

Scalable SLAM building Conditionally Independent Local Maps

Pedro Piniés and Juan D. Tardós

Abstract—Local maps algorithms have demonstrated to be well suited for mapping large environments as can reduce the computational cost and improve the consistency of the final estimation. In this paper we present a new technique that allows the use of local mapping algorithms in the context of EKF SLAM but without the constrain of probabilistic independence between local maps. By means of this procedure, salient features of the environment or vehicle state components as velocity or global attitude, can be shared between local maps without affecting the posterior joining process or introducing any undesirable approximations in the final global map estimate. The overload cost introduced by the technique is minimum since building up local maps does not require any additional operations apart from the usual EKF steps. As the algorithm works with covariance matrices, well-known data association techniques can be used in the usual manner. To test the technique, experimental results using a monocular camera in an outdoor environment are provided. The initialization of the features is based on the inverse depth algorithm.

I. INTRODUCTION

The Simultaneous Localization and Mapping problem has been the subject of continuous attention during approximately the last two decades (for recent reviews, see [1], [2], [3]). It consists in processing the information obtained by a sensor embarked on a mobile platform to obtain an estimate of its own pose while building a map of the environment. One of the most popular techniques used is based on the Extended Kalman Filter (EKF) [4], [5], which provides, under certain assumptions, a consistent way to deal with the uncertainties associated with the movement and measurement processes.

Despite of its relative success, EKF SLAM suffers from two main limitations. The growing computational cost per step that is $O(n^2)$ in the number of features stored in the map and the optimistic estimation of the covariance matrix due to linearization approximations which may result in inconsistency [6], [7].

In the context of Gaussian filters we can find several techniques in the literature that make improvements on the previous limitations.

To deal with the computational cost we find: The Compressed EKF filter [8] which makes local updates in small regions postponing the total map update until the vehicle moves to another area. However for each region it has to perform some additional jacobian matrix calculations in order to update the global map.

Techniques based on the Information Filter take advantage of the near to sparse structure of the information matrix (the inverse of the map covariance matrix) to reduce the

computational burden. In the Sparse Extended Information Filter (SEIF) [9] the information matrix of the SLAM posterior is approximated by rounding to zero the small off-diagonal elements. This prevents the inter-landmark links from forming and therefore limits the density of the information matrix. The Thin Junction Tree Filter (TJTF) [10] and the Exactly Sparse Extended Information Filter (ESEIF) [11] maintain the sparsity by discarding some *weak* information such as the robot odometry. The Exactly Sparse Delayed state Filters (ESDF) [12] avoids the previous approximations by including the trajectory of the vehicle in the state vector which makes the information matrix exactly sparse. Nevertheless, it performs some approximations as well when recovers partially the state in order to evaluate the jacobians in the current state estimate. A recent and very efficient technique that also works in information space is the Treemap algorithm [13], which requires $O(\log n)$ time per step to recover a part of the state and $O(n)$ to recover the whole map. However, it has only been tested on simulations, with known data association. One important limitation of the techniques based on the information form is the difficulty to perform data association since the covariance matrix is not available.

The problem of map consistency has motivated algorithms as the UKF [14] which achieves better consistency properties but without taking into account the computational complexity.

Finally, there exist some techniques that confront both problems at the same time as the Local Map Sequencing [15], the Constrained Local Submap Filter (CLSIF) [16], the Hierarchical Map algorithm [17] or the more recent D&C [18]. These algorithms allow direct implementation of data association methods since they work with covariance matrices. However, all of them are based on statistically independent local maps which prevent them from sharing information between maps as salient features of the environment or vehicle state components as velocity or global attitude.

We propose in this paper a new technique developed in the covariance space that allows correlated local maps share information in a consistent manner. As this technique can be used jointly with the previous local map algorithms it shares and can improve their attractive cost and more consistent behavior. It represents as well the natural choice for applications with sensors that have long time feature initializations as in monocular vision since a good estimate of a feature could require several robot poses or with inertial sensors (IMU) where the velocity and attitude of the IMU in the current map depends on their values in previous maps.

Pedro Piniés and Juan D. Tardós are with Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, María de Luna 1, E-50018, Zaragoza, Spain, {ppinies, tardos}@unizar.es

Section II develops the theoretical fundamentals of the new technique as well as the equations needed to implement it. In section III examples of possible algorithm implementations are provided. In section IV the results obtained in a large real outdoor experiment with a monocular camera are shown. Finally in section V we summarize the main characteristics of the algorithm presented and propose future work.

II. THEORETICAL DESCRIPTION

A. Basic Probability Concepts

For the reader convenience, this subsection summarizes the basic probability concepts that will be used in the rest of the paper. More detailed presentations can be found in [1], [19].

The *conditional probability* of a random variable \mathbf{x} given the value of the random variable \mathbf{y} is defined as:

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})} \quad (1)$$

where $p(\mathbf{x}, \mathbf{y})$ is the *joint distribution*, and $p(\mathbf{y})$ is the *marginal distribution* of \mathbf{y} .

Two random variables \mathbf{x} and \mathbf{y} are *conditionally independent* given \mathbf{z} when:

$$p(\mathbf{x}, \mathbf{y}|\mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{y}|\mathbf{z}) \quad (2)$$

which is equivalent to:

$$p(\mathbf{x}|\mathbf{y}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z}) \quad (3)$$

In this case, if \mathbf{z} is known, \mathbf{y} does not provide any additional information about \mathbf{x} .

In the case of two random variables that are jointly Gaussian, with mean and covariance given by:

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N} \left(\begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{y}} \end{bmatrix}, \begin{bmatrix} P_x & P_{xy} \\ P_{yx} & P_y \end{bmatrix} \right) \quad (4)$$

the process of *marginalization* consists simply in choosing the appropriate rows and columns of the mean vector and covariance matrix:

$$p(\mathbf{y}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{x} = \mathcal{N}(\hat{\mathbf{y}}, P_y) \quad (5)$$

and the process of *conditioning* is performed by [20]:

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})} = \mathcal{N}(\hat{\mathbf{x}}', P_x') \quad (6)$$

$$\hat{\mathbf{x}}' = \hat{\mathbf{x}} + P_{xy}P_{yy}^{-1}(\mathbf{y} - \hat{\mathbf{y}}) \quad (7)$$

$$P_x' = P_x - P_{xy}P_{yy}^{-1}P_{yx} \quad (8)$$

B. Notation Definitions

Fig(1) shows on the top a Bayesian Net which represents the probabilistic dependencies between the usual variables involved in a SLAM estimation process. The \mathbf{u}_i nodes represent the movement inputs introduced to the system, the \mathbf{x}_i nodes are the successive locations of the platform, the \mathbf{f}_i nodes the features of the map and the \mathbf{z}_i nodes the observations of the features gathered from the platform. We will use this example, without loss of generality, to illustrate the development of the technique.

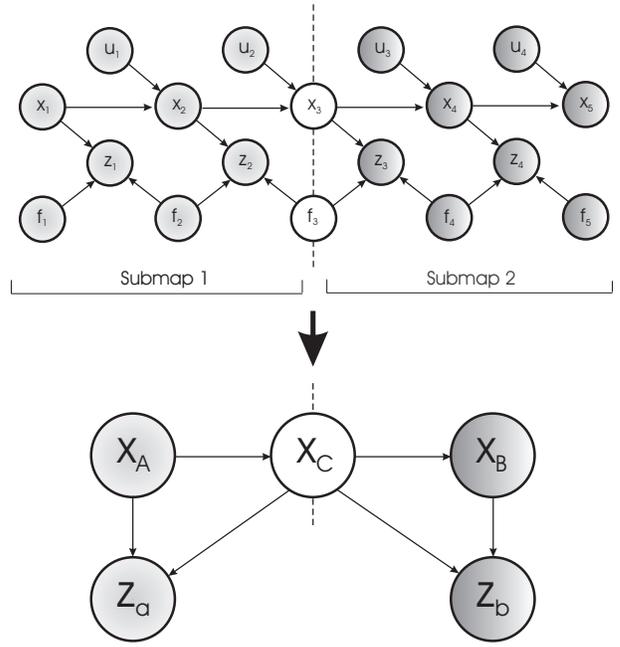


Fig. 1. Bayesian Net that describes the probabilistic dependencies between SLAM variables. It also reveals the intrinsic SLAM structure.

As can be noticed, the whole map has been split into two submaps that share in common one robot location \mathbf{x}_3 and a feature \mathbf{f}_3 . Both elements are involved through observations \mathbf{z}_2 and \mathbf{z}_3 in the estimation of the two submaps. For clarity reasons in the exposition and notation of the following subsections, several nodes in the bayesian net are grouped together as it is shown in the bottom of Fig(1). The description of each node is:

- \mathbf{x}_A : Features and robot positions that are only observed in the first submap.
- \mathbf{x}_B : Features and robot positions that are only observed in the second submap.
- \mathbf{x}_C : Common features and robot position that are observed both in the first and second submaps.
- \mathbf{z}_a : Observations in the first submap gathered from features in \mathbf{x}_A and \mathbf{x}_C .
- \mathbf{z}_b : Observations in the second submap gathered from features in \mathbf{x}_B and \mathbf{x}_C .

C. Conditionally Independent Maps

This subsection describes how the global map defined before can be completely recovered without approximations given its corresponding submaps. Notice that the submaps are not independent because they have elements in common, in this example, a feature and a robot position.

As can be seen in Fig(1), the only connection between the set of nodes $(\mathbf{x}_A, \mathbf{z}_a)$ and $(\mathbf{x}_B, \mathbf{z}_b)$ is through node \mathbf{x}_C , i.e. both subgraphs are *d-separated* given \mathbf{x}_C [21]. This implies that nodes \mathbf{x}_A and \mathbf{z}_a are conditionally independent of nodes \mathbf{x}_B and \mathbf{z}_b given node \mathbf{x}_C . Intuitively this means that if \mathbf{x}_C is known, submaps 1 and 2 do not carry any additional information about each other. In the following, we will call this the *Submap Conditional Independence (CI) Property*.

The estimation of the global map is completely defined by its probability density function:

$$p(\mathbf{x}_A, \mathbf{x}_B, \mathbf{x}_C | \mathbf{z}_a, \mathbf{z}_b) \quad (9)$$

which can be factorized as follows:

$$\begin{aligned} p(\mathbf{x}_A, \mathbf{x}_B, \mathbf{x}_C | \mathbf{z}_a, \mathbf{z}_b) &= \\ &= p(\mathbf{x}_A | \mathbf{x}_B, \mathbf{x}_C, \mathbf{z}_a, \mathbf{z}_b) p(\mathbf{x}_B, \mathbf{x}_C | \mathbf{z}_a, \mathbf{z}_b) \\ &= p(\mathbf{x}_A | \mathbf{x}_C, \mathbf{z}_a) p(\mathbf{x}_B, \mathbf{x}_C | \mathbf{z}_a, \mathbf{z}_b) \end{aligned} \quad (10)$$

where the first equality comes from Eq(1) and the second from the Submap CI Property.

Given the previous factorization, we will show how each of the factors can be recovered from the information stored in the probability densities of the submaps.

When the first submap is finished, its estimation is given by:

$$p(\mathbf{x}_A, \mathbf{x}_C | \mathbf{z}_a) = p(\mathbf{x}_A | \mathbf{z}_a, \mathbf{x}_C) p(\mathbf{x}_C | \mathbf{z}_a) \quad (11)$$

which has been factorized using Eq(1). Notice that the first term of the factorization equals the first factor of Eq(10)

Before starting the second submap, the elements of the first map that are not expected to be re-observed, i.e. \mathbf{x}_A , are marginalized out. The probability that describes the initial state of the second submap is then given by:

$$p(\mathbf{x}_C | \mathbf{z}_a) = \int p(\mathbf{x}_A, \mathbf{x}_C | \mathbf{z}_a) d\mathbf{x}_A \quad (12)$$

While building up the second map, the influence of the new observations \mathbf{z}_b is incorporated to the estimation either through re-observing \mathbf{x}_C features or by creating new features \mathbf{x}_B . The density probability that describes the final second map is:

$$p(\mathbf{x}_C, \mathbf{x}_B | \mathbf{z}_a, \mathbf{z}_b) \quad (13)$$

which is directly the last factor needed to recover the global map in Eq(10).

Therefore, as conclusion of this subsection, all the information needed to recover the global map can be obtained from the information stored in each of the submaps. Notice that no-assumptions has been made about the particular distribution of the probability densities. The previous factorizations only depend on general probabilistic theorems and on the intrinsic structure of SLAM.

D. Derivation for Gaussian probabilities

In this subsection we will focus on the case when the probability densities are Gaussian. A close analytical derivation to obtain the mean vector and covariance matrix of the global map given the moments of the submaps will be described. Suppose we have built two submaps:

$$p(\mathbf{x}_A, \mathbf{x}_C | \mathbf{z}_a) = \mathcal{N} \left(\begin{bmatrix} \hat{\mathbf{x}}_{A_a} \\ \hat{\mathbf{x}}_{C_a} \end{bmatrix}, \begin{bmatrix} P_{A_a} & P_{AC_a} \\ P_{CA_a} & P_{C_a} \end{bmatrix} \right) \quad (14)$$

$$p(\mathbf{x}_C, \mathbf{x}_B | \mathbf{z}_a, \mathbf{z}_b) = \mathcal{N} \left(\begin{bmatrix} \hat{\mathbf{x}}_{C_{ab}} \\ \hat{\mathbf{x}}_{B_{ab}} \end{bmatrix}, \begin{bmatrix} P_{C_{ab}} & P_{CB_{ab}} \\ P_{BC_{ab}} & P_{B_{ab}} \end{bmatrix} \right) \quad (15)$$

where upper case subindexes are for state vector components whereas lower case subindexes describe which observations \mathbf{z} have been used to obtain the estimate. For example, in the first submap, common elements \mathbf{x}_C have been estimated using only observations \mathbf{z}_a , hence, the mean and covariance estimates are denoted by $\hat{\mathbf{x}}_{C_a}$ and P_{C_a} respectively.

We are interested in recovering the global map, represented by:

$$\begin{aligned} p(\mathbf{x}_A, \mathbf{x}_B, \mathbf{x}_C | \mathbf{z}_a, \mathbf{z}_b) &= \\ &= \mathcal{N} \left(\begin{bmatrix} \hat{\mathbf{x}}_{A_{ab}} \\ \hat{\mathbf{x}}_{C_{ab}} \\ \hat{\mathbf{x}}_{B_{ab}} \end{bmatrix}, \begin{bmatrix} P_{A_{ab}} & P_{AC_{ab}} & P_{AB_{ab}} \\ P_{CA_{ab}} & P_{C_{ab}} & P_{CB_{ab}} \\ P_{BA_{ab}} & P_{BC_{ab}} & P_{B_{ab}} \end{bmatrix} \right) \end{aligned} \quad (16)$$

Comparing Eq(16) and Eq(15) we observe that the second submap by itself coincides exactly with the last two blocks of the global map. Only the terms related to \mathbf{x}_A in the global map will need to be computed. This is because the first submap has only been updated with the observations \mathbf{z}_a , but not with the more recent observations \mathbf{z}_b . In the following we will show how to *back-propagate* \mathbf{z}_b to update the first submap and how to compute the correlation between both submaps $P_{AB_{ab}}$.

1) *Back-propagation*: Since we want to update the first submap we will focus on the corresponding elements in the global map. Thus, the components of $\hat{\mathbf{x}}_{B_{ab}}$ are marginalized out.

$$p(\mathbf{x}_A, \mathbf{x}_C | \mathbf{z}_a, \mathbf{z}_b) = \int p(\mathbf{x}_A, \mathbf{x}_B, \mathbf{x}_C | \mathbf{z}_a, \mathbf{z}_b) d\mathbf{x}_B \quad (17)$$

which corresponds to eliminate the third row and column of Eq(16)

The conditional distribution $p(\mathbf{x}_A | \mathbf{z}_a, \mathbf{z}_b, \mathbf{x}_C)$ can be obtained conditioning on \mathbf{x}_C :

$$\hat{\mathbf{x}}_{A|C} = \hat{\mathbf{x}}_{A_{ab}} + P_{AC_{ab}} P_{C_{ab}}^{-1} (\mathbf{x}_C - \hat{\mathbf{x}}_{C_{ab}}) \quad (18)$$

$$P_{A|C} = P_{A_{ab}} - P_{AC_{ab}} P_{C_{ab}}^{-1} P_{CA_{ab}} \quad (19)$$

Applying the same process to the first submap, the mean and covariance of the conditional probability $p(\mathbf{x}_A | \mathbf{z}_a, \mathbf{x}_C)$ in terms of Eq(14) are given by:

$$\hat{\mathbf{x}}_{A|C} = \hat{\mathbf{x}}_{A_a} + P_{AC_a} P_{C_a}^{-1} (\mathbf{x}_C - \hat{\mathbf{x}}_{C_a}) \quad (20)$$

$$P_{A|C} = P_{A_a} - P_{AC_a} P_{C_a}^{-1} P_{CA_a} \quad (21)$$

Using the Submap CI Property we know that:

$$p(\mathbf{x}_A | \mathbf{z}_a, \mathbf{z}_b, \mathbf{x}_C) = p(\mathbf{x}_A | \mathbf{z}_a, \mathbf{x}_C) \quad (22)$$

and therefore the expression for the means in Eqs(18,20) must be equal for all \mathbf{x}_C , and the expressions for the covariances in Eqs(19,21) must be equal as well.

After performing some manipulations of the equalities, the following equations to update the components of the first

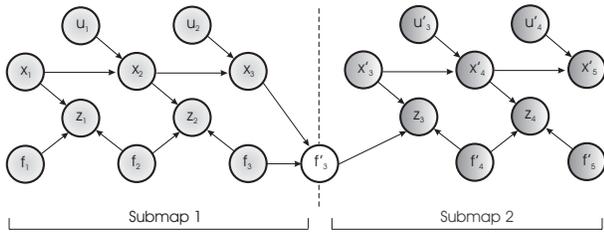


Fig. 2. Bayesian Net which illustrates the process to generate a local map which its own base reference

submap are obtained:

$$\begin{aligned} K &= P_{AC_a} P_{C_a}^{-1} \\ &= P_{AC_{ab}} P_{C_{ab}}^{-1} \end{aligned} \quad (23)$$

$$P_{AC_{ab}} = K P_{C_{ab}} \quad (24)$$

$$P_{A_{ab}} = P_{A_a} + K(P_{C_{A_{ab}}} - P_{C_{A_a}}) \quad (25)$$

$$\hat{\mathbf{x}}_{A_{ab}} = \hat{\mathbf{x}}_{A_a} + K(\hat{\mathbf{x}}_{C_{ab}} - \hat{\mathbf{x}}_{C_a}) \quad (26)$$

observe that only the updated components of the common elements between maps, i.e. $\hat{\mathbf{x}}_{C_{ab}}$ and $P_{C_{ab}}$, are required to propagate the influence of the new observations \mathbf{z}_b in the first map. The updated values of $\hat{\mathbf{x}}_C$ are known because they can be obtained from the probability density of the second submap Eq(15).

2) *Computing the Correlation between Submaps*: To obtain the covariance matrix of the whole map (for example when a Map joining technique is performed), the correlation term between them, $P_{AB_{ab}}$, must be calculated as well. For doing so, we first calculate the covariance of $p(\mathbf{x}_A, \mathbf{x}_B | \mathbf{z}_a, \mathbf{z}_b, \mathbf{x}_C)$ from the terms of the covariance of the global map:

$$\begin{bmatrix} P_{A_{ab}} - P_{AC_{ab}} P_{C_{ab}}^{-1} P_{CA_{ab}} & P_{AB_{ab}} - P_{AC_{ab}} P_{C_{ab}}^{-1} P_{CB_{ab}} \\ P_{BA_{ab}} - P_{BC_{ab}} P_{C_{ab}}^{-1} P_{CA_{ab}} & P_{B_{ab}} - P_{BC_{ab}} P_{C_{ab}}^{-1} P_{CB_{ab}} \end{bmatrix} \quad (27)$$

By the Submap CI Property we know that \mathbf{x}_A and \mathbf{x}_B are conditionally independent given \mathbf{x}_C and therefore the correlation term in Eq(27) must be 0. As a consequence:

$$\begin{aligned} P_{AB_{ab}} &= P_{AC_{ab}} P_{C_{ab}}^{-1} P_{CB_{ab}} \\ &= K P_{CB_{ab}} \end{aligned} \quad (28)$$

All derivations carried out until now represent all submaps with respect to the same common global reference, in this example with respect to the initial robot position \mathbf{x}_1 in the first submap as can be seen in Fig(1). With this representation the algorithm proposed can be considered as an improved CEKF filter since the final map estimate should be the same but where much less computational operations have been required.

E. Local reference implementation

Similar consistency properties of Local Map Sequencing algorithms can be achieved by using local references when building submaps. In Fig(2) a graphical representation of the transformations required in the algorithm is shown. The first

local map finishes when the robot arrives to position \mathbf{x}_3 . The last feature included in the submap is \mathbf{f}_3 when the robot was at \mathbf{x}_2 after observing \mathbf{z}_2 . We want now to represent submap 2 with respect to a local reference given by the current vehicle position \mathbf{x}_3 . In the following a tilde will be used to denote entities relative to the new base reference. As feature \mathbf{f}_3 will be shared between both submaps a local replication \mathbf{f}'_3 is generated and included in the maps:

$$\mathbf{f}'_3 = \ominus \mathbf{x}_3 \oplus \mathbf{f}_3 \quad (29)$$

To update the first local map the same *Back-propagation* algorithm can be applied. In the example proposed the update is carried out through the common element \mathbf{f}'_3 . If in addition the maps are joined, correlation between maps components has to be calculated and both maps have to be transformed to a common reference.

III. ALGORITHM IMPLEMENTATION

An implementation example of the *EKF-SLAM with CI submaps* algorithm is shown in algorithm 1. The implementation follows the structure of the standard EKF SLAM algorithm but introduces two new functions: $\{global, local\}_{map_transition}$ and *Back-propagation*.

The function called $\{global, local\}_{map_transition}$, allows us to create a new submap using a global or local reference frame when the number of features in the current local map exceeds a given threshold. In this example a global reference implementation is shown. In algorithm 1, variable \mathbf{m} represents a struct array that contains all submaps. Each map is defined by its estimated state vector and covariance matrix as follows:

$$\mathbf{m}_j \cdot \hat{\mathbf{x}}^W ; \mathbf{m}_j \cdot \mathbf{P}^W \quad \begin{bmatrix} \hat{\mathbf{x}}_R^W \\ \hat{\mathbf{x}}_{F_A}^W \\ \hat{\mathbf{x}}_{F_C}^W \end{bmatrix} ; \begin{bmatrix} \mathbf{P}_{RR}^W & \mathbf{P}_{RFA}^W & \mathbf{P}_{RFC}^W \\ \mathbf{P}_{FAR}^W & \mathbf{P}_{FA}^W & \mathbf{P}_{FAC}^W \\ \mathbf{P}_{FCR}^W & \mathbf{P}_{FCFA}^W & \mathbf{P}_{FC}^W \end{bmatrix} \quad (30)$$

where local map j is represented with respect to the global reference frame W , $\hat{\mathbf{x}}_R$ represents the robot position, $\hat{\mathbf{x}}_{F_A}$ are the features only seen in the current submap and $\hat{\mathbf{x}}_{F_C}$ are the features that will be in common with the next submap.

In a naive implementation of the function $global_map_transition(\mathbf{m}_j)$ the last robot location in the current map is replicated twice in the new submap \mathbf{m}_{j+1} and common features $\hat{\mathbf{x}}_{F_C}^W$ are directly copied to the new state vector. One of the copies of the robot position will change as the robot moves through the new map carrying the current position while the other will be used as a common element to propagate correlations. The initial new map state vector and covariance matrix will be:

$$\mathbf{m}_{j+1} \cdot \hat{\mathbf{x}}^W ; \mathbf{m}_{j+1} \cdot \mathbf{P}^W \quad \begin{bmatrix} \hat{\mathbf{x}}_{R_{j+1}}^W \\ \hat{\mathbf{x}}_{R_{j+1}}^W \\ \hat{\mathbf{x}}_{F_C}^W \end{bmatrix} ; \begin{bmatrix} \mathbf{P}_{R_{j+1}}^W & \mathbf{P}_{R_{j+1}}^W & \mathbf{P}_{R_{j+1}F_C}^W \\ \mathbf{P}_{R_{j+1}}^W & \mathbf{P}_{R_{j+1}}^W & \mathbf{P}_{R_{j+1}F_C}^W \\ \mathbf{P}_{F_C R_{j+1}}^W & \mathbf{P}_{F_C R_{j+1}}^W & \mathbf{P}_{F_C}^W \end{bmatrix} \quad (31)$$

The function called *Back-propagation* can be implemented directly using Eqs(23,24,25,26).

Algorithm 1 : EKF-SLAM with CI submaps

```
 $\mathbf{z}_0, \mathbf{R}_0 = \text{get\_observations}$   
 $\mathbf{m}_0 = \text{init\_map}(\mathbf{z}_0, \mathbf{R}_0)$   
 $j = 0$   
for  $k = 1$  to steps do  
   $\mathbf{u}_{k-1}, \mathbf{Q}_{k-1} = \text{get\_odometry}$   
   $\mathbf{m}_j = \text{EKF\_prediction}(\mathbf{m}_j, \mathbf{u}_{k-1}, \mathbf{Q}_{k-1})$   
   $\mathbf{z}_k, \mathbf{R}_k = \text{get\_observations}$   
   $\mathcal{DA}_k = \text{data\_association}(\mathbf{m}_j, \mathbf{z}_k, \mathbf{R}_k)$   
   $\mathbf{m}_j = \text{EKF\_update}(\mathbf{m}_j, \mathbf{z}_k, \mathbf{R}_k, \mathcal{DA}_k)$   
   $\mathbf{m}_j = \text{add\_new\_features}(\mathbf{m}_j, \mathbf{z}_k, \mathbf{R}_k, \mathcal{DA}_k)$   
  if  $\text{size}(\mathbf{m}_j) > \text{max\_features}$  then  
     $\mathbf{m}_{j+1} = \{\text{global}, \text{local}\}\text{-map\_transition}(\mathbf{m}_j)$   
     $j = j + 1$   
  end if  
end for  
for  $i = j$  to 2 do  
   $\mathbf{m}_{i-1} = \text{back\_propagation}(\mathbf{m}_i, \mathbf{m}_{i-1})$   
end for
```

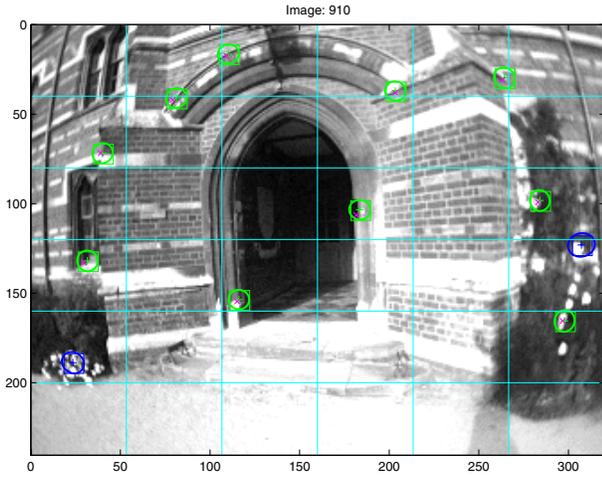


Fig. 3. A screenshot of salient features extracted during the trajectory

IV. EXPERIMENTS AND RESULTS

The algorithm proposed is tested using real data obtained from a large outdoor environment using a monocular camera at 30 fps. The features extracted from the images are 3D salient points. To initialize features the inverse depth algorithm is implemented [22]. Data association is carried out using the JCBB algorithm [23] which has been demonstrated to be well-suited for vision features [24].

The state vector to be estimated is given by:

$$\mathbf{x}^T = [\mathbf{x}_c^T, \mathbf{Y}^T] \quad (32)$$

where \mathbf{x}_c represents the camera state

$$\mathbf{x}_c^T = [\mathbf{r}^T, \Psi^T, \mathbf{v}^T, \omega^T] \quad (33)$$

and \mathbf{Y} the set of features in the map.

The camera state \mathbf{x}_c contains the position of the camera in cartesian coordinates \mathbf{r} , its attitude in Euler angles Ψ , the linear velocity \mathbf{v} and its angular velocity ω . With this

parametrization the movement of the camera in 6D can be well described. In the inverse depth approach each feature is parametrized with six components.

The characteristics of the experiment makes it suitable to show the benefits of sharing information between maps. Since a feature can be seen from different local maps and the time required to initialize it can last for several frames, the technique proposed turns to be very useful as allow us to reuse a feature without having to re-initialize it in each local map. In addition, linear and angular velocities of the camera, \mathbf{v} and ω , have to be shared as well since otherwise the scale of the maps could change dramatically, as was observed in [24].

The path followed with the hand-held camera is a large L-shaped trajectory where approximately 1000 salient features are extracted and tracked from the building facade. Fig(3) shows one of the images obtained in the experiment and the corresponding features extracted.

Global and local map references have been implemented for comparison purposes. In both implementations the maximum number of features per map has been limited to 50 and the total number of local maps created is 10. Fig(4) presents the results obtained. As it was expected both algorithms reduce the computational cost comparing with the standard EKF by dividing the global map into several smaller local maps. It can be noticed as well that the implementation based on local references achieves better consistency results and as a consequence a better estimate of the map and the trajectory. This is noticeable in the larger size of ellipsoids for the local reference implementation and in the better estimation of the orthogonal angle between building walls. The ellipsoids show the final position of the robot in each local map. The trajectory is shown only for reference purposes but it is not estimated in the state vector which explains the non-coincidence with the final estimate of camera positions after *Back-propagation*.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a new technique that allows the use of local maps algorithms but avoiding the constraint of probabilistic independence between them. By means of this procedure, salient features of the environment or vehicle state components as velocity or global attitude, can be shared between local maps in a consistent manner what can be extremely valuable to reduce the errors committed during the first steps of map initialization, specially for monocular vision.

The technique presents some similitude with the Treemap algorithm [13] in the sense that flows of information are transferred between submaps to update previous map estimates. However, our technique uses the covariance form instead of the information form, which allow us to apply directly effective data association algorithms such as JCBB. The second crucial difference is the use of local coordinates which improves precision, as shown in our experiments. Finally, our technique represents the information using sequential local maps instead of ordering features in a tree

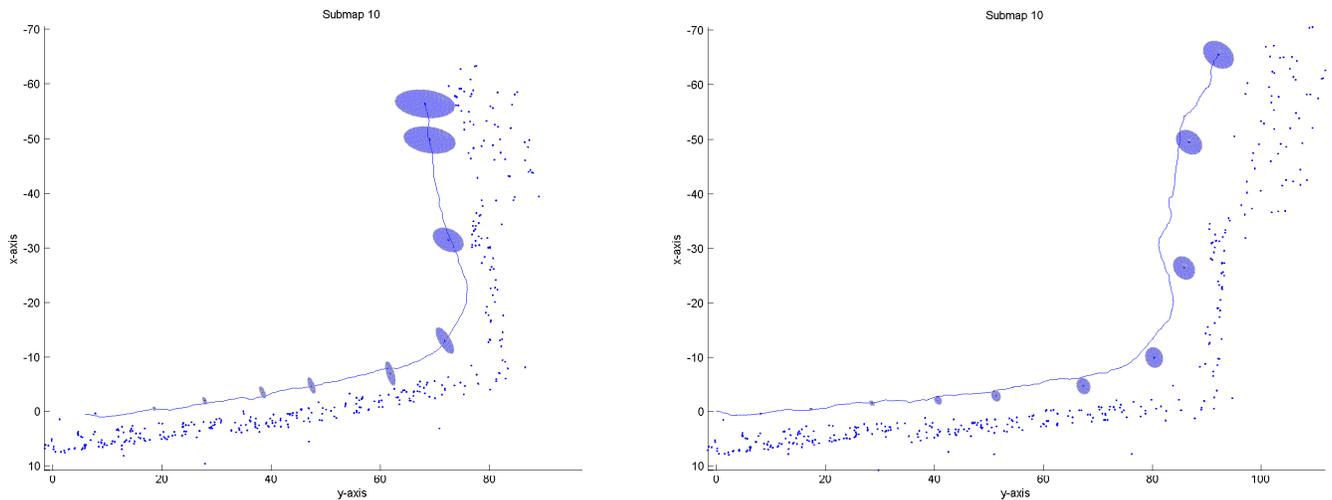


Fig. 4. Final 3D maps obtained using the algorithm proposed with local and global references for submaps. Left plot shows the result of the local reference implementation from a top perspective. Right plot shows the result but using the global implementation. Clearly the estimate achieved with local references is better and more consistent.

structure which has to be maintained and balanced, resulting in an algorithm which is easier to implement.

The *Back-propagation* algorithm allows us to make update steps from local map to local map in constant time which suggest a loop-closing algorithm that could be linear in the number of local maps which will be studied in future research.

Future work will also include the implementation of the technique in situations where inertial (IMU) information is available, in addition, techniques to perform and detect loop closing will be analyzed.

VI. ACKNOWLEDGMENTS

This research has been funded in part by the Dirección General de Investigación of Spain under project DPI2006-13578.

REFERENCES

- [1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, September 2005.
- [2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [3] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.
- [4] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *Robotics Research, The Fourth Int. Symposium*, O. Faugeras and G. Giralt, Eds. The MIT Press, 1988, pp. 467–474.
- [5] J. A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardós, "The SPmap: A probabilistic framework for simultaneous localization and map building," *IEEE Trans. Robotics and Automation*, vol. 15, no. 5, pp. 948–953, 1999.
- [6] S. J. Julier and J. K. Uhlmann, "A Counter Example to the Theory of Simultaneous Localization and Map Building," in *Proc. IEEE Int. Conf. Robotics and Automation*, Seoul, Korea, 2001, pp. 4238–4243.
- [7] J. A. Castellanos, J. Neira, and J. D. Tardós, "Limits to the consistency of EKF-based SLAM," in *5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, 2004.
- [8] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Trans. Robotics and Automation*, vol. 17, no. 3, pp. 242–257, 2001.
- [9] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous Localization and Mapping with Sparse Extended Information Filters," *Int. J. Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.
- [10] M. A. Paskin, "Thin Junction Tree Filters for Simultaneous Localization and Mapping," in *Proc. Int. Joint Conf. Artificial Intelligence*, San Francisco, CA., 2003, pp. 1157–1164.
- [11] M. R. Walter, R. M. Eustice, and J. J. Leonard, "Exactly sparse extended information filters for feature-based SLAM," *Int. J. Robotics Research*, vol. 26, no. 4, pp. 335–359, 2007.
- [12] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Trans. Robotics*, vol. 22, no. 6, pp. 1100–1114, Dec 2006.
- [13] U. Frese, "Treemap: An $O(\log n)$ algorithm for indoor simultaneous localization and mapping," *Autonomous Robots*, vol. 21, no. 2, pp. 103–122, September 2006.
- [14] S. Julier and J. Uhlmann, "A new extension of the Kalman Filter to nonlinear systems," in *International Symposium on Aerospace/Defense Sensing, Simulate and Controls*, Orlando, FL, 1997.
- [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*, vol. 21, no. 4, pp. 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*, vol. 1, Washington DC, 2002, pp. 406–411.
- [17] 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, August 2005.
- [18] L. M. Paz, P. Jensfelt, J. D. Tardós, and J. Neira, "EKF SLAM updates in $O(n)$ with Divide and Conquer SLAM," in *Proc. IEEE Int. Conf. Robotics and Automation*, Rome, Italy, April 2007.
- [19] A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 2002.
- [20] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: John Wiley and Sons, 2001.
- [21] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [22] J. M. M. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular SLAM," in *Proc. Robotics: Science and Systems*, Philadelphia, USA, August 2006.
- [23] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.
- [24] L. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardós, "Mapping large loops with a single hand-held camera," in *Proc. Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.