D*).

C. Loop Detection

When the vehicle is moving through a scenario, a sequence of local maps is built. Every time a local map M_j is completed, the possibility of closing a loop with some other previous local maps M_i, where $i = 1...j$, is checked. The data from other local maps can be referred to the base reference of the last map B_j using the relative transformations between maps $x_{B_i}^{B_j}$, as done in [15]. Loop hypothesis are created for those submaps that are around the last local map M_j, considering its uncertainties. These hypotheses are then confirmed by means

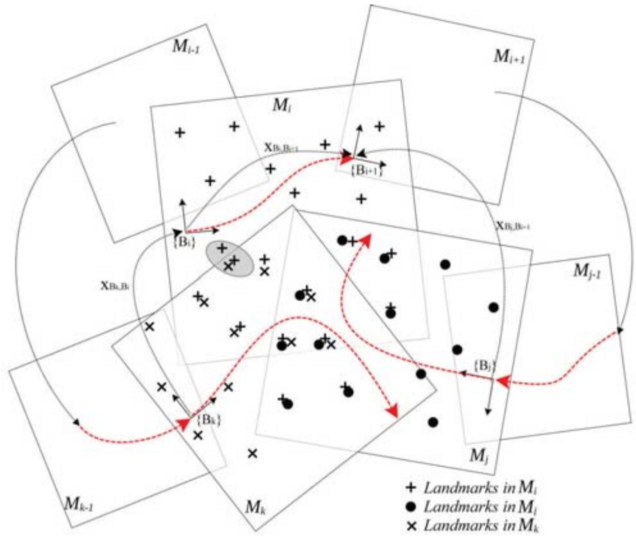

Fig. 2. Schematic representation of a loop closing situation, where three maps M_i, M_j and M_k are overlapping and closing two different loops. The grey ellipse contains three landmarks that may cause confusion.

of the data association algorithm JCBB, which determines if any landmark have been revisited. In addition, the result of the JCBB is verified by checking that only one feature from a local map M_i is associated to a single feature from a local map M_j. This extra check is required, since features closed by the boundaries of the local maps may cause confusion, as depicted in Fig. 2. If a feature is associated to more than one landmark from the other map, its neighboring information is then taken into account (as well as its joint mahalanobis distance). This way, the right associate is selected and the others are discarded. After this check, the loop hypothesis is accepted and therefore the loop constraints are imposed (*Section III.D*). It may happen, and it is desirable, that more than one submap overlap with the last submap. In this way, more than one hypothesis of loop is accepted. Assuming that we can have high overlapping between consecutive submaps, the detection of loops is not run immediately after closing a loop. The system waits until two new local maps have been build, in order to avoid loop detection at the end of every submap.

D. Computing last links to close loops

The global stochastic map χ_u does not contain the last link that closes the loop $x_{B_i}^{B_j}$. Hence, this last link has to be computed by first solving robustly the data association between the local map M_{i-1} and the last map M_j. A possible scenario where this step is required is shown in Fig. 2. The data from one map, for instance M_i is moved and referred with respect to the base B_j of M_j. Therefore the links from B_j to the last position of the vehicle within M_i is found. This is equivalent to the relative transformation $x_{B_i}^{B_j}$ from the base reference of the last local map B_j to the local map B_i. This last path to close loops is computed for each new hypothesis of loop closing.

E. Constraining the Loop

The approach to impose loop constraints is fully detailed in the Hierarchical SLAM algorithm [8]. The main idea is to use a geometric constraint: the composition of all those relative transformations between submaps that composes the loop should be equal to zero (2).

$$h(x) \equiv x_{B_{i+1}}^{B_i} \oplus x_{B_{i+2}}^{B_{i+1}} \oplus ... \oplus x_{B_j}^{B_{j-1}} \oplus x_{B_i}^{B_j} = 0 \qquad (2)$$

Once all links required to close the loops are found, the constraint $h(x) = 0$ can be applied by means of a constrained optimization problem (3).

$$min f(x) = min \frac{1}{2} (x - \widehat{x}_u)^t P_u^{-1} (x - \widehat{x}_u) \qquad (3)$$

This optimization is solved here with an adapted Sequential Quadratic programming method [8]. Its output is the constrained stochastic map χ_c and its associated covariance matrix P_c. The optimization works better when facing various loops, because the higher the number of loops is, the higher the number of geometric constraints will be. Thus, the process performs more accurately, but with higher time consumption.

In order to avoid using redundant loops and optimize the computational cost, it is necessary to select the loops that are used when imposing constraints. For this purpose, fundamental graph theory is applied. In particular an adapted spanning tree is built to define the elemental cycles from the adjacency graph belonging to the global stochastic map. This step is equivalent to a Breath First Search (BFS) algorithm, but adapted to the necessities of the presented system. This approach provides the minimum possible number of loops that are then used in the minimization.

F. Updating Local Maps

After constraining loops, those local maps involved in the loop closing, i.e. sharing common features, are corrected. The conceptual idea of this situation is shown in Fig. 2 and the algorithm is given in Table I. In this example three local maps are overlapping, M_i, M_j and M_k. Each local map is defined by its associated state vector and covariance matrix, (x_i, P_i), (x_j, P_j), (x_k, P_k), with respect to its local base reference B_i, B_j and B_k. These state vectors contain the position of the vehicle and the location of all landmarks visited within the local map. Some of these features from the three maps are overlapping, and they are associated running the JCBB algorithm. Once, these common landmarks have been associated, the local update procedure is computed. The basis of this procedure is to consider those common landmarks from one map, for instance from M_i, as the predictions of an EKF prediction stage, while the common features from the rest of overlapping maps are understood as the new mesurements. From these predictions and obsnew measurementes the innovation vector and the EKF gain are calculated and an EKF update is performed (See Table I). Finally, the corrected maps are referred back to its original bases B_i and B_j. This routine has to be executed for each overlapping local map, for instance, in the example on Fig. 2, M_i is first corrected using the information from M_j and the result is then further updated using the data from M_k. This double correction leads to overconfidence, which means that the vehicle could end up completely lost. In order to solve this overconfidence issue, landmark measurment process noise is added to the this update routine.

TABLE I
LOCAL MAP UPDATE ALGORITHM

Local Map update($M_n^{B_n} = (x_n^{B_n}, P_n^{B_n})$, $n = i, j, k, ...$)
for All maps **do**
    $M_p^{B_p}$ = Input map (as a *p*rediction)
    $Queue = [M_i, M_j, M_k, ...]$
    Delete M_p from Queue
    for maps in Queue **do**
        $M_o^{B_o}$ = Map from Queue (as a new *o*bservation)
        $L_{M_o = M_p}$ = Data Association(M_p, M_o)
        L_c = Common landmark with lower uncertainty
        $M_p^{B_p}$, $M_o^{B_o}$ w.r.t. $L_c \Rightarrow M_o^{L_c}$, $M_p^{L_c}$
        EKFupdate($L_{M_o = M_p}$, $M_o^{L_c}$, $M_p^{L_c}$)
        $M_p^{L_c}$, $M_o^{L_c}$ w.r.t. B_p and $B_o \Rightarrow M_p^{B_p}$, $M_o^{B_o}$
    endfor
endfor

NOTEs: Data Association solved using JCBB algorithm
w.r.t. stands for *with respect to*

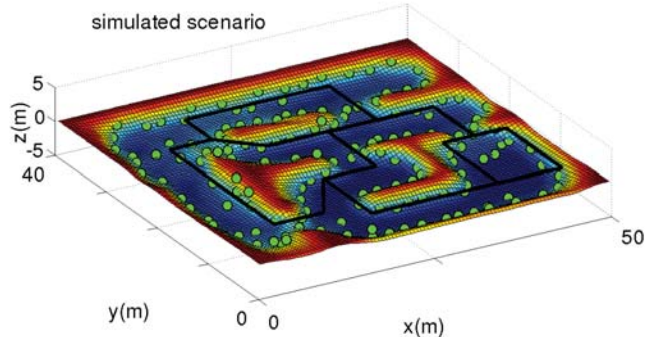

Fig. 3. Synthetic scenario populated with artificial landmarks (green circles). The trajectory ground truth is plotted in balck.

IV. EXPERIMENTAL RESULTS

The performance of the method presented in this paper was tested in a simulated scenario emulating a real environment of 50 x 40 meters, with loops from 40 to 100 meters long, and with 150 landmarks (see Fig. 3). The results presented here are the mean outputs of 100 different simulations varying the noise level and the number of landmarks per submap (implying a change in the map size and number of loops). All the simulations were conducted on a Pentium Core 2 Duo 2.66-GHz. The purpose of our experiments was twofold: 1) to evaluate the consistency of our approach; 2) to analyse the computational cost.

The experiments demonstrated that our approach performs consistently. In particular, the accuracy of the local maps improved compared to the classic Hierarchical SLAM approach.

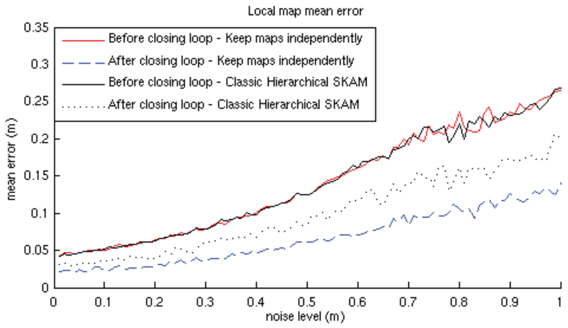

Fig. 4. Mean error trend for increasing levels of noise.

TABLE II
COMPUTATIONAL TIME COMPARISON (IN SECONDS) FOR 100 TESTS.

method	H-SLAM approach mean	std	total	Proposed Solution mean	std	total
Loop detection	0.12	0.001	8.48	0.05	0.001	2.42
Constraining loops	0.23	0.058	17.75	0.45	0.069	20.82

These results are shown in Fig. 4, where the mean error before and after constraining the loops from all the 100 simulations with different levels of noise is plotted. This consistency is also illustrated in Fig. 5, where we show the vehicle and landmarks location and uncertainty from a sequence of submaps.

Concerning the global level, the results were also positive. Fig. 6 presents the stochastic map containing the base reference of local maps (as nodes), and the links between these bases (as arcs). Both the unconstrained and the constrained values are shown after closing a few loops, demonstrating that the global optimization stage properly corrects the location of the local maps' base references.

Regarding to the computational cost, the experiments demonstrated that the computational time is slightly improved compared to the Hierarchical SLAM (see Table II). The mean time for running a single simulation was 28.93 seconds when using the classic Hierarchical SLAM approach and 25.25 seconds with our approach. The main differences are:

1) The Classic Hierarchical SLAM approach consumes more time than our approach, because it requires more time when checking loops. This is due to the fact that local maps' size increases after joining maps, therefore the data association becomes more costly.
2) Our approach is more time demanding when constraining loops due to the fact that keeping local maps independently produces a higher number of loops to be constrained.

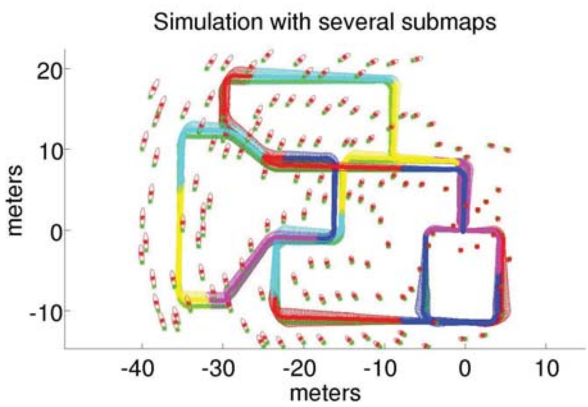

Fig. 5. Vehicle's and landmarks' estimates and its uncertainties. Every time a loop is constrained the uncertainties decrease, and the location of the landmarks and the vehicle are corrected.

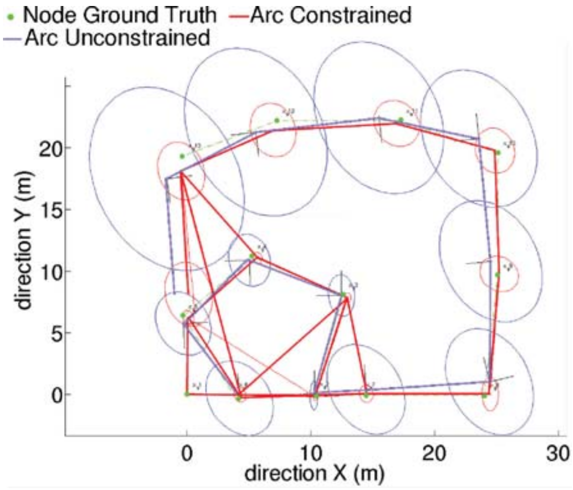

Fig. 6. Global map represented as an adjacency graph after several loops.

Finally, our algorithms was tested using the Victoria Park dataset recorded by Eduardo Nebot et al. [19] at the Australian Centre for Field Robotics. This data set describes a path through an area of around 197m x 93m. This sequence consists of 7247 frames along a trajectory of 4 kilometer length, recorded over a time frame of 26 minutes. The data set contains sensor readings from steering and rear-axis wheel (odometry) and laser range finder (one 360 degrees scan per second) along with the ground truth position data from GPS. For the laser range data a tree detector function is provided together with the dataset. These detected trees are used as feature landmarks. They usually have a large distance to each other and can be separated or uniquely identified with common data association techniques.

The results obtained after running our approach on the Victoria Park dataset show promising results, as presented in Fig. 7. This figure presents the trajectory of the vehicle according to our technique, which is considerably closed to the GPS trajectory. The final map is also plotted (i.e. the location of the landmarks and its uncertainties). This information is plotted on a recent satellite picture of the real park. Even if some of the trees are not there any more, most of the ones that still appear in the picture coincide with the ones estimated with our approach.

V. DISCUSSION AND FUTURE WORK

The approach presented in this paper is suitable to map large scale scenarios. The two main differences with respect to other methods that use submaps are:

1) Local maps are kept independent during the whole scenario. Keeping local maps independently means that their size will not increase. This is shown to be positive

Fig. 7. Victoria park satellite image with the corrected vehicle trajectory and its uncertainty (in blue), the landmarks' estimated position (in yellow) and its uncertainties (in red).

in terms of computational cost, i.e. time consumption. The total time required to build a local map is kept short and the total time required to navigate the whole scenario is also shorter. This decrease on computational demand is explained by the fact that covariance matrices are kept small and so it is the time required to build each submap. Another reason is the fact that the data association is less costly since less landmarks are involved in the process. On the contrary, the necessity to check this data association with more than one submap and also to correct local information introduces extra cost.

2) Local maps are updated every time a loop is closed. Although this process adds extra cost to the whole algorithm, it introduces two significant advantages: the data association algorithm performs more efficiently and the mean local map error is improved. The data association is more efficient since the uncertainties of the landmarks within local maps are smaller after being corrected. Therefore the confusion between very closed landmarks is simplified and also the time consumption is reduced due to the fact that less iterations are required. Concerning the error commited when building local maps, thanks to the local update stage this error is considerably improved. In addition to linearization errors, further approximations are introduced when neglecting the correlation between local maps. However, the results suggest that these approximations are not causing the solution to diverge, mainly because submaps are kept small, thus linearization error effects are also bounded.

As future work we plan to optimize the loop selection stage and to simplify the local map update by performing it in a single EKF step regardless of the number of submaps involved.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of the authors of the Hierarchical SLAM approach [8] and the data association algorithm JCBB [7], Carlos Estrada, José Neira and Juan Domingo Tardós from the University of Zaragoza, for the fruitful discussions maintained.

The authors acknowledge the support of the project *Computer vision systems for submarine cartography and aquaculture* founded by the Spanish Ministry of Science and Innovation (Reference: DPI-2007-66796-C03-02).

REFERENCES

[1] H. Durrant-Whyte and T. Bailey. "Simultaneous localization and mapping (SLAM): Part I". *IEEE Robotics and Automation Magazine*, vol. 13, num. 2, pp 99–108, 2006.

[2] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges", *18th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, Acapulco Mexico, pp. 1151–1156, 2003.

[3] W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun, "Sonar-based mapping with mobile robots using EM" *Proc. of the 16th Int. Conf. on Machine Learning*, 1999.

[4] S. Thrun, D. Burgard and W. Fox. "Probabilistic Robotics". MIT Press, 2005. ISBN-10: 0-262-20162-3.

[5] R. Smith, M. Self, and P. Cheeseman, "A Stochastic map for uncertain spatial relationships", *Robotics Research, the Fourth Int. Symposium*, O. Faugeras and G.Giralt (Editors), the MIT Press, pp. 467–474, 1988.

[6] J. A. Castellanos, R. Martínez-Cantón, J. Neira, and J. D. Tardós, "Robocentric map joining: Improving the consistency of EKF-SLAM", *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 21–29, 2007.

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

[8] C. Estrada, J. Neira, and J. D. Tardós, "Hierarchical SLAM: real-time accurate mapping of large environments", *IEEE Trans. Robotics*, vol. 21, no. 4, pp. 588–596, 2005.

[9] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation", *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 242–257, 2001.

[10] J. Knight, A. Davison, and I. Reid, "Towards constant time SLAM using postponement", *Proc. IEEE/RJS Int. Conf. on Intelligent Robots and Systems*, pp. 406–412, 2001.

[11] E. Wan and R. van der Merwe, "Kalman Filtering and Neural Networks, Chapter 7: The Unscented Kalman Filter". Wiley, 2001. ISBN: 978-0-471-36998-1.

[12] U. Frese, "Treemap: An $O(\log n)$ algorithm for simultaneous localization and mapping", *Autonomous Robots*, vol. 21, no. 2, pp. 103–122, 2006

[13] J. J. Leonard and H. J. S. Feder, "A computationally efficient method for large-scale concurrent mapping and localization", *Robotics Research: The Ninth Int. Symposium*, D. Koditschek and J. Hollerbach (Editors), Springer Verlag, pp. 169–176, 2000.

[14] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte, "An efficient approach to the simultaneous localisation and mapping problem", *IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 406–411, 2002.

[15] J. D. Tardós, J. Neira, P. M. Newman, and J. J. Leonard, "Robust mapping and localization in indoor environments using sonar data", *Int. Journal on Robotics Research*, vol. 21, no. 4, pp. 311–330, 2002.

[16] P.M. Newman and J.J. Leonard, "Consistent, convergent, and constant-time SLAM", *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pp. 1143–1150, 2003.

[17] M. Bosse, P. M. Newman, J. J. Leonard, and S. Teller, "SLAM in large scale cyclic environments using the atlas framework", *Int. Journal in Robotics Research*, vol. 23, no. 12, pp. 1113–1139, 2004.

[18] L. M. Paz, J. D. Tardós, and J. Neira, "Divide and conquer: EKF SLAM in O(n)", *IEEE Trans. in Robotics*, vol. 24, no. 5, ISSN 1552–3098, 2008.

[19] E. Nebot. "Victoria park data set". Retrieved July 3, 2009. Web site: *http://www-personal.acfr.usyd.edu.au/nebot/dataset.htm*