SLAM based Selective Submap Joining for the Victoria Park Dataset ⋆

Josep Aulinas * **Xavier Lladó** * **Joaquim Salvi** *
Yvan R. Petillot **

* *Computer Vision and Robotics group (ViCoRob), University of Girona, Girona 17071, Spain (e-mail: {jaulinas,qsalvi,llado}@eia.udg.edu.)*
** *Ocean Systems Lab, Heriot Watt University, Edinburgh EH14 4AS, United Kingdom, e-mail: (Y.R.Petillot@hw.ac.uk)*

Abstract: One of the main drawbacks of current SLAM algorithms is that they do not result in consistent maps of large areas, mainly because the uncertainties increase with the scenario. In addition, as the map size grows the computational costs increase, making SLAM solutions unsuitable for on-line applications. The use of local maps has been demonstrated to be useful in these situations, reducing computational cost and improving map consistency. Following this idea, this paper proposes a technique based on using independent local maps together with a global stochastic map. The 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. Thus, maps sharing a high number of features are updated through fusion and the correlation between landmarks and vehicle is maintained. Results on synthetic data and on the Victoria Park Dataset show that our approach is able to consistently map large areas and the computational costs are lower.

Keywords: Localization, mapping, Extended Kalman Filter, data association, mobile robots.

1. INTRODUCTION

Simultaneous Localization and Mapping (SLAM), also known as Concurrent Mapping and Localization (CML), is a fundamental challenge of robotics. An autonomous vehicle must be able to localize itself in either known or unknown environments. When the environment is known, the problem is only a localization problem. But in many other occasions, the environment is completely unknown. Therefore, the vehicle needs to build a map, and also needs to localize itself inside this map. The idea of solving both problems at the same time originated the SLAM term. SLAM algorithms aim to accomplish these two goals at the same time, building a map of an unknown environment and determining the position of the vehicle within this map, see Durrant-Whyte and Bailey (2006). This unknown environment might be structured or unstructured, populated with artificial or natural salient features. In order to solve this problem, a known kinematic model of the vehicle is defined. The vehicle moves through the unknown environment. On-board sensors measure these salient features, which are then used as landmarks for the SLAM algorithms. One of the main characteristics on the field of robot mapping and localization is the uncertainty caused by sensor noises, see Thrun et al. (2005). For this reason, finding appropriate representation for both, the observation and the motion models, is of vital importance.

⋆ The authors acknowledge the support of the research project funded by the Spanish Ministry of Science and Innovation (DPI-2007-66796-C03-02). J. Aulinas holds a University of Girona scholarship (BR-07/03).

Probabilistic techniques are able to model explicitly different sources of noise and their effects on the measurements. For this reason, probabilistic algorithms, such as Kalman Filters (KF), Particle Filters (PF), Montemerlo et al. (2003) and Expectation Maximization (EM) Burgard et al. (1999) are very popular in the SLAM context.

A well known and widely used SLAM approach is the Extended Kalman Filter SLAM (EKF-SLAM), see Smith et al. (1988). EKF-SLAM represents the vehicle's pose and the location of a set of environment features in a joint state vector. This vector is estimated and updated by the EKF. EKF relies on the assumption that the state transition and the measurement functions are linear with added Gaussian noise, and the initial posteriors are also Gaussian. Unfortunately, real world is non-linear. EKF accommodates the non-linearities from the real world by approximating the robot motion model using linear functions. Therefore, the EKF solution is only suboptimal due to the approximations introduced when linearising the models. These approximations together with the assumption that the uncertainties associated to the motion and measurement models are only additive white Gaussian noise may result in inconsistencies, see Castellanos et al. (2007). In fact, this is one of the main drawbacks of EKF for long missions. A second but not least important drawback, is the scaling increase on complexity that arises because each landmark is correlated to all the other landmarks. 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 current pose 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 the whole mission.

1.1 Contribution

In this paper we propose a technique based on the use of independent local maps together with a global level stochastic map. The main contribution of our approach is in the local map update strategy. This strategy determines whether the last local map built so far has to be fused with previous local maps or kept independent. The general idea of our approach is as follows. The vehicle navigates through the environment. Local maps are built, together with a global stochastic map containing the relative transformations between these local maps. The use of a global level allows to identify loop closure events. A loop closure event is given when the vehicle is revisiting an area, therefore a certain amount of map features are repeated in two local maps. The decision of whether to joint and fuse two maps that share features is the main novelty in our approach. This decision is made according to the amount of repeated features. Then, when this number of repeated features exceeds a certain value, the decision is to joint and fuse the two maps, becoming a single one. This decision is made on the basis that fusing two maps that share many landmarks will produce a better update than fusing two maps that only share few landmarks. This novelty has been implemented and tested on synthetic and real dataset. The experiments show a consistent reconstruction of the map, since the drift suffered and the linearisation errors are small in short distances.

The rest of the paper is organised as follows. A brief background on the state of the art on related algorithms is given in Section II. Section III presents the key novelty of our approach, together with its mathematical derivations. Section IV summarises the experimental results under synthetic and real data. Finally, the paper ends with conclusions and future work.

2. RELATED WORK

In the introduction, the two main drawbacks of an EKF based SLAM algorithm for large scale missions have been identified: 1) the computational complexity and 2) the inconsistencies caused by the linearisation errors. Both issues have been analysed by several researchers during the last years.

Regarding the computational complexity, Guivant and Nebot (2001) presented the Compressed Extended Kalman Filter algorithm (CEKF), while Knight et al. (2001) proposed the Postponement approach. Both approaches delay the global update stage after several observations, reducing significantly the cost without introducing any penalties in the accuracy of the results. On the other hand, map consistency problem has motivated the use of different filtering techniques, for instance the Unscented Kalman Filter (UKF), see Wan and Van Der Merwe (2001). The UKF achieves better consistency addressing the approximation issues of the EKF, but increasing the computational complexity.

The Information Filter (IF) and its variations compose another set of approaches. Some examples of this group are the Sparse Extended Information Filter (SEIF) by Thrun et al. (2004), the Exactly Sparse Extended Information Filter (ESEIF) by Walter et al. (2007) or the Exactly Sparse Delayed state Filter (ESDF) by Eustice et al. (2006). All these approaches reduce the computational cost taking advantage of the sparsity structure of the inverse of the covariance matrix (information matrix). However, these techniques suffer from the difficulty to perform data association since no covariance matrix is involved. Another efficient example following this strategy is the Treemap algorithm presented by Frese (2006), which requires $O(\log n)$ time per step to recover part of the state and $O(n)$ to recover the whole map.

More recent techniques address the consistency problem and computational complexity by dividing the map. These techniques use submaps with the only constraint of keeping complete independence between them. The Decoupled Stochastic Map (DSM) from Leonard and Feder (2000) uses submaps, but they are not statistically independent, and therefore, introduces inconsistencies in the map. A similar behaviour is observed in the Aulinas et al. (2009) work, where local maps are kept separately while sharing common information, which means that some correlations are approximated, thus leading to divergences.

The Constrained Local Submap Filter (CLSF) by Williams et al. (2002) or the Local Map Joining (MJS) by Tardós et al. (2002) produce efficient global maps by consistently combining completely independent local maps, with a total cost of $O(n^2)$. Paz et al. (2008) presented the Divide and Conquer SLAM (DCS), which is capable to recover the global map in approximately $O(n)$ time. More efficient techniques, such as the Constant Time SLAM (CTS) by Leonard and Newman (2003), the Atlas approach by Bosse et al. (2004), and the Hierarchical SLAM by Estrada et al. (2005), 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 performs consistent global maps by imposing loop constraints. However, the common information shared by different maps is discarded or only used when maps are joint. In this way only the joint map information is kept. Finally, Piniés and Tardós (2008) proposed the Conditionally Independent Local Maps (CILM), which is based on sharing information between consecutive submaps, this way a new local map is initialised considering the a priori knowledge.

All these submapping techniques demonstrate that using submaps both linearization errors and computational cost can be addressed at the same time, improving the consistency of EKF-SLAM, see Castellanos et al. (2007). Working with limited size submaps maintains the uncertainties of the submap and the linearisation errors small. Furthermore, having small uncertainty matrices improves 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 (see Neira and Tardós (2001)).

Algorithm 1: Extended Kalman Filter: (x_k, P_k, u_{k+1}, z_{k+1})

Prediction (estimate)

$\widehat{x}_{k+1} = f(x_k, u_{k+1})$

$\widehat{P}_{k+1} = F_k P_k F_k^T + G_k Q_k G_k^t$

$\widehat{z}_{k+1} = h(\widehat{x}_{k+1})$

Observation (innovation vector and matrix)

$\nu_{k+1} = z_{k+1} - \widehat{z}_{k+1}$

$S_{k+1} = H_k \widehat{P}_{k+1} H_k^T + R_k$

Update (correction, Kalman Gain)

$W_{k+1} = \widehat{P}_{k+1} H_k^T S_{k+1}^{-1}$

$x_{k+1} = \widehat{x}_{k+1} + W_{k+1}\nu_{k+1}$

$P_{k+1} = \widehat{P}_{k+1} - W_{k+1} S_{k+1} \widehat{P}_{k+1}$

Iterate with x_{k+1} *and* P_{k+1}

Notice that:

$F_k = f'(x_k, u_{k+1}) = \frac{\partial f(x_k, u_{k+1})}{\partial x_k}$

$G_k = f'(x_k, u_{k+1}) = \frac{\partial f(x_k, u_{k+1})}{\partial u_k}$

$H_k = h'(x_{k+1}) = \frac{\partial h(\widehat{x}_{k+1})}{\partial x_k}$

Another advantage of working with small maps is that the amount of data involved in the EKF-SLAM is kept small, reducing computational cost.

3. MATHEMATICAL FORMULATION

Our purpose is to work with a 3DOF vehicle, i.e. a 2D motion model. We assume that this vehicle carries several sensors, giving navigation data relative to the vehicle's reference frame, such as velocity and orientation. In addition, the vehicle must carry some sort of sensors capable of acquiring information from the scenario. The scenario is supposed unknown and unstructured, but populated with objects, trees, rocks and other detectable features. From the information given by these sensors the vehicle and map state estimates can be written in a joint state vector $\widehat{\mathbf{x}}$, as in (1). In addition, the state uncertainty is stored in a covariance matrix $\widehat{\mathbf{P}}$, and the procedure to filter the measurement noise is the well known EKF (see *Algorithm 1*), where f and h are the transition and the observation models from time k to $k+1$, and F and H are their linearised version.

$$\widehat{\mathbf{x}} = (x_v \ y_v \ \psi_v \ x_{l1} \ y_{l1} \ ... \ x_{ln} \ y_{ln})^t \qquad (1)$$

The main idea of our approach is to use the EKF to perform SLAM on a limited size areas of the whole scenario. Hence, a map is initialised with zeros $\mathbf{x}_0 = \mathbf{0}$ and $\mathbf{P}_0 = \mathbf{0}$ and for every discrete step time k the vehicle position and the map are estimated, new measures are done, and an update stage is conducted. This iterative process stops when the number of landmarks in the map exceeds a certain *threshold*, limiting the problem to small size matrices and vectors; or when the uncertainties involved exceed certain boundaries, because this would lead to overconfidence or to inconsistencies.

Once a map is finished, the last position of the vehicle gives the topological transformation between the beginning of

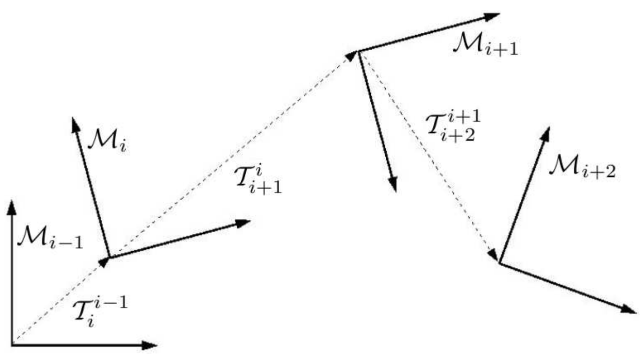

Fig. 1. Global level example, where $\mathcal{M}$ are the submaps, and $\mathcal{T}$ are the relative transformations from adjacent maps.

the map and its end, or what is the same, the relative transformation between the base reference of the current map $\mathcal{M}_i$ and the next one $\mathcal{M}_{i+1}$ (see Fig. 1). This next map is initialised with zeros and the EKF iterative process starts again. The link between the two maps ${}^{\mathcal{M}_i}\mathcal{T}_{\mathcal{M}_{i+1}}$ is stored in a global level $\mathbf{x}_G$, together with its corresponding uncertainty $\mathbf{x}_G$, as in (2).

$$\mathbf{x}_G = \begin{bmatrix} \cdot \\ {}^{\mathcal{M}_{i-1}}\mathcal{T}_{\mathcal{M}_i} \\ {}^{\mathcal{M}_i}\mathcal{T}_{\mathcal{M}_{i+1}} \\ \cdot \end{bmatrix} \ \mathbf{P}_G = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \sigma^2_{{}^{\mathcal{M}_{i-1}}\mathcal{T}_{\mathcal{M}_i}} & 0 & \cdot \\ \cdot & 0 & \sigma^2_{{}^{\mathcal{M}_i}\mathcal{T}_{\mathcal{M}_{i+1}}} & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad (2)$$

This global level can be understood as an adjacency graph, where the submaps are the nodes, and the relative transformations to go from one to another are the arcs. The global level is used to check the possibility of being in front of a loop closing event. A loop closure is accepted when the vehicle is revisiting a region. In order to know how big is the revisited region, the data association between those maps that are closing the loop is computed. The loop closing strategy involves a decision on whether to fuse maps, it is therefore a Selective Submap Joining SLAM (SSJS), see *Algorithm 2*. If the correspondences between maps are higher than a *threshold*, they are joint and fused to a single map, as it is done in the Map Joining algorithm (see Tardós et al. (2002)). Given two submaps $\mathcal{M}_i$ and $\mathcal{M}_j$ referred to a common base $\mathcal{B}$, they are first stored into a joint state vector, as in (3).

$$ {}^{\mathcal{B}}\mathbf{x}_{ij} = \begin{bmatrix} {}^{\mathcal{B}}\mathbf{x}_i \\ {}^{\mathcal{B}}\mathbf{x}_j \end{bmatrix} \ \mathbf{P}_{ij} = \begin{bmatrix} \mathbf{P}_i & \mathbf{P}_i\mathbf{J}_1^t \\ \mathbf{J}_1\mathbf{P}_i & \mathbf{J}_1\mathbf{P}_i\mathbf{J}_1^t + \mathbf{J}_2\mathbf{P}_j\mathbf{J}_2^t \end{bmatrix} \qquad (3)$$

The common landmarks from $\mathcal{M}_i$ are the predictions (as in a standard EKF) and the common landmarks from $\mathcal{M}_j$ are understood as new observations. Afterwards, the innovation vector and matrix are computed, followed by the EKF update stage. Finally, the rows and columns corresponding to common landmarks from $\mathcal{M}_j$ are removed from the joint state, to avoid repetition. Together with the map fusion, the corresponding link in the global level is corrected. This correction is obtained directly from the map fusion since

Algorithm 2: Selective Submap Joining SLAM

begin mission
while navigating **do**
  $\widehat{\mathbf{x}}_i, \widehat{\mathbf{P}}_i$ = EKF SLAM() $\leftarrow$ (*Build submap* $\mathcal{M}_i$)
  $\widehat{\mathbf{x}}_G, \widehat{\mathbf{P}}_G$ = build global map($\widehat{\mathbf{x}}_i$, $\widehat{\mathbf{P}}_i$)
  $\mathcal{H}_{Loop}$ = check possible loops($\widehat{\mathbf{x}}_G$, $\widehat{\mathbf{P}}_G$)
  for $j = \mathcal{H}_{Loop}$ **do**
    refer $\mathcal{M}_i$ and $\mathcal{M}_j$ to a common base reference
    $\mathcal{H}_{ij}$ = data association($\widehat{\mathbf{x}}_i, \widehat{\mathbf{x}}_j, \widehat{\mathbf{P}}_i, \widehat{\mathbf{P}}_j$)
    if $\mathcal{H}_{ij} > threshold$ **then**
      $\widehat{\mathbf{x}}_{ij}, \widehat{\mathbf{P}}_{ij}$ = map fusion($\widehat{\mathbf{x}}_i$, $\widehat{\mathbf{P}}_i$, $\widehat{\mathbf{x}}_j$, $\widehat{\mathbf{P}}_j$, $\mathcal{H}_{ij}$)
      $\widehat{\mathbf{x}}_G, \widehat{\mathbf{P}}_G$ = update global map($\widehat{\mathbf{x}}_{ij}, \widehat{\mathbf{P}}_{ij}$)
    endif
  endfor
endwhile

the links within the fused maps are correlated and updated with all the information. After deciding whether to fuse the maps, a new submap is built and the whole process is repeated again.

4. REAL EXPERIMENTS

The performance of our method is analysed in this section. In order to do so, the SSJS algorithm was tested using synthetic data. An example of this synthetic experiments is shown in Fig. 2. This figure represents a loop closing event, showing the performance of our selective fusion approach. These synthetic experiments allowed us to check: 1) the overall consistency of our method, and 2) the computational complexity improvement compared to a standard EKF.

The overall consistency of our method was checked via the statistical test, Normalized Estimation Error Squared (NEES) (4).

$$NEES = (\mathbf{x}_k - \widehat{\mathbf{x}}_k)^t \mathbf{P}_k^{-1} (\mathbf{x}_k - \widehat{\mathbf{x}}_k) < \chi^2_{r,1-\alpha} \tag{4}$$

where $r = dim(\mathbf{x}_k)$ is the degree of freedom, and α is the desired significance level (usually 0.05). Given the ground truth, the state $(\widehat{\mathbf{x}}, \mathbf{P})$ estimation is consistent when $NEES < \chi^2_{r,1-\alpha}$, otherwise the estimation is too optimistic, becoming inconsistent. Fig. 3 shows the SSJS performing inside the theoretical consistency boundaries.

Regarding the computational demand, Fig. 4 shows the improvement of our method with respect to a standard EKF. The EKF computational time increases on a quadratic order to the number of features in the map, while our approach increases almost lineally.

After obtaining positive results using synthetic data, real experiments were conducted using the Victoria Park dataset recorded by Nebot (2009) at the Australian Centre for Field Robotics, see Fig. 5. This dataset describes a path through an area of around 197m x 93m. This sequence consists of 7247 frames along a trajectory of 4 kilometres,

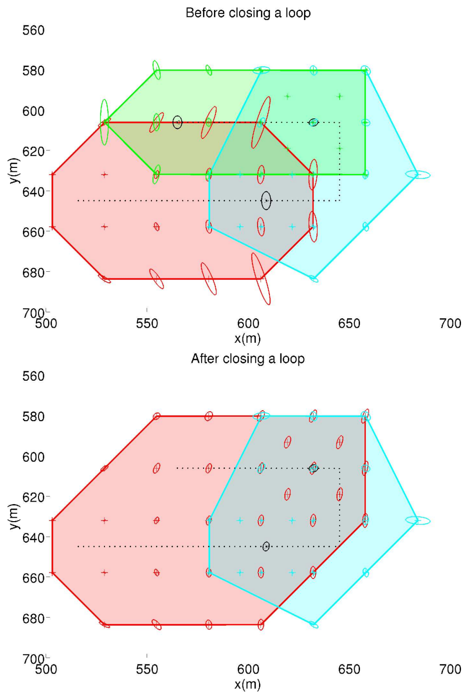

Fig. 2. Example of a map fusion step. In the top image, three local maps have been built. In the bottom, two maps have been fused after closing the loop.

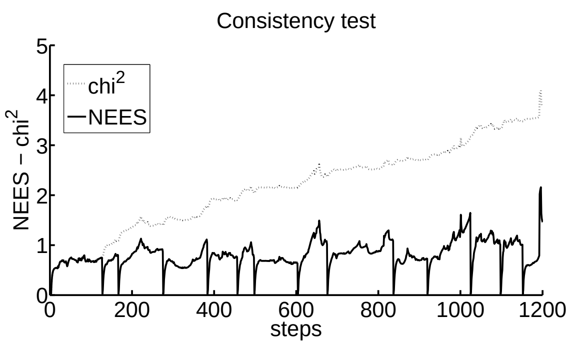

Fig. 3. Consistency test using the NEES.

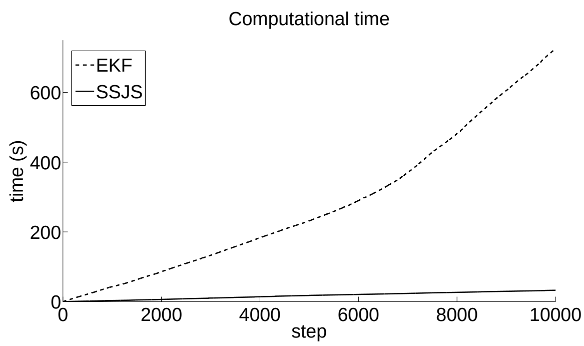

Fig. 4. Accumulated time along a simulation on synthetic data. SSJS time demand grows lineally, while for the EKF it grows dramatically, because the scenario was populate with a high amount of landmarks.

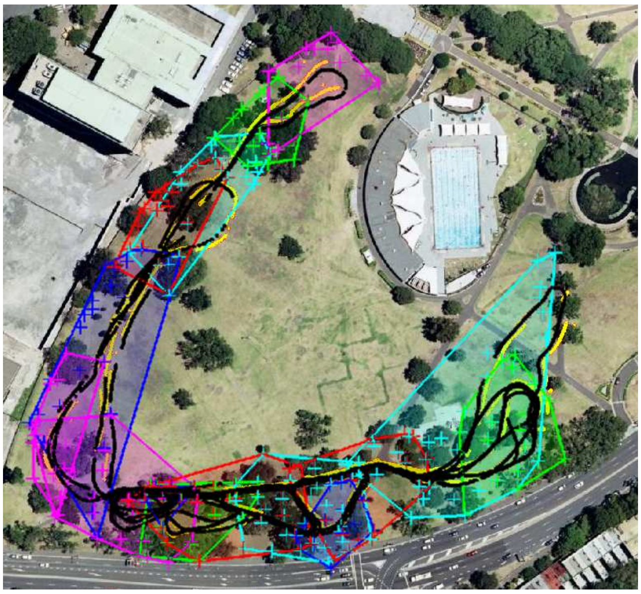

Fig. 5. Satellite image of the Victoria Park dataset (font: *Google Earth*). The GPS data captured during the mission is drawn in yellow, while the vehicle estimated trajectory is represented in black. The final submaps obtained with the SSJS are also shown.

recorded over a total time 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 data from a GPS. For the laser range data a tree detector function is provided together with the dataset. These detected trees are used as point feature landmarks. They usually have a large distance to each other and can be separated or uniquely identified with common data association techniques. However, at some cases, spurious data is detected and has to be removed. All the experiments were conducted on a Pentium Core Duo 1.77-GHz. The purpose of this experiment was twofold: 1) to evaluate the consistency of our approach, and 2) to analyse the computational complexity.

The final solution of the SSJS on the Victoria Park dataset is shown in Fig. 5. This figure shows qualitatively the level of correction of our approach. The final estimated map (black trajectory) is almost the same as the one generated using GPS data (yellow trajectory). However, having a single look on the plot it is not possible to extract any further conclusion such as the consistency of the method or the time consumption. Therefore, we decided to perform a quantitative evaluation. Hence, the consistency of our approach is checked via map consistency analysis. When the ground truth for the state variable is not known, the Normalised Innovation Squared (NIS) (5), can be used to analyse the consistency.

$$NIS = \nu_k^t \mathbf{S}_k^{-1} \nu_k < \chi^2_{r,1-\alpha} \tag{5}$$

where $r = dim(\mathbf{x}_k)$ is the degree of freedom, and α is the desired significance level (usually 0.05). Given the estimation of the innovation vector ν and the innovation matrix $\mathbf{S}$, the state $(\widehat{\mathbf{x}}, \mathbf{P})$ estimation is consistent when

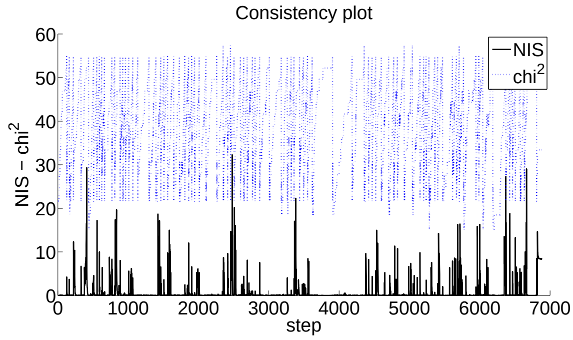

Fig. 6. Consistency test, where the dashed line represents the χ^2 corresponding to each step, and the continuous line corresponds to the NIS.

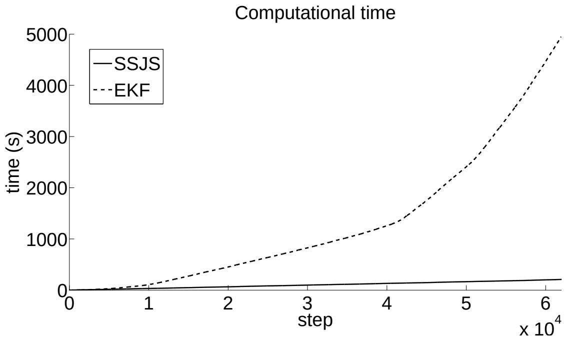

Fig. 7. Time required to compute the whole mission. Our approach is able to finish in about 200 seconds, while a standard EKF required almost 5000 seconds.

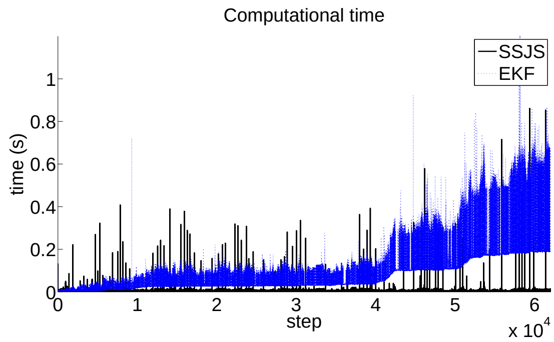

Fig. 8. Time per step along the whole mission. As the map gets larger, the EKF time per step increases considerably, while our approach, requires a constant time per step, except on the steps where a map fusion is conducted.

$NIS < \chi^2_{r,1-\alpha}$, otherwise the estimation is too optimistic, becoming inconsistent. Fig. 6 shows the SSJS always performing inside the theoretical consistency boundaries.

Regarding the computational demand, Fig. 7 shows the improvement of our method with respect to a standard EKF. As it happened with synthetic data, the EKF computational time increases on a quadratic order to the number of features in the map, while our approach increases almost lineally. This improvement is also visible in the time per step, see Fig. 8. The time per step is almost constant in our approach, while the EKF time per step increases with the size of the map. The time per step for our approach has some peaks corresponding to map fusion events. However, this peaks does not produce any dramatic delay, therefore the improvement with respect to the EKF is still considerable.

5. CONCLUSIONS

The Selective Submap Joining SLAM approach presented in this paper has been demonstrated to be suitable to map consistently large scale scenarios. The main contribution of our approach is the local map fusion strategy. This strategy determines whether to fuse two local maps, depending on the amount of information they share. The experimental results have shown the consistency of the method. Then, we demonstrated that the algorithm reduces the computational cost compared to the standard EKF method. Further work is intended to develop a 6DOF version of the algorithm. Extensions for the proposed approach include the generalization of the models to different vehicles and sensors.

ACKNOWLEDGEMENTS

The authors acknowledge the contribution of the authors of the Hierarchical SLAM approach Estrada et al. (2005) and the data association algorithm JCBB Neira and Tardós (2001), from the University of Zaragoza, for the fruitful discussions maintained.

REFERENCES

Aulinas, J., Lladó, X., Petillot, Y., and Salvi, J. (2009). Independent local mapping for large scale SLAM. In *Proceedings European Conference on Mobile Robotics (ECMR)*, volume 1, 67–76.

Bosse, M., Newman, P., Leonard, J., and Teller, S. (2004). SLAM in large scale cyclic environments using the atlas framework. *International Journal Robotics Research*, 23(12), 1113–1139.

Burgard, W., Fox, D., Jans, H., Matenar, C., and Thrun, S. (1999). Sonar-based mapping of large-scale mobile robot environments using EM. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML'99)*, 67–76. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Castellanos, J.A., Martínez-Cantin, R., Tardós, J.D., and Neira, J. (2007). Robocentric map joining: Improving the consistency of EKF-SLAM. *Robotics and Autonomous Systems*, 55(1), 21–29.

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

Estrada, C., Neira, J., and Tardós, J. (2005). Hierarchical SLAM: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4), 588–596.

Eustice, R., Singh, H., and Leonard, J. (2006). Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, 22(6), 1100–1114.

Frese, U. (2006). Treemap: An O(log n) algorithm for indoor simultaneous localization and mapping. *Autonomous Robots*, 21(2), 103–122.

Guivant, J. and Nebot, E. (2001). Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, 17(3), 242–257.

Knight, J., Davison, A., and Reid, I. (2001). Towards constant time SLAM using postponement. In *Proceedings IEEE/RJS International Conference on Intelligent Robots and Systems*, 406–412.

Leonard, J. and Newman, P. (2003). Consistent, convergent, and constant-time SLAM. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, 1143–1150. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Leonard, J.J. and Feder, H.J.S. (2000). A computationally efficient method for large-scale concurrent mapping and localization. In *Proceedings of the Ninth International Symposium on Robotics Research*, 169–176. Springer-Verlag.

Montemerlo, M., Thrun, S., Roller, D., and Wegbreit, B. (2003). FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, 1151–1156. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Nebot, E. (2009). Victoria park data set. http://www-personal.acfr.usyd.edu.au/nebot/dataset.htm.

Neira, J. and Tardós, J. (2001). Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6), 890 – 897.

Paz, L.M., Tardós, J., and Neira, J. (2008). Divide and conquer: EKF SLAM in O(n). *IEEE Transactions on Robotics*, 24(5), 1107–1120.

Piniés, P. and Tardós, J. (2008). Large scale SLAM building conditionally independent local maps: Application to monocular vision. *IEEE Transactions on Robotics*, 24(5), 1094–1106.

Smith, R., Self, M., and Cheeseman, P. (1988). A stochastic map for uncertain spatial relationships. In *Proceedings of the 4th International Symposium on Robotics Research*, 467–474. MIT Press, Cambridge, MA, USA.

Tardós, J.D., Neira, J., Newman, P.M., and Leonard, J.J. (2002). Robust mapping and localization in indoor environments using sonar data. *International Journal of Robotics Research*, 21(4), 311–330.

Thrun, S., Liu, Y., Koller, D., Ng, A., Ghahramani, Z., and Durrant-Whyte, H. (2004). Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7-8), 693–716.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press. ISBN-10: 0-262-20162-3.

Walter, M.R., Eustice, R.M., and Leonard, J.J. (2007). Exactly sparse extended information filters for feature-based SLAM. *International Journal of Robotics Research*, 26(4), 335–359.

Wan, E. and Van Der Merwe, R. (2001). Chapter 7 the unscented kalman filter. In *Kalman Filtering and Neural Networks*, 221–280. Wiley.

Williams, S.B., Dissanayake, G., and Durrant-whyte, H. (2002). An efficient approach to the simultaneous localisation and mapping problem. In *Proceedings IEEE International Conference Robotics and Automation*, volume 1, 406–411.