Efficient large scale SLAM including data association using the Combined Filter

César Cadena* Fabio Ramos† José Neira*

* Departamento de Informática e Ingenieria de Sistemas, Centro Politécnico Superior, Universidad de Zaragoza, Zaragoza 50018, Spain

† ARC Centre of Excellence for Autonomous Systems, Australian Centre for Field Robotics, The University of Sydney, Sydney, NSW 2006, Australia

***Abstract*— In this paper we describe the Combined Filter, a judicious combination of Extended Kalman (EKF) and Extended Information filters (EIF) that can be used to execute highly efficient SLAM in large environments. With the CF, filter updates can be executed in as low as $\mathcal{O}(\log n)$ as compared with other EKF and EIF based algorithms: $\mathcal{O}(n^2)$ for Map Joining SLAM, $\mathcal{O}(n)$ for Divide and Conquer (D&C) SLAM, and $\mathcal{O}(n^{1.5})$ for the Sparse Local Submap Joining Filter (SLSJF). We also study an often overlooked problem in computationally efficient SLAM algorithms: data association. In situations in which only uncertain geometrical information is available for data association, the CF Filter is as efficient as D&C SLAM, and much more efficient than Map Joining SLAM or SLSJF. If alternative information is available for data association, such as texture in visual SLAM, the CF Filter outperforms all other algorithms. In large scale situations, both algorithms based on Extended Information filters, CF and SLSJF, avoid computing the full covariance matrix and thus require less memory, but still the CF Filter is the more computationally efficient. Both simulations and experiments with the Victoria Park dataset, the DLR dataset, and an experiment using visual stereo are used to illustrate the algorithms' advantages.**

I. Introduction

In recent years, researches have devoted much effort to develop algorithms that improve the computational efficiency of SLAM, with the goal of being able to map large scale environments in real time [1]. One important contribution has been the idea of splitting the full map into local maps and then put the pieces back together in some way. Decoupled Stochastic Mapping [9], Constant Time SLAM [10] and the ATLAS system [2] are local mapping solutions close to constant time, although through approximations that reduce precision. Map Joining SLAM [15] and the Constrained Local Submap Filter [16] are exact solutions (except for linearizations) that require periodical $\mathcal{O}(n^2)$ updates. Exact solutions also include Treemap [5], Divide and Conquer (D&C) SLAM [14], Tectonic SAM [13] and Sparse Local Submap Joining Filter (SLSJF) SLAM [8]. Given a map of n features, the classical EKF SLAM algorithm is known to have a cost of $\mathcal{O}(n^2)$ per step. Two recent algorithms have provided important reductions in computational cost: D&C SLAM has an amortized cost $\mathcal{O}(n)$ per step, SLSJF SLAM reports a cost $\mathcal{O}(n^{1.5})$ per step in the worst cases. The Treemap has a cost $\mathcal{O}(\log n)$, although with topological restrictions on the environment, and a rather complex implementation.

This research has been funded in part by the Dirección General de Investigación of Spain under project DPI2006-13578 and by the European Union under project RAWSEEDS FP6-IST-045144.

In this paper we study the Combined Filter SLAM (CF SLAM), first introduced in [3], an algorithm to carry out large scale SLAM with computational cost as low as $\mathcal{O}(\log n)$ per step, without approximations other than linearizations. The CF SLAM algorithm is a judicious combination of Extended Kalman and Extended Information filters, combined with a Divide and Conquer local mapping strategy. Being a local mapping algorithm, it provides more consistent results, compared with Treemap, reported to have the same $\mathcal{O}(\log n)$ cost, but computing an absolute map [7]. CF SLAM is also conceptually simple and rather easy to implement. In this paper we show that the CF SLAM can also compute the data association and remain much more efficient than other EKF and EIF based SLAM algorithms.

This paper is organized as follows: the next section contains a brief description of the CF SLAM algorithm. In section III we describe the process to solve data association in the CF algorithm given either geometrical information or appearance-only information. In section IV we test the algorithm using four experiments, one simulated environment, the Victoria Park dataset, the DLR dataset and an experiment done with a stereo camera-in-hand. In the final section we summarize the results and draw the fundamental conclusions of this work.

II. Combined Filter SLAM

The Combined Filter SLAM algorithm has three main highlights (see algorithm 1):

Algorithm 1 CF SLAM Algorithm

```
maps ← {}
while data from sensor do
  map ← ekf_slam          Section: II-A
  if isempty(maps) then
    maps ← {map}
  else
    while size(map) ≥ size(maps{last})          Section: II-C
    or global map is needed do
      map ← eif_map_join(maps{last}, map) Section: II-B
      maps{last} ← {}
    end while
    maps ← {maps, map}
  end if
end while
```

	EKF-SLAM
Jacobians	$F_t = \left.\frac{\partial g(u_t,\mu_{t-1})}{\partial \mu_{t-1}}\right\rvert_{\hat{\mu}_{t-1}}$
	$G_t = \left.\frac{\partial g(u_t,\mu_{t-1})}{\partial u_t}\right\rvert_{\hat{u}_t}$
	$H_t = \left.\frac{\partial h(\mu_{t\|t-1})}{\partial \mu_{t\|t-1}}\right\rvert_{\hat{\mu}_{t\|t-1}}$
Prediction	$\mu_{t\|t-1} = g(u_t, \mu_{t-1})$
	$\mathbf{\Sigma}_{t\|t-1} = F_t \mathbf{\Sigma}_{t-1} F_t^T + G_t R_{t-1} G_t^T$
Innovation	$\nu_t = z_t - h(\mu_{t\|t-1})$
	$S_t = H_t \mathbf{\Sigma}_{t\|t-1} H_t^T + Q_t$
Test χ^2	$D^2 = \nu_t^T S_t^{-1} \nu_t$
Update	$K_t = \mathbf{\Sigma}_{t\|t-1} H_t^T / S_t$
	$\mathbf{\Sigma}_t = (I - K_t H_t)\mathbf{\Sigma}_{t\|t-1}$
	$\mu_t = \mu_{t\|t-1} + K_t \nu_t$

TABLE I
OPERATIONS CARRIED OUT USING EKF.

	Join with EIF
Jacobians	$G = \left.\frac{\partial g(\mu^+)}{\partial \mu^+}\right\rvert_{\hat{\mu}^+}$
	$H = \left.\frac{\partial h(\mu^-)}{\partial \mu^-}\right\rvert_{\hat{\mu}^-}$
Initialization	$\mu^- = g(\mu_1, \mu_2)$
	$\xi^- = \begin{bmatrix} \xi_1 \\ 0 \end{bmatrix}$
	$\mathbf{\Omega}^- = \begin{bmatrix} \mathbf{\Omega}_1 & 0 \\ 0 & 0 \end{bmatrix}$
Innovation	$\nu = \mu_2 - h(\mu^-)$
	$Q^{-1} = \mathbf{\Omega}_2$
Update	$\mathbf{\Omega} = \mathbf{\Omega}^- + H^T \mathbf{\Omega}_2 H$
	$\xi = \xi^- + H^T \mathbf{\Omega}_2 (\nu + H\mu^-)$
	$\mu = \mathbf{\Omega} \backslash \xi$

TABLE II
OPERATIONS CARRIED OUT FOR LOCAL MAP JOINING USING THE EXTENDED INFORMATION FILTER.

A. Local Map Building with Extended Kalman Filters

Local mapping is carried out using the well known Extended Kalman Filter (EKF). In EKF SLAM, a map $(\mu, \mathbf{\Sigma})$ includes the state μ to be estimated, which contains the current vehicle location and the location of a set of environment features. The covariance of μ, represented by $\mathbf{\Sigma}$, gives an idea of the precision in the estimation, 0 meaning total precision. EKF SLAM is an iterative prediction-sense-update process whose formulation we believe is widely known and is thus summarized in Table I. In CF SLAM, a sequence of maps of constant size p is produced. Each local map $(\mu_i, \mathbf{\Sigma}_i)$ is also stored in information form $(\xi_i, \mathbf{\Omega}_i)$, with $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$ and $\xi = \mathbf{\Sigma}^{-1}\mu$; both the information vector and the information matrix are computed in $\mathcal{O}(p^3)$, constant with respect to the total map size n. Each local map only keeps the final pose of the robot. EKF SLAM in local maps allows robust data association, e.g. with JCBB [12], and small local maps remain consistent.

B. Map Joining SLAM with Extended Information Filters

Local map Joining is carried out using the Extended Information filter (EIF). As reported in the Sparse Local Submap Joining filter (SLSJF) SLAM [8], given two consecutive local maps $(\mu_1, \mathbf{\Sigma}_1)$ and $(\mu_2, \mathbf{\Sigma}_2)$ to join, the resulting map $(\xi, \mathbf{\Omega})$ is predicted in the information form with the information of the first map, and an initial 0 (no information) from the second map. The innovation is computed considering the second map as a set of measurements for the full map ($z = \mu_2$, $Q = \mathbf{\Omega}_2^{-1}$), and the final update step computes the information state ξ and information matrix $\mathbf{\Omega}$ using the standard EIF equations, see table II (see [3] or [8] for more details). Keeping the vehicle locations from each local map in the final map allows to exploit the exact sparse structure of the information matrix and the join can be carried out in time linear with the final size of the map. . The covariance matrix is not kept after joins at the lower level.

C. Divide and Conquer Strategy

1) Best case: exploration: The computational cost of CF SLAM is studied in detail in [3]. For completeness, here we briefly summarize the most important aspects. As in the Divide and Conquer (D&C)SLAM algorithm [14], in CF SLAM map joining operations are carried out in a hierarchical binary tree fashion, instead of a sequential fashion (see algorithm 1). The leafs of the binary tree represent the sequence of l local maps of constant size p, computed with standard EKF SLAM, with cost $\mathcal{O}(p^3)$ each. These maps are joined pairwise to compute $l/2$ local maps of double their size ($2p$), with cost $\mathcal{O}(2p)$ each join, which will in turn be joined pairwise into $l/4$ local maps of size $4p$, with cost $\mathcal{O}(4p)$ each join, until finally two local maps of size $n/2$ will be joined into one full map of size n, the final map, with cost $\mathcal{O}(n)$.

During exploration, with the D&C strategy the total computational complexity of CF SLAM is:

$$\begin{aligned} C &= \mathcal{O}\left(p^3 l + \sum_{i=1}^{\log_2 l} \frac{l}{2^i}(2^i\, p)\right) \\ &= \mathcal{O}\left(p^2 n + \sum_{i=1}^{\log_2 n/p} n\right) \\ &= \mathcal{O}\left(p^2 n + n \log_2 n/p\right) \\ &= \mathcal{O}\left(n \log_2 n\right) \end{aligned}$$

CF SLAM offers a reduction in the total computational cost to $\mathcal{O}\left(n \log n\right)$, as compared with the total cost $\mathcal{O}\left(n^3\right)$ for EKF SLAM, and $\mathcal{O}\left(n^2\right)$ for D&C SLAM and SLSJF SLAM. Furthermore, as in D&C SLAM, the map to be generated at step t will not be required for joining until step $2\,t$. This allows us to amortize the cost $\mathcal{O}(t)$ at this step by dividing it up between steps $t+1$ to $2\,t$ in equal $\mathcal{O}(1)$ computations for each step. In this way, our amortized algorithm becomes $\mathcal{O}(\log n)$ per step.

2) Worst case: repeated traversal: The worst case occurs when two maps to join have full overlap, such as when traversing a loop for the second time. In this case the cost of recovery state in the join with EIF is $\mathcal{O}\left(n^2\right)$ [3]. The amortized cost per step will be $\mathcal{O}\left(n\right)$. In this same situation, the cost will be $\mathcal{O}\left(n^2\right)$ per step for Map Joining SLAM, SLSJF SLAM and the D&C SLAM amortized. Thus, CF SLAM is more efficient in all cases. In the experimental section we will show simulations and experiments that illustrate the computational cost of CF SLAM .vs the other algorithms in different situations.

III. The Data Association problem

The problem of data association is often ignored when evaluating the efficiency and effectiveness of a SLAM algorithm. Here we show that the CF SLAM algorithm remains in the worst of case as computationally effective as D&C SLAM, the fastest to our knowledge that maintains the full covariance matrix and thus allows data association based on stochastic geometry. In the order of thousands of features, CF SLAM will outperform D&C SLAM because of reduced memory requirements.

Data association can be usually solved in two ways. If a covariance matrix is available, we can carry out statistical tests to find possible matches based on stochastic geometry. If additional information is available, such as texture or appearance in vision sensors, we can obtain matchings based not on location but on appearance.

In the following we describe two data association algorithms, one belonging to each category, that can be used in CF SLAM.

A. Data association using geometrical information

In some situations, only geometrical information, the uncertain location of features relative to the sensor location, is available for data association. Such is the case of 2D points or straight walls obtained with a laser sensor.

In CF SLAM, data association inside a local map is carried out using JCBB [12] because the corresponding covariance matrix is available. The data association that remains to be solved is the identification of features that appear in two consecutive maps, lets call them M_1 and M_2, either local maps at the lower level, or maps resulting from previous joins in the D&C map joining process. When the two maps to be joined are local maps, their corresponding covariances are available and data association can be done with the JCBB also. If not, we first determine the overlap between the two maps, features that can potentially have pairings, and then we recover the covariance matrix for those features only. We proceed as follows:

1) Identify the overlap, a set of potential matches: This is denominated individual compatibility, IC. Individually compatible features are obtained by tessellating the environment space, as was proposed in [14], but in our case we represent the grid in polar coordinates, see fig. 1. For each feature in the second local map M_2, we assign an angular window of constant width in angle and height proportional to its distance from the origin, so that more distant features will have a larger region of uncertainty. The features in the first local map M_1 are referenced on M_2 through the last vehicle pose in M_1, μ_{x1}, which is the origin of map M_2. The uncertainty of μ_{x1} is recovered using the equation 1 and propagated on the these features, transformed to polar coordinates and embedded to angular windows. Features that intersect are considered individually compatible, giving IC.

2) Partial recovery of covariances: for intermediate maps that are not at the lower local level CF SLAM does not compute covariances. As shown in [8], we can recover some columns of the covariance matrix by solving the sparse linear equation 1. The columns that we require are given by IC. We

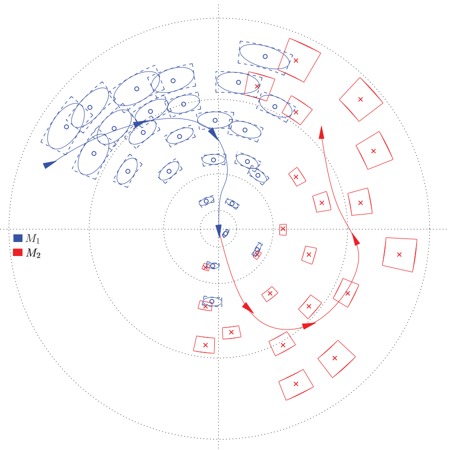

Fig. 1. Computing the individual compatibility matrix for two local maps using polar coordinates. The angular windows for the features in the M_2 have constant width in angle and height proportional to the distance to the origin. Blue ellipses represent the uncertainties of the predicted features of the first local map with respect to the base reference of the second. The ellipses are approximated by bounding windows.

form a column selection matrix E_{IC} to obtain the columns that are given by IC. If column i of the covariance matrix is required, we include this column vector in E_{IC}:

$$e_i = [\overbrace{0, \cdots, 0, 1}^{i}, 0, \cdots, 0]^T$$

The sparse linear system to be solved is as follows:

$$\mathbf{\Omega \Sigma}_{IC} = E_{IC} \tag{1}$$

This partial recovery of the covariance matrix allows to use robust joint compatibility tests for data association. The efficiency of solving equation 1 is the same as for the recovery of the state vector: in the best case, in exploration trajectories, the order is $\mathcal{O}(n)$, amortized $\mathcal{O}(\log n)$. In the worst case, for example when traversing a loop for the second time, the overlap will be the full map, and the cost will be $\mathcal{O}(n^3)$, amortized $\mathcal{O}(n^2)$.

3) Prediction and Observation: at this point the features of M_1 that have potential matches according to IC, and their covariances, are transformed to be referenced on M_2.

4) Randomized Joint Compatibility: we use the RJC algorithm of [14], a combination of JCBB and RANSAC that allows to carry out robust data association very efficiently.

In the experimental section we will show that using this algorithm, CF SLAM is computationally as efficient as D&C SLAM but requires less memory because the full covariance matrix is not computed.

B. Data association using appearance information

In some cases, features in local maps can have associated appearance information, such as texture coming from vision. In these cases, appearance can be coded using a descriptor vector, for example SIFT [11] or SURF [6]. In these cases, we proceed as follows in CF SLAM:

1) Obtain a set of potential matches: we find the best possible matches between the descriptors in M_1 and M_2 by searching for the nearest neighbor in the descriptor space. In this way we obtain the individual compatibility matrix, IC.

Algorithm 2 Data association for the Combined Filter using geometrical information only

Require: Two maps: $\{M_1, M_2\}$
Ensure: Hypothesis $\mathcal{H}$
  Find the set of potential matches $IC \leftarrow (\mu_1, \mu_2)$
  if covariance matrices are available **then**
    extract covariances
    $(\mathbf{\Sigma}_{1i}, \mathbf{\Sigma}_{2j}) \leftarrow select(\mathbf{\Sigma}_1, \mathbf{\Sigma}_2, IC)$
  else
    recover partial covariances
    $(\mathbf{\Sigma}_{1i}, \mathbf{\Sigma}_{2j}) \leftarrow recoveryP(\mathbf{\Omega}_1, \mathbf{\Omega}_2, IC)$ eq: 1
  end if
  $predictions = (h, H\Sigma H) \leftarrow predict\_map(\mu_{1i}, \mathbf{\Sigma}_{1i})$
  $observations = (z, R) \leftarrow (\mu_{2j}, \mathbf{\Sigma}_{2j})$
  $\mathcal{H} \leftarrow RJC(predictions, observations, IC)$

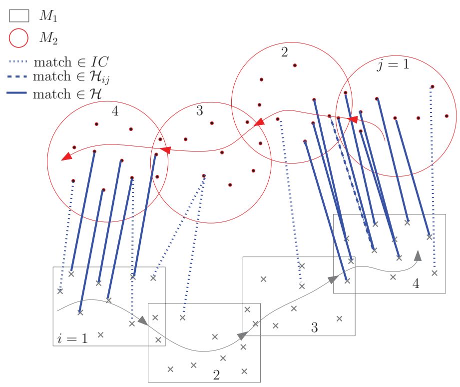

Fig. 2. Computing the data association hypothesis from two local maps. The rectangles and circles show the minimum local maps of M_1 and M_2 respectively. All lines are potential matches, and form IC. The lines that are not dotted belong to a hypothesis between a pair of the minimum local maps, $\mathcal{H}_{ij}$. The solid lines represent the final hypothesis $\mathcal{H}$.

2) Obtain a pairwise hypothesis using RANSAC: for each pair of minimum local maps, map i belonging to M_1 and map j belonging to M_2, that have a minimum number of matches (5 in our case) in IC, we use RANSAC [4] to find the subset $\mathcal{H}_{ij}$ of matches that corresponds to the best rigid-body transform between the two local maps. In fig. 2, the transformations between the pairs $(i = 1, j = 3)$, $(i = 2, j = 3)$ and $(i = 3, j = 2)$ are not evaluated because they do not have sufficient matches.

3) Obtain the final hypothesis: in most cases, the final hypothesis $\mathcal{H}$ is simply the result of joining all $\mathcal{H}_{ij}$. When there is ambiguity (one local map matched with two or more other local maps), we prefer the hypothesis for pairs of maps that have a smallest relative distance, because they have least relative errors. In fig 2 there is ambiguity between $\mathcal{H}_{41}$ and $\mathcal{H}_{42}$. In this case we accept the hypothesis $\mathcal{H}_{41}$.

Algorithm 3 Data association using appearance information

Require: Two maps: $\{M_1, M_2\}$
Ensure: Hypothesis $\mathcal{H}$
  Find set of potential matches $IC \leftarrow (descr_1, descr_2)$
  for each pair minimum local maps i, j in IC **do**
    $\mathcal{H}_{ij} \leftarrow RANSAC(\mu_{1i}, \mu_{2j})$
  end for
  $\mathcal{H} \leftarrow select(\mathcal{H}_{ij})$

IV. Simulations and Experiments

To compare the performance of the Combined Filter with other popular EKF and EIF based algorithms, we use simulations, publicly available datasets, and our own stereo visual SLAM experiment[1]. All algorithms were implemented in Matlab and executed on 2.4 GHz Intel Core 2 CPU 6600 and 3GB of RAM.

A. Simulated experiment

We simulated a robot moving in a 4-leaf clover trajectory. The robot is equipped with a range and bearing sensor. The features are uniformly distributed with a separation between them of $6m$. Data association was determined based only on geometric information using algorithm 2. Fig. 3(top) shows the computation cost per step of Map Joining SLAM and SLSJF vs. the amortized cost for the D&C SLAM and CF SLAM: middle left, cost of map updates, middle right: cost of data association, right: total cost including local map building. We can see the sublinear cost of CF SLAM in the map updates as expected. Fig. 3(bottom) shows the cumulative costs of map updates (middle left), data association (middle rigtht) and total cost (right). The algorithms based on EKFs did not solve the problem completely because they exceeded the available memory before the end of the experiment.

B. The Victoria Park dataset

The figure 4 shows the resulting map obtained by CF SLAM on the Victoria Park dataset. All algorithms solve this dataset correctly. This dataset has 228 features and 3615 odometry steps. The trajectory of the vehicle explores and revisits frequently, so the uncertainty does not grow much and errors are kept small. The data association was determined with the algorithm 2. This dataset is interesting to compare CF SLAM and SLSJF. Both algorithms require the recovery of the covariance submatrix for the overlap between the maps to be joined. There are some areas where the overlap is almost complete, thus requiring the recovery of almost the full covariance matrix. The cumulative computational costs are in figure 4. We can see that CF SLAM is the most efficient for map updates, but Map Joining SLAM is most efficient for data association. In total, both algorithms that use the D&C strategy tend to be most efficient.

C. The DLR dataset

In the DLR dataset, the robot is equipped with a camera, and carries out a trajectory almost all indoors. Features are white cardboard circles placed on the ground. This dataset has 560 features and 3297 odometry steps. The path consists of a large loop with several smaller loops in the way. Position errors grow enough so that sequential algorithms (map joining SLAM and SLSJF) become weak and fail in the data association to close the loop (see fig. 5, bottom right). The D&C algorithms, D&C SLAM and CF SLAM have better consistency properties, and both solve the data association for the loop closing in this dataset (see fig. 5, left). The cumulative computational costs are show in fig. 5, right. In this mostly

[1]Videos are available at `http://webdiis.unizar.es/~ccadena/research.html`

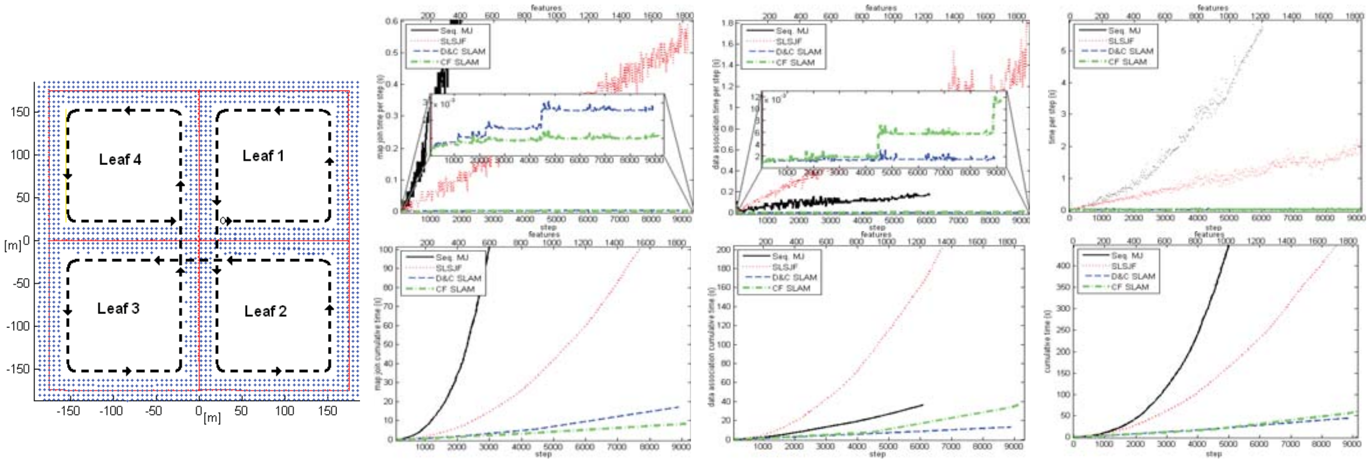

Fig. 3. Simulated experiment of a 4-leaf clover trajectory. On the left, the final map. The computational costs are shown on the right; top: map update time per step (left), data association time per step (middle), total time per step (right). Bottom: cumulative times for all algorithms.

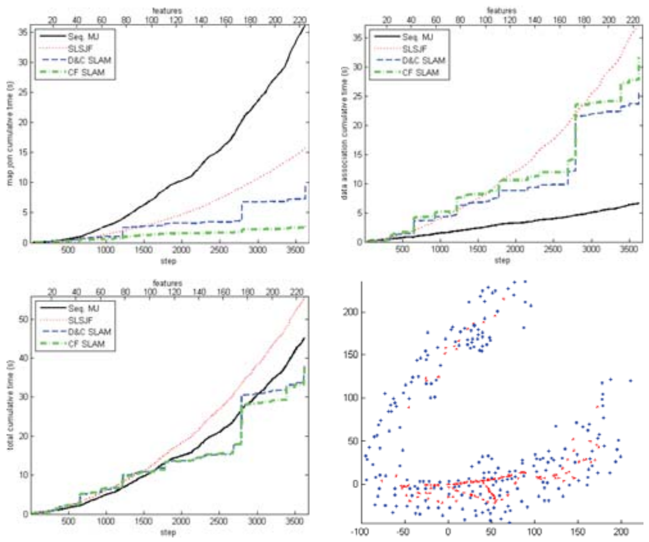

Fig. 4. Results using the Victoria Park dataset. Top: cumulative cost for map updates (left) and data association (right). Bottom left, cumulative total cost including local map building, map joining and data association. Bottom right, the final map with CF SLAM.

exploratory dataset, both D&C SLAM algorithms are clearly superior than both sequential algorithms. During loop closing, the cost of data association for both D&C algorithms is higher than that of both sequential algorithms, for the good reason that data association is computed correctly and the loop can be closed.

D. The visual stereo SLAM experiment

Finally, the CF SLAM was tested in a 3D environment with a high density of features. The sensor is a camera Triclops carried in hand. The path consists of a loop inside the Rose Building at the University of Sydney. We obtain the 3D position of points from the computation of the dense stereo point cloud that corresponds to each SIFT feature. The experiment consists of 132 shoots, with a total of 6064 features. The figure 6(left) shows the map obtained with the CF SLAM. The SLSJF obtains an incorrect map, fig. 6, bottom right. SLSJF is a sequential map joining algorithm, thus it is expected to provide less consistent results than D&C algorithms. In this experiments, this results in incorrect loop closure.

Both Map Joining SLAM and the D&C SLAM exceeded available memory in Matlab. The data association is obtained with the algorithm fig. 3. Cumulative computational costs are show in fig. 6: for map updates (top center), for data association (top right), and total cumulative cost (bottom center). CF SLAM clearly outperforms all the other algorithms.

V. Discussion

In this paper we have described the Combined Filter, an algorithm that can carry out SLAM in as low as$\mathcal{O}(\log n)$ per step . It brings together the advantages of different methods that have been proposed to optimize EKF and EIF SLAM. There is no loss of information, because the solution is computed without approximations, except for linearizations. It is conceptually simple and easy to implement. There are no restrictions on the topology of the environment and trajectory, although, as it is the case in all other SLAM algorithms, the computational efficiency will depend on this.

The problem of data association has also been addressed in this paper. We have shown that we can provide data association based on stochastic geometry that makes the CF SLAM algorithm as efficient as the most efficient algorithm that computes covariance matrices, with far less memory requirements. If appearance information is available for data association, then CF can clearly outperform all other algorithms based on EKFs and EIFs.

From our experiments it is clear that the greatest computational weight can lie in data association. Our future work includes the development of more robust and more efficient data association techniques to use in CF SLAM. Among the most promising appearance-based techniques, we will consider the 'bag-of-words' methods and Conditional Random Fields.

References

[1] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

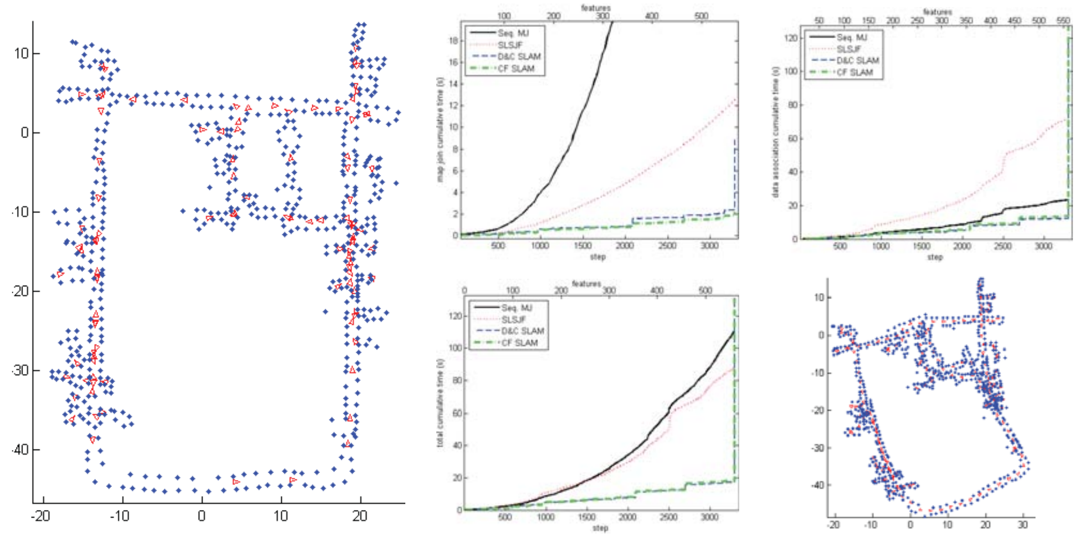

Fig. 5. Results using the DLR-Spatial-Cognition dataset with D&C SLAM and CF SLAM (left), and with Map Joining SLAM and SLSJF (bottom right). Cumulative cost per map joining (center top), data association (top right) and total cost including local map building (center bottom).

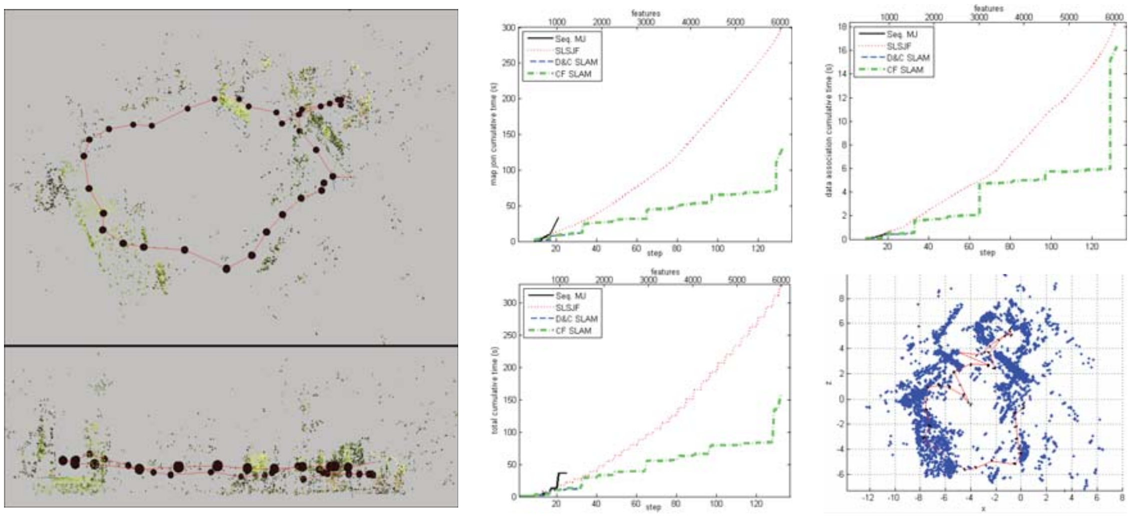

Fig. 6. Final 3D map using 132 shoots of dense stereo data from a Triclops Pointgrey camera with the CF SLAM (left). We show the cumulative cost per map joining (center top), data association (top right) and total cost including build local maps, map joining and data association (center bottom). Map Joining SLAM and D&C SLAM exceed available memory capacity in shoots 24 and 32, respectively. The incorrect final map with SLSJF (bottom right).

[2] M. Bosse, P. M. Newman, J. J. Leonard, and S. Teller. SLAM in large-scale cyclic environments using the atlas framework. *Int. J. Robotics Research*, 23(12):1113–1139, December 2004.

[3] C. Cadena and J. Neira. SLAM in O(log n) with the Combined Kalman - Information Filter. In *Proc. IEEE/RJS Int. Conference on Intelligent Robots and Systems*, 2009. (submitted).

[4] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

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

[6] Tinne Tuytelaars Herbert Bay and Luc Van Gool. Surf: Speeded up robust features. In *Proceedings of the 9th European Conference on Computer Vision*, volume 3951, pages 404–417. Springer LNCS, 2006.

[7] Shoudong Huang and Gamini Dissanayake. Convergence and Consistency Analysis for Extended Kalman Filter Based SLAM. *IEEE Trans. Robotics*, 23(5):1036–1049, October 2007.

[8] Shoudong Huang, Zhan Wang, and Gamini Dissanayake. Sparse local submap joining filters for building large-scale maps. *IEEE Trans. Robotics*, 24:1121–1130, 2008.

[9] J. Leonard and H.S.J. Feder. Decoupled stochastic mapping. *IEEE Journal of Oceanic Engineering*, 26(4):561–571, 2001.

[10] J.J. Leonard and P.M. Newman. Consistent, convergent and constant-time SLAM. In *Proc. Int. Joint Conf. Artificial Intelligence*, Acapulco, Mexico, August 2003.

[11] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[12] J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robotics and Automation*, 17(6):890–897, 2001.

[13] K. Ni, D. Steedly, and F. Dellaert. Tectonic SAM: Exact, Out-of-Core, Submap-Based SLAM. In *2007 IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, April 2007.

[14] Lina M. Paz, J. D. Tardós, and J. Neira. Divide and conquer: Ekf slam in O(n). *IEEE Trans. Robotics*, 24(5):1107–1120, October 2008.

[15] J. D. Tardós, J. Neira, P. M. Newman, and J. J. Leonard. Robust Mapping and Localization in Indoor Environments using Sonar Data. *Int. J. Robotics Research*, 21(4):311–330, 2002.

[16] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte. An efficient approach to the simultaneous localisation and mapping problem. In *Proc. IEEE Int. Conf. Robotics and Automation*, volume 1, pages 406–411, Washington DC, 2002.