

Consistent Data Association in Multi-robot Systems with Limited Communications

Rosario Aragués Eduardo Montijano Carlos Sagués

Departamento de Informática e Ingeniería de Sistemas, Instituto de Investigación en Ingeniería de Aragón,
Universidad de Zaragoza, María de Luna, 50018 Zaragoza, Spain
raragues@unizar.es, emonti@unizar.es, csagues@unizar.es

Abstract—In this paper we address the data association problem of features observed by a robot team with limited communications. At every time instant, each robot can only exchange data with a subset of the robots, its neighbors. Initially, each robot solves a local data association with each of its neighbors. After that, the robots execute the proposed algorithm to agree on a data association between all their local observations which is globally consistent. One inconsistency appears when chains of local associations give rise to two features from one robot being associated among them. The contribution of this work is the decentralized detection and resolution of these inconsistencies. We provide a fully decentralized solution to the problem. This solution does not rely on any particular communication topology. Every robot plays the same role, making the system robust to individual failures. Information is exchanged exclusively between neighbors. In a finite number of iterations, the algorithm finishes with a data association which is free of inconsistent associations. In the experiments, we show the performance of the algorithm under two scenarios. In the first one, we apply the resolution and detection algorithm for a set of stochastic visual maps. In the second, we solve the feature matching between a set of images taken by a robotic team.

I. INTRODUCTION

During the last years, there has been an increasing interest in multi robot systems. In these scenarios, a team of robots cooperatively perform some task in a more efficient way than a single robot would do. In this paper, we address the data association problem in multi robot scenarios. It consists of establishing correspondences between different measurements or estimates of a common element. It is of high interest in localization, mapping, exploration, and tracking applications [1]. The Nearest Neighbor (NN), and the Maximum Likelihood (ML), are widely used methods which associate each observation with its closest feature in terms of the Euclidean or the Mahalanobis distance [23], [15], [12]. Other popular method is the Joint Compatibility Branch and Bound (JCBB) [18], which considers the compatibility of many associations simultaneously. The Combined Constraint Data Association [2] builds a graph where the nodes are individually compatible associations and the edges relate binary compatible assignments. Over this graph, a Maximal Common Subgraph problem is solved for finding the maximum clique in the graph. Scan matching and Iterative Closest Point (ICP) [6] are popular methods for comparing two laser scans. Other methods, like the Multiple Hypothesis Tracking, and the Joint Probabilistic Data Association, maintain many association

hypothesis instead of selecting one of them. And exist many variations of these techniques that combine RANSAC [9] for higher robustness.

In solutions based on submaps, one of them is usually transformed into an observation of another. The local submaps are merged with the global map following a sequence [22], or in a hierarchical binary tree fashion [5]. All the mentioned data association approaches, operate on elements from two sets. One set usually contains the current observations, and the other one consists of the feature estimates. These sets may be two images, two laser scans, or two probabilistic maps.

Lately, many localization, mapping, and exploration algorithms for multi robot systems have been presented. However, they have not fully addressed the problem of multi robot data association. Some solutions have been presented for merging two maps [21], [23] that do not consider a higher number of robots. Many approaches rely on broadcasting all controls and observations measured by the robots. Then, the data association is solved like in a single robot scenario, using scan matching and ICP for laser scans [14], [10], [16], [20], or NN, ML, and visual methods for feature-based maps [12], [17]. In these methods, the problem of inconsistent data associations is avoided by forcing a cycle-free merging order. This limitation has also been detected in the computer vision literature. In [8] they approach an inconsistent association problem for identifying equal regions in different views. They consider a centralized scenario, where each 2 views are compared among them in a 2-by-2 way. Then, their results are arranged on a graph where associations are propagated and conflicts are solved. The work in [7], from the target tracking literature, simultaneously considers the association of all local maps. It uses an expectation-maximization method for both computing the data association and the final global map. The main limitation of this work is that the data from all sensors needs to be processed together, what implies a centralized scheme, or a broadcast method.

All the previous methods rely on centralized schemes, full communication between the robots, or broadcasting methods. However, in multi robot systems, distributed approaches are more interesting. They present a natural robustness to individual failures since there are no central nodes. Besides, they do not rely on any particular communication scheme, and they are robust to changes in the topology. On the other hand, distributed algorithms introduce an additional level of

complexity in the algorithm design. Although the robots make decisions based on their local data, the system must exhibit a global behavior.

In this paper, we address the data association problem for distributed robot systems. Each of our robots possesses a local observation of the environment. Instead of forcing a specific order for associating their observations, we allow the robots to compute its data association with each of its neighbors in the graph. Although this scenario is more flexible, it may lead to inconsistent global data associations in the presence of cycles in the communication graph. These inconsistencies are detected when chains of local associations give rise to two features from one robot being associated among them. These situations must be correctly identified and solved before merging the data. Otherwise, the merging process would be wrong and could not be undone. In this paper, we approach a distributed data association, under limited communications. Instead of comparing any 2 local observations among them, only the local observations of neighboring robots can be compared. Besides, there is no central agent that has knowledge of all the local associations and each robot exclusively knows the associations computed by itself. Then, each robot updates its local information by communicating with its neighbors. We present an algorithm where, finally, each robot is capable of detecting and solving any inconsistent association that involves any of its features.

II. PROBLEM DESCRIPTION

We consider a robotic team composed of $n \in \mathbb{N}$ robots. The n robots have communication capabilities to exchange information with the other robots. However, these communications are limited. Let $\mathcal{G}_{com} = (\mathcal{R}_{com}, \mathcal{E}_{com})$ be the undirected communication graph. The nodes are the robots, $\mathcal{R}_{com} = \{1, \dots, n\}$. If two robots i, j can exchange information then there is an edge between them, $(i, j) \in \mathcal{E}_{com}$. Let N_i be the set of neighbors of robot i ,

$$N_i = \{j \mid (i, j) \in \mathcal{E}_{com}\}.$$

Each robot i has observed a set \mathcal{S}_i of m_i features,

$$\mathcal{S}_i = \{f_1^i, \dots, f_{m_i}^i\}. \quad (1)$$

It can compute the data association between its own set \mathcal{S}_i and the sets of its neighbors \mathcal{S}_j , with $j \in N_i$. However, these data associations are not perfect. There may appear inconsistent data associations relating different features from the same set \mathcal{S}_i (Fig. 1). If the robots merge their data as soon as they solve the local data association, inconsistent associations cannot be managed since the merging cannot be undone. The goal of our algorithm is to detect and resolve these inconsistent associations before executing the merging.

In order to make the reading easy, along the paper we will use the indices i, j and k to refer to robots and indices r, r', s, s' , to refer to features. The r^{th} feature observed by the i^{th} robot will be denoted as f_r^i . Given a matrix A , the notation $[A]_{r,s}$ will correspond to the component (r, s) of the matrix

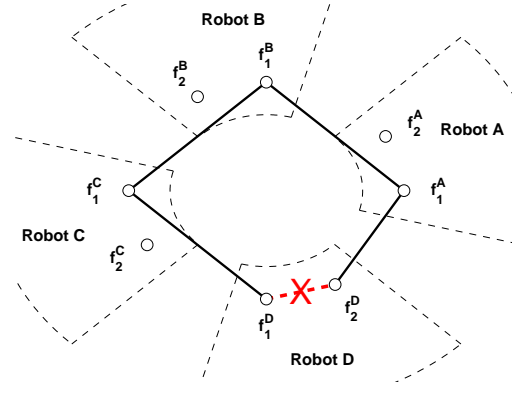


Fig. 1. Robots A, B, C and D associate their features comparing their maps in a two-by-two way. Robot A associates its feature f_1^A with f_2^D and with f_1^B ; robot B associates f_1^B with f_1^C ; robot C associates f_1^C with f_1^D (solid lines). As a result, there is a path (dashed line) between f_1^D and f_2^D . This is an inconsistent situation. Finding this path would require the knowledge of the whole association graph.

whereas A_{ij} will denote the block (i, j) when the matrix is defined by blocks.

Let F be a function that computes the data association between any two sets of features, \mathcal{S}_i and \mathcal{S}_j , and returns an association matrix $A_{ij} \in \mathbb{N}^{m_i \times m_j}$ where

$$[A_{ij}]_{r,s} = \begin{cases} 1 & \text{if } f_r^i \text{ and } f_s^j \text{ are associated,} \\ 0 & \text{otherwise,} \end{cases}$$

for $r = 1, \dots, m_i$ and $s = 1, \dots, m_j$. We assume that F satisfies the following conditions.

Assumption 2.1: When F is applied to the same set \mathcal{S}_i , it returns the identity, $F(\mathcal{S}_i, \mathcal{S}_i) = A_{ii} = I$.

Assumption 2.2: The returned association A_{ij} has the property that the features are associated in a one-to-one way,

$$\sum_{r=1}^{m_i} [A_{ij}]_{r,s} \leq 1 \text{ and } \sum_{s=1}^{m_j} [A_{ij}]_{r,s} \leq 1,$$

for all $r = 1, \dots, m_i$ and $s = 1, \dots, m_j$.

Assumption 2.3: Robots i and j associate their features in the same way. Given two sets \mathcal{S}_i and \mathcal{S}_j it holds that $F(\mathcal{S}_i, \mathcal{S}_j) = A_{ij} = A_{ji}^T = (F(\mathcal{S}_j, \mathcal{S}_i))^T$.

Most of the existing data association methods satisfy these assumptions.

If F is applied to all pairs of sets of features $\mathcal{S}_i, \mathcal{S}_j$, for $i, j \in \{1, \dots, n\}$, its results can be represented by an undirected graph $G_{map} = (V_{map}, E_{map})$. Each node in V_{map} is a feature f_r^i , for $i = 1, \dots, n, r = 1, \dots, m_i$. There is an edge between two features f_r^i, f_s^j iff $[A_{ij}]_{r,s} = 1$. If F were providing the ground truth data association, then G_{map}^{gt} would exclusively contain disjoint cliques, identifying features observed by multiple robots [8]. Since F is not perfect, G_{map} is a perturbed version of G_{map}^{gt} that includes additional spurious edges, while it misses others.

In addition, due to communication restrictions, the data associations available to our robots are just a subset of G_{map} . We name $G = (V, E)$ this available association graph. It has

the same set of nodes than G_{map} , $V = V_{map}$. It has an edge between two features f_r^i , f_s^j only if the edge exists in G_{map} and the robots i and j are neighbors in the communication graph. The goal of our algorithm is to detect and resolve inconsistencies in this graph $G \subseteq G_{map}$ in a decentralized fashion.

Definition 2.4: An *association set* is a set of features such that they form a connected component in G . Such set is an *inconsistent association* or a *conflictive set* if there exists a path in G between two or more features from the same robot. A feature is *inconsistent* or *conflictive* if it belongs to an inconsistent association.

Along this paper, we name m_{sum} the number of features in G , $m_{sum} = \sum_{i=1}^n m_i$. We name d the diameter of G , the length of the longest path between any two nodes in G . The diameter satisfies $d \leq m_{sum}$. We name $W \in \mathbb{N}^{m_{sum} \times m_{sum}}$ the adjacency matrix of G ,

$$W = \begin{bmatrix} W_{11} & \cdots & W_{1n} \\ \vdots & \ddots & \vdots \\ W_{n1} & \cdots & W_{nn} \end{bmatrix}, \quad (2)$$

where W_{ij} is the block within W for the associations between robots i and j ,

$$W_{ij} = \begin{cases} A_{ij} & \text{if } j \in \{N_i \cup i\}, \\ \mathbf{0} & \text{otherwise.} \end{cases} \quad (3)$$

III. DECENTRALIZED DETECTION OF INCONSISTENT ASSOCIATIONS

Considering definition 2.4 we observe that in order to detect an inconsistent association it is required to compute the paths that exist among the elements in G . As the following lemma states [4], given a graph G , the powers of its adjacency matrix contains the information about the number of paths existing between the nodes of G :

Lemma 3.1 (Lemma 1.32 [4]): Let G be a weighted graph of order $|V|$ with un-weighted adjacency matrix $W \in \{0, 1\}^{|V| \times |V|}$, and possibly with self loops. For all $i, j \in \{1, \dots, |V|\}$ and $t \in \mathbb{N}$ the (i, j) entry of the t^{th} power of W , W^t , equals the number of paths of length t (including paths with self loops) from node i to node j .

The computation of the powers of W requires, a priori, the information about the whole matrix. We show now that this computation can also be done in a decentralized manner. Let each robot $i \in \mathcal{R}_{com}$ maintain the blocks within W^t associated to its own features, $X_{ij}(t) \in \mathbb{N}^{m_i \times m_j}$, $j = 1, \dots, n$, $t \geq 0$, which are initialized as

$$X_{ij}(0) = \begin{cases} I, & j = i, \\ \mathbf{0}, & j \neq i, \end{cases} \quad (4)$$

and are updated, at each time step, with the following algorithm

$$X_{ij}(t+1) = \sum_{k \in \{N_i \cup i\}} W_{ik} X_{kj}(t), \quad (5)$$

with W_{ik} as defined in (3). It is observed that the algorithm is fully distributed because the nodes only use information about its direct neighbors in the communication graph.

Theorem 3.2: Let $[W^t]_{ij} \in \mathbb{N}^{m_i \times m_j}$ be the block within W^t related to the associations between robot i and robot j . The matrices $X_{ij}(t)$ computed by each robot i using the decentralized algorithm (5) are exactly the sub-matrices $[W^t]_{ij}$,

$$X_{ij}(t) = [W^t]_{ij}, \quad (6)$$

for all $i, j \in \{1, \dots, n\}$ and all $t \in \mathbb{N}$.

Proof: The proof is done using induction. First we show that eq. (6) is satisfied for $t = 0$. In this case we have that $W^0 = I$, thus for all $i, j \in \{1, \dots, n\}$, $[W^0]_{ii} = I$ and $[W^0]_{ij} = \mathbf{0}$, which is exactly the initial value of the variables X_{ij} (eq. (4)).

Now we have that for any $t > 0$,

$$[W^t]_{ij} = \sum_{k=1}^n W_{ik} [W^{t-1}]_{kj} = \sum_{k \in \{N_i \cup i\}} W_{ik} [W^{t-1}]_{kj},$$

because $W_{ik} = \mathbf{0}$ for $k \notin \{N_i \cup i\}$,

Assuming that for all $i, j \in \{1, \dots, n\}$ and a given $t > 0$, $X_{ij}(t-1) = [W^{t-1}]_{ij}$ is true, then

$$\begin{aligned} X_{ij}(t) &= \sum_{k \in \{N_i \cup i\}} W_{ik} X_{kj}(t-1) = \\ &= \sum_{k \in \{N_i \cup i\}} W_{ik} [W^{t-1}]_{kj} = [W^t]_{ij}. \end{aligned} \quad (7)$$

Then, by induction, $X_{ij}(t) = [W^t]_{ij}$ is true for all $t > 0$. ■

Corollary 3.3: The variables $X_{ij}(t)$ contain the information about all the paths of length t between features observed by robots i and j .

Proof: By direct application of Lemma 3.1. ■

Analyzing the previous algorithm the first issue to deal with is how to simplify the computation of the matrices in order to avoid high powers of W . In the case we are studying it is just required to know if there is a path between two elements in G and not how many paths are. This means that in this situation it is enough that $[X_{ij}(t)]_{r,s} > 0$ in order to know that features f_r^i and f_s^j are connected by a path. Another issue is to decide when the algorithm in (5) must stop. Since the maximum length of a path between any two nodes in a graph is its diameter, then after d iterations the algorithm should stop. However, in general situations the robots will not know neither d nor m_{sum} , which makes this decision hard to be made a priori.

Definition 3.4: We will say that two matrices W and \bar{W} of the same dimensions are equivalent, $W \sim \bar{W}$, if for all r and s it holds

$$[W]_{r,s} > 0 \Leftrightarrow [\bar{W}]_{r,s} > 0 \text{ and } [W]_{r,s} = 0 \Leftrightarrow [\bar{W}]_{r,s} = 0.$$

In practice any equivalent matrix to the $X_{ij}(t)$ will provide the required information, which allows us to simplify the computations simply by changing any positive value in the matrices by 1. Moreover, the equivalency is also used to find a criterion to stop the algorithm:

Proposition 3.5: For a robot i , let t_i be the first time instant, t , such that $X_{ij}(t) \sim X_{ij}(t-1)$ for all $j = 1, \dots, n$. Then robot i can stop to execute the algorithm at time t_i .

Proof: Let $\bar{X}_{ij}(t)$ be the components in $X_{ij}(t)$, such that $[X_{ij}(t-1)]_{r,s} = 0$ and $[X_{ij}(t)]_{r,s} > 0$. The cardinal, $|\bar{X}_{ij}(t)|$, represents the number of features $f_s^j \in \mathcal{S}_j$ such that the minimum path length in G between them and one feature $f_r^i \in \mathcal{S}_i$ is t . At time t_i , $X_{ij}(t_i) \sim X_{ij}(t_i - 1) \quad \forall j$ for the first time, and then $\sum_{j=1}^n |\bar{X}_{ij}(t_i)| = 0$ because no component has changed its value from zero to a positive. This means that there is no path of minimum distance t_i linking any feature f_r^i with any other feature in G . By the physical properties of a path, it is obvious that if there are no features at minimum distance t_i , it will be impossible that a feature is at minimum distance $t_i + 1$ and all the paths that connect features of robot i with any other feature have been found. ■

Corollary 3.6: All the robots end the execution of the iteration rule (5) in at most in $d + 1$ iterations.

Proof: Recalling that the maximum distance between two nodes in G is the diameter of the graph, denoted by d , then $\sum_{j=1}^n |\bar{X}_{ij}(d+1)| = 0$ for all $i = 1, \dots, n$. ■

If a robot j at time t does not receive the information $X_{ij}(t)$ from robot i then it will use the last matrix received, because robot i has already finished computing its paths and $X_{ij}(t) \sim X_{ij}(t-1)$.

It remains to analyze which features are conflictive and which are not. Each robot has the information about all the association paths of its features and the features of the rest of the robots in the network in the different variables $X_{ij}(t_i)$. The robots detect all the conflictive features using two simple rules. A feature f_r^i is conflictive if and only if one of the following conditions are satisfied:

- (i) There exists other feature $f_{r'}^i$, with $r \neq r'$, such that

$$[X_{ii}(t_i)]_{r,r'} > 0; \quad (8)$$

- (ii) There exist features f_s^j and $f_{s'}^j$, $s \neq s'$, such that

$$[X_{ij}(t_i)]_{r,s} > 0 \text{ and } [X_{ij}(t_i)]_{r,s'} > 0. \quad (9)$$

In conclusion, the proposed algorithm will be able to find all the inconsistencies in a finite number of iterations. The algorithm is decentralized and it is based only on local interactions between the robots. Each robot only needs to know its local data associations. It updates its information based on the data exchanged with its neighbors. When the algorithm finishes, each robot i can extract from its own matrices $X_{ij}(t_i)$ all the information of any conflict that involves any of its features. If the robot has any conflictive feature, it also knows the rest of features that belong to the conflictive set independently of the robot that observed such features. In the next section we present an additional technique to solve the inconsistencies found by the algorithm.

IV. DECENTRALIZED RESOLUTION OF INCONSISTENT ASSOCIATIONS

The resolution of inconsistent associations consists of deleting edges from G so that the resulting graph is conflict-free.

Definition 4.1: Let c denote the number of conflictive sets in G . We say a conflictive set C is *detectable* by a robot i

if there exists a $r \in \{1, \dots, m_i\}$ such that $f_r^i \in C$. The set of robots that detect a conflictive set C is $R \subseteq \mathcal{R}_{com}$. The number of features from each robot $i \in R$ involved in C is \tilde{m}_i . We say G is *conflict-free* if $c = 0$.

All the edges whose deletion transforms G into a conflict-free graph, belong to any of the c conflictive sets of G . Since the conflictive sets are disjoint, they can be considered separately. From now on, we focus on the resolution of one of the conflictive sets C . The other conflictive sets are managed in the same way. The resolution problem consists of partitioning C into a set of disjoint conflict-free components C_q such that

$$\cup_q C_q = C, \text{ and } C_q \cap C_{q'} = \emptyset,$$

for all $q, q' = 1, 2, \dots$. The number of such conflict-free components is a priori unknown and it will be discussed later in this section.

Obtaining an optimal partition that minimizes the number of deleted edges is complicated. If there were only two inconsistent features $f_r^i, f_{r'}^i$, it could be approached as a max-flow min-cut problem [19]. However, in general there will be more inconsistent features, $\tilde{m}_i \geq 2$, within C associated to a robot $i \in R$. Besides, there may also be $\tilde{m}_j \geq 2$ inconsistent features belonging to a different robot $j \in R$. The application of [19] separately to any pair of inconsistent features does not necessarily produce an optimal partition. It may happen that a single edge deletion simultaneously resolves more than one inconsistent association. Therefore, an optimal solution should consider multiple combinations of edge deletions, what makes the problem computationally intractable, and imposes a centralized scheme. We propose a resolution algorithm that is not optimal but is efficient and is proven to be correct. Besides, it allows a decentralized computation.

Proposition 4.2: Let R be the set of robots that detect C . Let i_* be the robot with the most features involved in C ,

$$i_* = \arg \max_{i \in R} \tilde{m}_i. \quad (10)$$

The number of conflict-free components in which C can be decomposed is lower bounded by \tilde{m}_{i_*} .

Proof: Each conflict-free component can contain, at most, one feature from a robot $i \in R$. Then there must be, at least, $\max_{i \in R} \tilde{m}_i = \tilde{m}_{i_*}$ components. ■

The resolution algorithm constructs \tilde{m}_{i_*} conflict-free components using a strategy close to a BFS tree construction. Initially, each robot i detects the conflictive sets for which it is the root using its local information $X_{i1}(t_i), \dots, X_{in}(t_i)$. The root robot for a conflictive set is the one with the most inconsistent features involved. In case two robots have the same number of inconsistent features, the one with the lowest robot id is selected. Then, each robot executes the resolution algorithm (Algorithm 1).

The root robot creates \tilde{m}_{i_*} components and initializes each component C_q with one of its features $f^{i_*} \in C$. Then, it tries to add to each component C_q the features directly associated to $f^{i_*} \in C_q$. Let us consider that f_s^j has been assigned to C_q . For all f_r^i such that $[W_{ij}]_{r,s} = 1$, robot j sends a component

Algorithm 1 Decentralized resolution algorithm - Robot i

```
1: – Initialization
2: for each conflictive set  $C$  for which  $i$  is root ( $i = i_*$ ) do
3:   create  $\tilde{m}_{i_*}$  components
4:   assign each inconsistent feature  $f_r^{i_*} \in C$  to a different
     component  $C_q$ 
5:   send component request to all its neighboring features
6: end for
7: – Algorithm
8: for each component request from  $f_s^j$  to  $f_r^i$  do
9:   if (b) or (c) then
10:     $[W_{ij}]_{r,s} = 0$ 
11:    send reject message to  $j$ 
12:   else if (d) then
13:    assign  $f_r^i$  to the component
14:    send component request to all its neighboring features
15:   end if
16: end for
17: for each component reject from  $f_s^j$  to  $f_r^i$  do
18:    $[W_{ij}]_{r,s} = 0$ 
19: end for
```

request message to robot i . When robot i receives it, it may happen that

- (a) f_r^i is already assigned to C_q ;
- (b) f_r^i is assigned to a different component;
- (c) other feature $f_{r'}^i$ is already assigned to C_q ;
- (d) f_r^i is unassigned and no feature in i is assigned to C_q .

In case (a), f_r^i already belongs to the component C_q and robot i does nothing. In cases (b) and (c), f_r^i cannot be added to C_q ; robot i deletes the edge $[W_{ij}]_{r,s}$ and replies with a reject message to robot j ; when j receives the reject message, it deletes the equivalent edge $[W_{ji}]_{s,r}$. In case (d), robot i assigns its feature f_r^i to the component C_q and the process is repeated.

Theorem 4.3: Let us consider that each robot $i \in \mathcal{R}_{com}$ executes the decentralized resolution algorithm (Algorithm 1) on G , obtaining G' ,

- (i) after $t = n$ iterations no new features are added to any component C_q and the algorithm finishes;
- (ii) each obtained C_q is a connected component in G' ;
- (iii) C_q is conflict free;
- (iv) C_q contains at least two features;

for all $q \in \{1, \dots, \tilde{m}_{i_*}\}$ and all conflictive sets.

Proof: (i) The maximal depth of a conflict-free component is n since, if there were more features, at least two of them would belong to the same robot. Then, after at most n iterations of this algorithm, no more features are added to any component C_q and the algorithm finishes.

(ii) There is a path in G between any two features belonging to a conflictive set C . Therefore, there is also a path in G between any two features assigned to the same component C_q . Since the algorithm does not delete edges from G within a component (case (a)), then C_q is also connected in G' . Since

none feature can be assigned to more than one component (case (b)), the components are disjoint. Therefore, C_q is a connected component in G' .

(iii) By construction, two features from the same robot are never assigned to the same component C_q (case (c)). Therefore, each component is conflict-free.

(iv) Each conflictive set has more than one feature. Because of Assumptions 2.1 and 2.2, each feature and its neighbors are conflict free. Therefore, each component C_q contains, at least, its originating feature, and a neighboring feature. Thus, it has at least two features. ■

Corollary 4.4: After executing Algorithm 1, the size of each conflict set C is reduced by at least $2 \tilde{m}_{i_*}$, where $\tilde{m}_{i_*} \geq 2$.

When the algorithm finishes, each original conflictive set C has been partitioned into \tilde{m}_{i_*} conflict-free components. It may happen that a subset of features remains unassigned. These features may still be conflictive in G' . The detection algorithm can be executed on the subgraph defined by this smaller subset of features.

Proposition 4.5: Consider each robot i iteratively executes the detection (Section III) and the resolution (Section IV) algorithms. Then, in a finite number of iterations, all conflictive sets disappear.

Proof: After each execution of the resolution algorithm, the size of each conflict set C is reduced by, at least, $2 \tilde{m}_{i_*} \geq 4$ (Corollary 4.4). Then, in a finite number of iterations, it happens that $|C| < 4$. A set with 3 features $f_r^i, f_{r'}^i, f_s^j$ cannot be conflictive; this would require the existence of edges (f_r^i, f_s^j) and $(f_{r'}^i, f_s^j)$, what is impossible (Assumption 2.2). A set with 2 features cannot be conflictive (Assumptions 2.1 and 2.2), and a set with a single feature cannot be inconsistent by definition. Therefore, there will be no remaining inconsistencies or conflictive sets. ■

The main interest of the presented resolution algorithm, is that it is fully decentralized and it works on local information. Each robot uses its own $X_{ij}(t_i)$ for detecting the root robot of each conflictive set. During the resolution algorithm, the decisions and actions taken by each robot are based on its local associations W_{ij} , and the components assigned to its local features. Moreover, each robot is responsible of deleting the edges from its local association matrices W_{ij} , with $j \in \{1, \dots, n\}$. In addition, the presented algorithm works in finite time. Let us note that although we presented the algorithm for a single conflictive set, all conflictive sets are managed in parallel.

V. EXPERIMENTS

A. Data association of stochastic maps

The presented algorithm is of high interest in multi-robot exploration scenarios with limited communications. Each robot has explored a section of the environment and it has built a stochastic map. When the exploration finishes, the local maps are merged into a global map of the environment. Each robot can compute the data association between its local map, and the maps of its neighbors. However, it does not know the data

associations obtained by other robots in the team. If they start the merging process using only its local associations, and there is any inconsistency, at some point a robot will be forced to fuse two or more of its features into a single one. To avoid this situation, they execute the presented algorithm for resolving any inconsistent association before merging the maps.

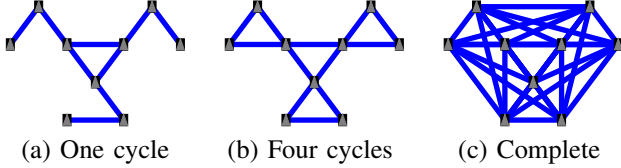


Fig. 2. Communication graphs between the 9 robots after the exploration.

The behavior of the algorithm is analyzed under real data. We use a data set [11] with bearing information obtained with vision (Sony EVI-371DG). The landmarks are vertical lines extracted from the images. The measurements are labeled so that we can compare our results with the ground-truth data association. We select 9 subsections of the whole path for the operation of 9 different robots (Fig. 3 (a)). A separate SLAM is executed on each subsection, producing the 9 local maps (Fig. 3 (b)). The local data associations are computed using the JCBB [18] since it is very convenient for clutter situations like the considered scenario (Fig. 3 (b)). The JCBB is applied to the local maps of any pair of neighboring robots. We analyze the performance of the algorithm under 3 communication graphs (Fig. 2).

Table I gives statistics about the number of inconsistencies found considering the different network topologies in Fig. 2. We show the obtained associations compared to the ground truth results. The number of association sets is the number of connected components of W^t . The number of good links (true positives) are obtained associations between 2 features which are true (ground truth). The missing links (false negatives) are associations that are in the ground truth information, but have been not detected. And spurious links (false positives) are associations found between features that are different according to the ground truth. The sixth row, c , is the number of conflictive sets. The next row in the table shows the total number of features which have been associated to any other feature from other local map. The last row gives information about how many of those features are conflictive. The amount

TABLE I
INITIAL ASSOCIATIONS BETWEEN THE 9 LOCAL MAPS

Comm. graph	(a)	(b)	(c)
Association sets (ground truth)	242	284	400
Association sets	182	218	290
Good links (true positives)	160	190	228
Missing links (false negatives)	82	94	172
Spurious links (false positives)	22	28	62
Conflictive sets (c)	3	5	8
Number of features m_{sum}	138	144	154
Conflictive features	16	24	51

of missing and spurious associations obtained is very high

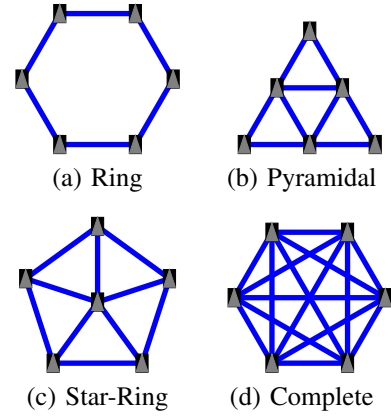


Fig. 4. Formations used in the experiments

for the three network topologies. This is the expected result for many real scenarios, where the landmarks are close to each other, and where the only available information are their cartesian coordinates. As a result, the conflictive features are more than a 10% of the total. In communication graphs with more cycles (Fig. 2 (b)(c)), there are more conflictive features. In the three cases, after a single execution of the detection and the resolution algorithms, all the inconsistencies are solved (Table II, 1st row). An interesting result is that, although our algorithm cannot distinguish between good and spurious edges, in practice a high number of the deleted edges (last row) are spurious ones.

TABLE II
MANAGEMENT OF THE INCONSISTENCIES

Comm. graph	(a)	(b)	(c)
Iterations	1	1	1
Initial conflictive sets	3	5	8
Deleted links	6	10	34
Good deleted links (true positives)	2	2	12
Spurious deleted links (false positives)	4	8	22

B. Matching sets of images

We have also tested the performance of our proposal with a set of images. A team of robots equipped with cameras and limited communication capabilities is a typical situation in which not all the images are available to execute a global matching. A solid set of matches is required independently of the task the team is performing (visual SLAM, formation control, etc.). In the proposed experiment 6 robots moving in formation are considered. Each robot acquires one image with its onboard camera and extracts SURF features [3]. The local matching is only applied to pairs of images which are connected in the communication graph. For the local matching the epipolar constraint combined with RANSAC is imposed [13]. The detection and resolution of inconsistencies is analyzed for four different typical communication graphs (Fig. 4).

Although the epipolar constraint discards most part of the wrong matches, some spurious associations are still found.

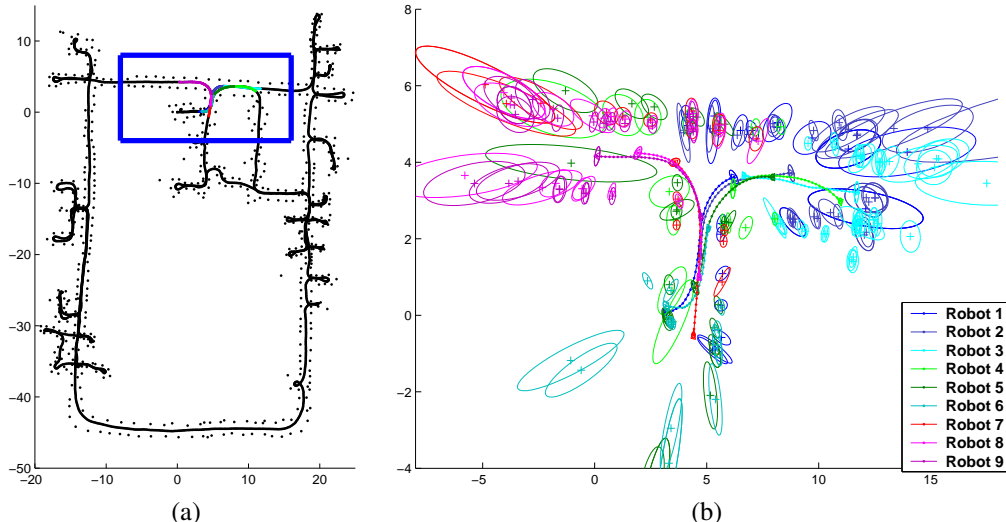


Fig. 3. (a) Section of the dataset used in the experiments. (b) Local maps acquired by 9 robots exploring the region in (a).

Figure 5 shows an example of an inconsistency found by our algorithm in the case of the formation (d). The figure also shows how the Algorithm 1 solves such inconsistency; the result is a subset of the initial matches with more connected components than before but without any conflict.

More general results about the experiment can be seen in Table III. Since for this experiment there is no ground truth available, the number of missing and spurious links found in the local matching cannot be provided. Some interesting conclusions are extracted with the obtained results. First of all, the number of association sets and the number of features involved m_{sum} are increased with the number of edges in \mathcal{G}_{com} . The more local associations there are, the more matches the algorithm will have available. With respect to the inconsistencies, they grow with the number of cycles in \mathcal{G}_{com} because each cycle can generate inconsistencies independently of the rest of the communication network. The size of the cycles also influence on the conflicts, cycles of smaller length will cause more inconsistencies because the number of local associations required to find a conflict is also smaller.

TABLE III
INITIAL ASSOCIATIONS FOR THE DIFFERENT FORMATIONS

Formation	(a)	(b)	(c)	(d)
Association sets	528	590	605	632
Conflictive sets (c)	2	22	31	72
Number of features m_{sum}	1335	1643	1743	2069
Conflictive features	14	144	204	521

Another important aspect to analyze is how the resolution of the conflicts is performed. We have observed that some of the inconsistencies are caused by features that are very close to each other in one image and due to the RANSAC threshold the miss-match appears. These inconsistencies are introduced because of the extractor parameters but the errors they can cause are negligible. However, there are also some other inconsistencies which can generate problems to the

vision system because they include features from the same image which are far away from each other. Table IV shows the statistics on how are the solutions to the conflicts provided by our algorithm. Although ground truth is not available in this experiment it is easily observable when the inconsistent matches have been separated correctly in the used images and when not (Fig. 5). We have manually classified the conflicts as well solved and wrongly solved. We consider a conflict as well solved when it does not contain any spurious edge in any of the associations that the algorithm returns. The spurious and the good links that the algorithm deletes, also manually counted, are shown in fifth and sixth rows of table IV respectively.

TABLE IV
MANAGEMENT OF THE INCONSISTENCIES

Formation	(a)	(b)	(c)	(d)
Initial conflictive sets	2	22	31	72
Inconsistencies well solved	1	20	24	55
Inconsistencies wrongly solved	1	2	7	17
Deleted links	2	26	46	131
Good deleted links (true positives)	1	2	19	43
Spurious deleted links (false positives)	1	24	27	88

VI. CONCLUSIONS

We have presented a new technique to match several sets of features observed by a team of robots in a consistent way under limited communications. Local associations are found only within robots that are neighbors in the communication graph. After that a fully decentralized method to compute all the paths between local associations is carried out, allowing the robots to detect all the inconsistencies related with their observations. For every conflictive set detected, in a second step the method is able to delete local associations to break the conflict using only local communications. The whole method is proved to finish in a finite amount of time finding and solving all the inconsistent associations. Experimental results show the performance of the method in two different scenarios with

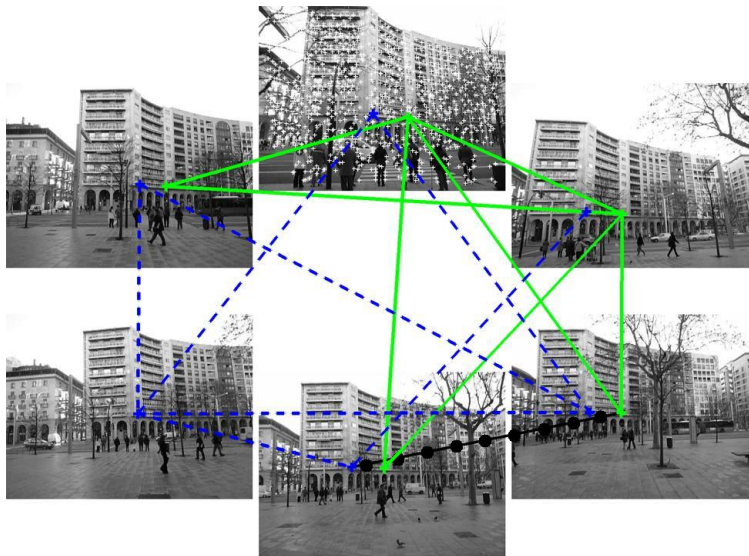


Fig. 5. Example of one inconsistency found by the algorithm. The inconsistency is represented by the whole set of depicted links. It is observed that if all the links are considered, features of the same image are matched. After executing Algorithm 1 the inconsistency is solved. In this example the root of the inconsistency is the top-middle image. The dark line with circles is the link deleted by the algorithm. Solid lines represent one of the conflict free components and dashed lines the second. For clarity, the rest of the SURF features are only shown in the top middle image.

great interest in robotic tasks. To the best of our knowledge this is the first algorithm that is able to do it.

ACKNOWLEDGMENT

This work was supported by project Ministerio de Ciencia e Innovacion DPI2009-08126, and grants MEC BES-2007-14772 and AP2007-03282. The data set used in one of the experiments was provided by U. Frese and J. Kurlbaum. The authors gratefully acknowledge them for providing data and support.

REFERENCES

- [1] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping: part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.
- [2] T. Bailey, E. M. Nebot, J. K. Rosenblatt, and H. F. Durrant-Whyte. Data association for mobile robot navigation: a graph theoretic approach. In *IEEE Int. Conf. on Robotics and Automation*, pages 2512–2517, San Francisco, USA, April 2000.
- [3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, 2006.
- [4] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009. Electronically available at <http://coordinationbook.info>.
- [5] C. Cadena, F. Ramos, and J. Neira. Efficient large scale SLAM including data association using the Combined Filter. In *European Conference on Mobile Robotics, ECMR*, pages 217–222, Mlini/Dubrovnik, Croatia, September 2009.
- [6] A. Censi. An accurate closed-form estimate of ICP’s covariance. In *IEEE Int. Conf. on Robotics and Automation*, pages 3167–3172, Roma, Italy, April 2007.
- [7] R. W. Deming and L. I. Perlovsky. Concurrent multi-target localization, data association, and navigation for a swarm of flying sensors. *Information Fusion*, 8(3):316 – 330, 2007.
- [8] V. Ferrari, T. Tuytelaars, and L. Van Gool. Wide-baseline multiple-view correspondences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 718–725, Madison, USA, June 2003.
- [9] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, pages 381–395, 1981.
- [10] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart. Distributed multirobot exploration and mapping. In *Proceedings of the IEEE*, volume 94, pages 1325–1339, July 2006.
- [11] U. Frese and J. Kurlbaum. A data set for data association, June 2008. <http://www.sfbtr8.spatial-cognition.de/insidedataassociation/>.
- [12] A. Gil, O. Reinoso, M. Ballesta, and M. Julia. Multi-robot visual SLAM using a rao-blackwellized particle filter. *Robotics and Autonomous Systems*, 58(1):68–80, 2009.
- [13] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2000.
- [14] A. Howard. Multi-robot simultaneous localization and mapping using particle filters. *International Journal of Robotics Research*, 25(12):1243–1256, 2006.
- [15] M. Kaess and F. Dellaert. Covariance recovery from a square root information matrix for data association. *Robotics and Autonomous Systems*, 57(12):1198 – 1210, 2009.
- [16] K. Konolige, J. Gutmann, and B. Limketkai. Distributed map-making. In *Workshop on Reasoning with Uncertainty in Robotics, Int. Joint Conf. on Artificial Intelligence*, Acapulco, Mexico, August 2003.
- [17] H. S. Lee and K. M. Lee. Multi-robot SLAM using ceiling vision. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 912–917, St. Louis, USA, October 2009.
- [18] J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.
- [19] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*, chapter 6.1 The Max-Flow, Min-Cut Theorem, pages 120–128. Dover, 1998.
- [20] M. Pfingsthorn, B. Slamet, and A. Visser. A scalable hybrid multi-robot SLAM method for highly detailed maps. In U. Visser, F. Ribeiro, T. Ohashi, and F. Dellaert, editors, *RoboCup 2007: Robot Soccer World Cup XI*, volume 5001 of *Lecture Notes in Artificial Intelligence*, pages 457–464. Springer Verlag, Berlin Heidelberg New York, 2008.
- [21] S. Thrun and Y. Liu. Multi-robot SLAM with sparse extended information filters. In *Int. Symposium of Robotics Research*, pages 254–266, Sienna, Italy, October 2003.
- [22] S. B. Williams and H. Durrant-Whyte. Towards multi-vehicle simultaneous localisation and mapping. In *IEEE Int. Conf. on Robotics and Automation*, pages 2743–2748, Washington, DC, USA, May 2002.
- [23] X. S. Zhou and S. I. Roumeliotis. Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1785–1792, Beijing, China, October 2006.