Distributed multi-camera visual mapping using topological maps of planar regions

E. Montijano*, C. Sagues

*Departamento de Informática e Ingeniería de Sistemas - Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain.

Abstract

This paper presents a multi-agent solution for cooperative visual mapping using planar regions. Each agent is assumed to be equipped with a conventional camera and has limited communication capabilities. Our approach starts building topological maps from independent image sequences where natural landmarks extracted from conventional images are grouped to create a graph of planes. With this approach the features observed in several images belonging to the same planar region are stored only once, reducing the size of the individual maps. In a distributed scenario this is very important because smaller maps can be transmitted faster, which makes our approach better suited for cooperative mapping. The later fusion of the individual maps is obtained via distributed consensus without any initial information about the relations between the different maps. Experiments with real images in complex scenarios show the good performance of our proposal.

Keywords: Computer Vision, Distributed Systems, Mapping

1. Introduction

Advances in communication technologies and vision systems have made feasible the idea of sets of cameras performing different perception tasks such as surveillance, tracking or mapping, in a cooperative way. In this paper we focus on the problem of map building using several cameras which communicate their perceived information to others. A set of cameras is able to map the environment faster than a single one would do. However, this adds new challenges that must be solved.

The problem of mapping the environment considering a single camera has been deeply studied, specially in the robotics community. The importance of creating and maintaining a map for localization and navigation tasks is obvious and a lot of effort has been made in this research line. A common approach is to simultaneously localize the camera and map the environment (SLAM) [1], [2]. In this approaches the map is usually represented by a set of 3D features, whose position is updated every time they are observed in a new image. To make this process more robust, view-based maps [3] introduce geometric constraints between pairs of images in the SLAM algorithm. Computing the structure of the scene, usually defined with the positions of the features and the cameras, makes the errors and drift grow with the size of the map. Our approach overcomes this limitation because we consider planar regions and no explicit metric information for the global map. A plane is defined as a set of features that belong to the same planar region in the scene. The errors in different planar regions are uncorrelated because the extraction of each one is independent of the others.

Topological maps [4] where no metric information is computed are another widely used approach to organize the visual information. Topological visual maps can be built from conventional [5], [6] or omnidirectional [7], [8] images. The whole image can be stored but usually only the extracted features are saved. Most recent and successful works use visual words [9] to represent the scene in a more compact way. Although topological maps based on images give good results, the space required to manage these maps is considerably large. If we store all the features, many of them will be seen in several images, so that the map will take up a lot of repeated data for the same features. By using planar regions, features are stored only once independently of the number of images in which they are observed. Moreover, the complexity of the maps is also reduced. Graphs made from raw images are usually dense because of the number of connections among close images whereas with the proposed maps the number of connections between planes is considerably smaller. Semantic meaning [10] in the maps is essential to human-robot interaction. Planes also provide a good semantic information meaningful for humans, making this choice more adequate than just features or raw images.

In addition to the previous advantages, it is well known that the structure estimation is improved in terms of accuracy and stability when considering the scene represented by planes [11]. Moreover, there are several works in the literature that assume the presence of planes in the environment to solve different tasks such as visual servoing.
[12], [13], [14], visual navigation with maps [15], [16], structure reconstruction [17] or camera calibration [18]. In all these approaches the 3D structure of the scene is not required and only sets of coplanar features in the image domain are used. Since the proposed solution based on planes follows these guidelines it can also be useful in the above situations.

Plane detection in images is a common problem in computer vision [19], [20], [21]. In order to detect the planes in the images we use a triple set, plane-image-image, for feature matching and homography computation; the planar regions are tracked along the sequence and new features are added to them once they are detected [22]. Homology constraints are used to detect new planes and also give a geometric criterion to relate them. Our approach is different to existing techniques to detect planes in images because previous works are usually more concerned about detecting planes in pairs of images instead of along sequences, as we do.

The extension of the mapping problem to a situation in which several cameras cooperatively explore the environment presents new challenges. Problems such as data association of multiple elements [23] [24], different intrinsic parameters of the cameras [25] or the fusion of the different maps [26] appear in this new scenario and new solutions should be given. Surveillance systems deal with the problem of sensor fusion using static cameras [27]. In [28], [29] the global map is managed by a centralized entity and the observations of the different cameras are introduced as they are captured. Distributed map merging is considered in [26] but data association is supposed to be given among the different maps. In the previous approaches the initial relative positions and the intrinsic parameters of the cameras are known. Our approach is independent of the relative positions of the cameras or their intrinsic parameters. We consider a set of individual maps created by different cameras with no a priori information about the relation between them. A distributed fusion based on consensus algorithms [30] is proposed. Distributed data association is done in the image domain considering invariant descriptors (e.g., SURF descriptors [31]) and homography constraints [32]. Our system is able to transform the initial local maps to similar maps which satisfy the conditions required to execute a decentralized fusion, achieving consensus.

As contributions of this paper, we can emphasize our proposal for detection of planar regions in image sequences through a set of homographies using a triple set, plane-image-image. We also propose an approach for building visual topological maps based on planar regions without requirements of metric information. Besides, a solution for a multi-camera scenario is proposed, giving a technique to fuse several maps based on distributed consensus without knowledge about the cameras, the positions or the data association.

The remainder of the paper is arranged as follows. The construction of a topological map using only one camera is explained in section 2. The extension of the problem to several cameras and map fusion using distributed consensus techniques is presented in section 3. In section 4 experimental results with real images illustrate the performance of the proposal. Finally, in section 5 some conclusions are given.

2. Single robot topological map

Let \( I = \{I_1, I_2, \ldots\} \) be the set of images acquired by a 6DOF camera mounted on a mobile robot. Let us assume that several rigid planar regions appear in the scene. There is no knowledge about either the internal parameters of the camera, represented by the calibration matrix \( K \), nor about the motion between consecutive frames, \( R \) and \( t \). For an easy understanding of the section, subscript indices will correspond to images in \( I \) whereas superscript indices will correspond to the planar regions, for example \( \pi_k^m \) will represent the features of the \( m^{th} \) planar region seen in the \( k^{th} \) image.

The topological map is managed using a graph, \( G = (\mathcal{P}, \mathbf{R}, \mathbf{A}) \), represented by a finite non empty set of planes \( \mathcal{P} \) with cardinality \( |\mathcal{P}| = P \), a vector \( \mathbf{R} \) containing the features observed in the planar regions and a matrix of relations between the planes \( \mathbf{A} \in \{0, 1\}^{P \times P} \). If \( \mathbf{A}(m, n) = 1 \) then there exists a relation between the planes \( m \) and \( n \), whereas for planes with no relation, \( \mathbf{A}(m, n) = 0 \).

If one plane, \( m \), is visible in two consecutive images of the sequence, \( I_k \) and \( I_{k+1} \), it is possible to compute a projective mapping (inter-image homography), \( H_{k,k+1}^m \), that relates the features belonging to the plane, \( \pi_k^m \), with features observed in the planar regions and a matrix of relations between the planes \( \mathbf{A} \in \{0, 1\}^{P \times P} \). If \( \mathbf{A}(m, n) = 1 \) then there exists a relation between the planes \( m \) and \( n \), whereas for planes with no relation, \( \mathbf{A}(m, n) = 0 \).

The planes are extracted from the sequence following the approach in [22]. The two initial frames of the sequence, \( I_1 \) and \( I_2 \), are picked up and all the planes seen in both images (Fig. 1-a) are extracted using DLT+RANSAC [33]. In order to perform the matching between features we have chosen SURF descriptors [31]. Nevertheless, the algorithm will work with any descriptor as long as it is able to extract enough features to compute homographies. The process is summarized as follows:

1. For any plane, \( m \), visible in the first two images:
   (a) All the features from the plane, \( \pi_k^m \), are stored expressed in the coordinates of the first image where they were detected. This first image is marked as the reference image of the plane, \( I_r \).
   (b) The identifiers of the features in every image are also stored to make automatic the future search of these features.
Every time a new plane is detected it is added to the topological map. One plane, \( m \), is formally added to the graph by

\[
\begin{align*}
\mathbf{A} &= [\mathbf{I}_P \mid \mathbf{0}_P]^{T} \mathbf{A} [\mathbf{I}_P \mid \mathbf{0}_P], \\
\mathbf{P} &= (\mathbf{P}^{T}, \pi^{m})^{T},
\end{align*}
\]

with \( \mathbf{I}_P \) the identity matrix of \( P \times P \) dimensions and \( \mathbf{0}_P \) a null vector of dimension \( P \).

The final information used to represent one plane is the set of features that belong to the plane with their SURF descriptors. The coordinates of each feature are expressed in the reference image of the plane.

### 2.1. Links between planes

Two planes \( m \) and \( n \) are defined as co-visible if there are at least two consecutive images in which both planes are detected. This idea of co-visibility has a great interest for navigation and localization tasks in future uses of the map. If a camera localizes one plane of the set it will know which other planes it might see when it moves, and so the space searched during the task execution will be reduced.

When two planar regions are visible together in two consecutive images it is possible to extract multi-plane constraints between them. An homology matrix, also called “relative homography” captures the relative motion between the images through two planes visible in the two images. Let us suppose that \( m \) and \( n \) are both visible in \( I_{k} \) and \( I_{k+1} \). The homology is obtained by multiplying one of the homographies by the inverse of the other one, \( \mathbf{H}^m_{k,k+1} = (\mathbf{H}^n_{k,k+1})^{-1} \mathbf{H}^n_{k,k+1} \). The homology is the chosen criterion to create a link between planes in the topological map. When two planes have been detected together in two consecutive images the algorithm sets the connecting edges to 1:

\[
\mathbf{A}(m,n) = \mathbf{A}(n,m) = 1 \Leftrightarrow \exists I_{k}, I_{k+1} \in \mathcal{I} \mid \mathcal{H}^m_{k,k+1}.
\]

Let us note that although the link is created considering a geometric criterion between the planes, the map does not include any metric information.

The homology has also some properties that can be useful for robust detection of new planes in the sequence. Using the Sherman-Morrison formula [34], as in [20], the homology matrix can be decomposed in \( \mathcal{H}^m_{k,k+1} = \mathbf{I} + \mathbf{v} \mathbf{p}^{T} \), where

\[
\mathbf{v} = (v_1, v_2, v_3)^{T} = \frac{\mathbf{R}^{-1}_{k,k+1} \mathbf{t}_{k,k+1}}{1 + \frac{(\mathbf{m}^{m}_{k+1})^{T} \mathbf{R}^{-1}_{k,k+1} \mathbf{m}^{m}_{k+1}}{d^{m}_{k+1}}},
\]

is a view dependent vector and

\[
\mathbf{p} = (p_1, p_2, p_3)^{T} = (\frac{\mathbf{n}^{m}_{k+1})^{T}}{d^{m}_{k+1}} - (\frac{\mathbf{n}^{n}_{k+1})^{T}}{d^{n}_{k+1}}) \mathbf{K}^{-1}
\]

is a plane dependent vector. The homology is used to separate real planes from false and repeated ones. This is
done using its eigenvalues, \( \{\lambda_1, \lambda_2, \lambda_3\} \), which for a correct homology must have the form \( (1, 1, 1 + \nu_1 p_1 + \nu_2 p_2 + \nu_3 p_3) \). Before adding a new plane to the map the eigenvalues of the homology must hold

\[
|\lambda_1 - 1| \leq \epsilon, \quad |\lambda_2 - 1| \leq \epsilon, \quad |\lambda_3 - 1| \geq \epsilon,
\]

for a sufficient small \( \epsilon \). If the three eigenvalues are close to the unity it means that the two planes are actually the same one (the homology is an identity matrix), so instead of creating a new plane, the new features are added to the existing one. On the other hand, if two of the three eigenvalues are not close enough to the unity there is an homography that is not describing a real plane. In this second case the new plane is ignored. Let us notice that the test is pure image-based and the method still does not need any information about neither the camera calibration nor the motion between the images.

2.2. Loop closing

The last problem considered to build the individual maps is to detect when a plane appears in the sequence because the camera is revisiting the same place (loop closing). To consider this situation every time a new plane is detected, the algorithm matches the features of the new plane with the rest of existing planes and tries to compute a robust homography between them. If for some plane the corresponding homography exists and it is supported by most of the matches it means that both planes are the same and must be merged. The merging process is performed by adding to the existing plane the new features and by updating \( \mathbf{A} \) through eq. (2).

In the end, all the method can be summarized in the Algorithm 1

Algorithm 1 Single camera topological map

1: Extract planes from \( \mathcal{I}_1 \) and \( \mathcal{I}_2 \)
2: Create \( \mathcal{G} \) with the initial planes
3: \( \mathbf{A}(m,n) = \mathbf{A}(n,m) = 1, \forall m \neq n \)
4: for all \( \mathcal{I}_k \in \mathcal{T} \) do
5: Match features in \( \mathcal{I}_k \) and \( \mathcal{I}_{k+1} \)
6: for all \( \text{visible } m \in \mathcal{P} \) do
7: Select the matches that belong to \( \pi^m \)
8: Compute \( \mathbf{H}^m_{k,k+1} \) and \( \mathbf{H}^m_{r,m,k+1} \) with DLT+Ransac
9: Add new features to \( \pi^m \) using \( \mathbf{H}^m_{r,m,k+1} \)
10: end for
11: Search for new planes in the remaining matches
12: if new plane was already in the map then
13: Update \( \mathcal{G} \)
14: else
15: Add the new plane to \( \mathcal{G} \) (eq. (1))
16: end if
17: Modify \( \mathbf{A} \) with the new homologies (eq. (2)-(5))
18: end for

3. Multi-Camera Distributed Topological map

Let us consider now a set, \( \mathcal{C} \), of \( C \) mobile agents with cameras and limited communication capabilities. Each agent manages an individual topological map \( \mathcal{G}_i = \{ \mathcal{P}_i, \pi_i, \mathbf{A}_i \} \), \( i = 1, \ldots, C \). The communications among the agents are defined by a graph \( \mathcal{G}_{\text{comm}} = (\mathcal{C}, \mathcal{E}) \), where \( (i,j) \in \mathcal{E} \) if agent \( i \) can communicate with agent \( j \). The set of neighbors of the agent \( i \) is the set of agents that can directly communicate with it, \( N_i = \{ j \in \mathcal{C} | (i,j) \in \mathcal{E} \} \). The diameter of the graph \( \text{Diam}(\mathcal{G}_{\text{comm}}) \) is the maximum length of a path between any two nodes of the graph.

Assumption 3.1. The communication graph, \( \mathcal{G}_{\text{comm}} \), is connected, i.e., there exists a path that connects any two nodes of the network.

Not every agent can directly communicate with each other and the goal is to make all the agents compute an identical global map, \( \mathcal{G}^* = (\mathcal{P}^*, \pi^*, \mathbf{A}^*) \), containing the information acquired by the whole set of agents. The computation of the global map can be divided in two parts. On one hand the information about the graph of planes and their relations (\( \mathcal{P}^* \) and \( \pi^* \)), and on the other hand the reference image of each planar region, how many features they contain and their coordinates (\( \mathbf{A}^* \)).

A distributed consensus approach is followed to compute the global map. Distributed consensus techniques [30] study how to reach an average of the total information considering only local interactions. A leader election approach is followed to obtain the consensus on the global graph and the SURF descriptors, whereas for the feature coordinates a distributed averaging rule is used. In order to use these techniques several properties must be ensured:

1. Information of the agents: it is required that all the agents have an initial value of the information.
2. Data association: in order to perform the fusion it is necessary to know which planes in the local maps are associated. The data association between features belonging to the same plane in different maps is also required.
3. Common reference frame: given two planes from two different maps which correspond to the same planar region in the world, the corresponding sets of features must be expressed in the same reference frame.

However, local maps composed by planes and image features do not satisfy the properties above mentioned. Solutions to overcome these problems are proposed, obtaining a common global map equal for all the agents.

3.1. Information of the agents

It is supposed that none of the agents has the information of all the planes seen by the whole team. Therefore, in the first step the local maps, \( \mathcal{G}_i \), are augmented so that the size and the order of each of them is the same, \( \mathcal{P}_i^* = \mathcal{P}_j^* \) and \( |\pi^*_i| = |\pi^*_j| \), \( \forall i,j \in \mathcal{C} \).
The ordering of the planes is done by univocal identification of both the agents and the planes. The agents can be identified, for example, using the ip addresses whereas the planes are ordered as they were detected in the local sequences. These two elements define a global order of the whole set of maps. The function \( O : \mathcal{P} \to \mathbb{N} \) is defined in such a way that it returns the order of a given plane in the map. For example, the first plane observed by the first agent will have a smaller position in the map than the fourth plane observed by the same agent but a larger value in the global order than any plane observed by the second or the first agent.

The different size of the initial maps is solved creating fictitious planes in the local maps so that all the agents have a final map of the same size. A fictitious plane, \( \tilde{m} \), is a plane with no relations in the graph and for which all the coordinates of all its features, \( \pi_i \), are initialized to zero. Every agent creates as many fictitious planes as the total number of planes observed by the other agents. This is done exchanging local messages so that, every agent \( i \) eventually has a feature vector

\[
\Pi_i^* = (0_1^T \ldots \Pi_i^T \ldots 0_n^T)^T
\]

and an adjacency matrix \( A_i^* \) with the form

\[
A_i^* = \begin{bmatrix}
0_{11} & \cdots & 0_{1i} & \cdots & 0_{1C} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
0_{i1} & \cdots & A_i & \cdots & 0_{iC} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
0_{C1} & \cdots & 0_{Ci} & \cdots & 0_{CC}
\end{bmatrix}
\]

where \( 0_i \) is a vector of zeros with dimension \( |\Pi_i| \) and \( 0_{ij} \) is a matrix of zeros with dimension \( P_i \times P_j \), \( j = 1, \ldots, C, j \neq i \). Let us note that in order to create a fictitious plane it is only necessary to know which agent has seen it, the order in the local map and the number of features it contains. This reduces considerably the size of the exchanged messages.

When new messages containing information from other agents are received, the fictitious planes are added to the local maps,

\[
P_i^* = P_i^* \cup \tilde{m}.
\]

Regarding the adjacency matrices, \( A_i^* \), for any new fictitious plane a new row and column with zeros is created

\[
A_i^* = P_{O(\tilde{m})} \left[ I_{P_i} \mid 0 \right]^T A_i^* \left[ I_{P_i} \mid 0 \right] P_{O(\tilde{m})}^T,
\]

where the middle matrices augment the adjacency matrix as in eq. (1) and \( P_{O(\tilde{m})} \) is a permutation matrix that moves the last row, \( P_i \), to the row \( O(\tilde{m}) \) and displaces all the rows in between one position down. Finally, \( \Pi_i^* \) is also updated by adding fictitious features, with coordinates equal to zero, in the corresponding position. After \( \text{Diam}(G_{\text{comm}}) \) rounds of exchanging information, all the agents will know the size of the global map, and all the vectors \( \Pi_i^* \) and matrices \( A_i^* \) will have the same dimension and the form of eqs. (6) and (7) respectively. This process is summarized in Algorithm 2.

### 3.2. Data association

Here two issues must be addressed. The first one is the data association between planes in different maps and the data association of features between matched planes. Once the data association is known, the second issue is to reduce the adjacency matrices (respectively feature vectors) so that they have the correct number of elements and in the right order.

The data association is performed based only on local information. Each agent exchanges its initial local map with its set of neighbors. For all the maps, the data association is carried out computing homographies between the local planes and the planes of the other maps. Two planar regions are associated if a robust homography exists between them. The homography is computed matching the invariant features and using DLT+RANSAC algorithm.

**Definition 3.1.** Given two maps \( G_i^* \) and \( G_j^* \) the function \( M : G_i^* \times G_j^* \to \{0, 1\}^{P_i \times P_j} \) returns an association matrix, \( W_{ij} \), where

\[
W_{ij}(m, n) =
\begin{cases}
1 & \text{if } \pi_i^m \text{ and } \pi_j^n \text{ are related by a homography } H_{ij}^{mm}, \\
0 & \text{otherwise}.
\end{cases}
\]

(10)

**Assumption 3.2.** The matching function has the following properties

- **Self association**, \( M(G_i^*, G_i^*) = I_{P_i} \)
- **Planes are associated in a one to one way**

\[
\sum_{m=1}^{P_i} W_{ij}(m, n) \leq 1 \text{ and } \sum_{n=1}^{P_j} W_{ij}(m, n) \leq 1,
\]

- **Given two maps \( G_i^* \) and \( G_j^* \)** it holds that \( M(G_i^*, G_j^*) = W_{ij}^T = (M(G_j^*, G_i^*))^T \).

Let us note that these assumptions are satisfied by the robust homography matching.

All the local associations can be put together in a general association matrix, \( W \in \mathbb{N}^{P_i \times P_j} \),

\[
W = \begin{bmatrix}
W_{11} & \cdots & W_{1C} \\
\vdots & \ddots & \vdots \\
W_{C1} & \cdots & W_{CC}
\end{bmatrix},
\]

(11)
where
\[
W_{ij} = \begin{cases} 
M(G_i, G_j) & \text{if } j \in \{N_i \cup i\}, \\
0 & \text{otherwise},
\end{cases}
\] (12)
in which each agent has the information of one of the block-rows.

The rows give information to the agents about associations of planes with direct neighbors’ maps. The question now is how two agents that are not direct neighbors but have seen the same plane are aware of this situation. The global association matrix can also be seen as an adjacency matrix of the whole set of planes, \(G^*\). Following this reasoning, two planes are associated if there exists a path in \(W\) that links them. As the following lemma states, given a graph \(G^*\), the powers of its adjacency matrix contain the information about the number of paths existing between the nodes of \(G^*\):

**Lemma 3.1 (Lemma 1.32 [30])**. Let \(G^*\) be a graph of order \(P^*\) with un-weighted adjacency matrix \(W \in \{0,1\}^{P^* \times P^*}\), and possibly with self loops. For all \(m,n \in \{1, \ldots, P^*\}\) and \(t \in \mathbb{N}\) the \((m,n)\) 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 \(m\) to node \(n\).

In the problem presented here the nodes refer the different planes.

We have proposed a distributed algorithm to compute the powers of such adjacency matrix [32]. The algorithm is proved to finish in finite time and finds all the paths between elements in \(G^*\). After the execution of the algorithm each agent knows all the associations of its planes with the planes of the rest of the agents, even if they are not direct neighbors.

Given a plane, \(m\), let \(B(m) = \{n\mid W^t(m,n) = 1\}\) be the set of planes associated to \(m\) and \(\bar{n}(m) = \arg \min_{n \in B(m)} O(n)\) be the plane with the lowest value in the global order. For all the associations of the agent \(i\), the adjacency matrix \(A_i^*\) is updated to put together all the associated planes
\[
A_i^* = I_i(A_i^* \lor P_{\bar{n}(m),n} A_i^* \lor A_i^* P_{\bar{n}(m),n} I_n^T),
\] (13)
\(\forall m \in \mathcal{P}_i, n \in B(m)\), where \(P_{\bar{n}(m),n}\) is a permutation matrix of the rows \(\bar{n}(m)\) and \(n\), and \(I_n\) is an identity matrix where the \(n^{th}\) row has been deleted. The symbol \(\lor\) represents the or operation between the matrices, which can be done taking into account that all the elements of the matrices are in the set \(\{0,1\}\). Let us note that row \(\bar{n}(m)\) will be the same for all the agents with planes in the set \(B(m)\). This means that all the agents will move the information of each association to the same row and will delete the rest of the rows, maintaining the size and the order of their maps.

The last problem is to combine the associations of planes in which there is no plane belonging to \(\mathcal{P}_i\). Since the agent \(i\) has access only to the \(i^{th}\) block of rows within \(W^t\), all the associations in which there is no plane belonging to \(\mathcal{P}_i\) will be unknown. To solve this problem a similar exchange of messages like the one to augment the local maps is carried out. In this case the exchanged information are the sets of associated planes \(B(m)\). After \(\text{Diam}(G_{\text{comm}})\) iterations all the associations are received by all the agents and using eq. (13) all the updates are done. Algorithm 3 schematizes the data association step for planar regions.

**Algorithm 3 Data association for planes - Agent \(i\)**

1: Exchange local maps with neighbors
2: Compute \(W_{ij}, j \in \mathcal{N}_i\)
3: repeat
4: Update the \(W_{ij}\) as in [32]
5: until \(W_{ij}\) does not change \(\forall j \in \mathcal{C}\)
6: for \(it = 1 \ldots \text{Diam}(G_{\text{comm}})\) do
7: Send block associations to \(j \in \mathcal{N}_i\)
8: Receive block associations from \(j \in \mathcal{N}_i\)
9: Update \(A_i^*\) (eq. (13))
10: end for

Regarding the association of the features a similar process is performed for all the features belonging to the same planar region. After such process all the vectors \(\Pi^*\) are updated, having the same size and order.

### 3.3. Common reference

At this point the local maps of planes have been associated and all of them have the same size and are equally sorted. Although the adjacency matrices are ready to execute the consensus algorithm, the features still require a common reference image. Otherwise posterior consensus will give erroneous results. In order to solve this problem a leader election algorithm is used to decide, for each plane, which one of the local reference images is defined as the common reference image. A distributed approach of the leader election algorithm can be found in [35]. Initially every agent starts considering itself as leader. At each step the leader value, \(\text{lead}_i(t)\), is updated considering the values of its neighbors.

\[
\text{lead}_i(t + 1) = \max(\text{lead}_i(t), \text{lead}_j(t)) \quad j \in \mathcal{N}_i.
\] (14)

After \(\text{Diam}(G_{\text{comm}})\) iterations the rule in (14) converges to the same leader for all the nodes in the network.

In the case of planes, not only a leader (reference image) needs to be chosen for every plane but also the homography that transforms the features from the local references to the global one. The number of observed features in the plane, \(|\pi|\), is the criterion chosen to decide the reference. For every plane, algorithm 4 is executed by every agent. In the algorithm, \(H_{ij}\) represents the homography that transforms the plane from the coordinates in the local reference to the coordinates of the neighbor’s reference. This homography is the one computed in the local data association step. The variables \(\text{feats}_i\) represent the number of real features that each plane, \(j\), has. For the global
reference no subscript is used. After \( \text{Diam}(G_{\text{comm}}) \) iterations all the agents know the common reference (the one which contains more real features) and the homography to transform their coordinates. Applying this transformation and normalizing the coordinates, all the agents have their features in the same reference frame. The fictitious planes are not affected by changes of the reference. Taking this into account the agents with a fictitious plane do not participate in the leader election algorithm of such plane.

**Algorithm 4** Choice of a common reference - Agent \( i \)

1. \( \text{lead} = i; \text{feats} = |\pi|; H_{r,i} = I \)
2. for \( it = 1 \ldots \text{Diam}(G_{\text{comm}}) \) do
3. Send \([\text{lead,feats},H_{r,i}]\) to all \( j \in N_i \)
4. Receive \([\text{lead}_j,\text{feats}_j,H_{r,j}]\) from all \( j \in N_i \)
5. if \( \text{feats}_j > \text{feats} \) then
6. \( \text{lead} = \text{lead}_j; \text{feats} = \text{feats}_j; H_{r,i} = H_{r,j}H_{j,i} \)
7. end if
8. end for
9. Transform the features’ coordinates of \( \pi \) using \( H_{r,i} \)

3.4. Consensus on the global map

From here on all the nodes have the information needed to perform the consensus. The graphs \( G_i^* \) satisfy now all the requirements to apply distributed consensus algorithms.

The adjacency matrices of the local maps are updated with the following rule

\[
A_i^*(m,n) = A_i^*(m,n) \lor A_j^*(m,n), \forall j \in N_i
\]  

**Theorem 3.1 (Convergence of \( A^* \)).** The set of adjacency matrices \( A_i^* \) under iteration rule (15), converges in \( \text{Diam}(G_{\text{comm}}) \) iterations to a common matrix \( A^* \) that includes all the links between planes.

**Proof.** Let us consider separately each element of the matrices \( A_i^* \). Considering that the initial value of the elements is \( \{0,1\} \), then it holds that

\[
A_i^*(m,n) \lor A_j^*(m,n) = \max(A_i^*(m,n), A_j^*(m,n)).
\]  

Then eq. (15) can also be seen as an update of a leader election algorithm, which is proved to converge in \( \text{Diam}(G_{\text{comm}}) \) iterations [35].

Since \( A_i^*(m,n) = 1 \) implies that there is a link between planes \( m \) and \( n \) and \( \max(0,1) = 1 \) then all the links of the adjacency matrices are preserved.

With respect to the features, since no observations are assumed to be better than others, an average of the matched observations is computed. Let us consider one feature, \( f \), belonging to a plane of the global map. Let \( f_i \) denote the value that agent \( i \) has of \( f \) after the common reference transformation. The common consensus update rule followed by every agent is

\[
f_i(t+1) = f_i(t) + \sum_{j \in N_i} w_{ij} (f_j(t) - f_i(t)),
\]

where \( w_{ij} \) are the weights used to fuse the information. If assumption 3.1 holds and the weights \( w_{ij} \) have the property of configuring a double stochastic matrix then the iteration rule in (17) converges for all \( i \) to the average of the initial values,

\[
\forall i \in C \ f_i(t) \to \bar{f} = \frac{1}{C} \sum_{i \in C} f_i(0) \text{ as } t \to \infty.
\]  

In order to obtain weights composing a doubly stochastic matrix the Metropolis formula is used. Metropolis weights’ definition is based on the number of neighbors of each agent:

\[
w_{ij} = \begin{cases} 
\frac{\min\left(1, \frac{1}{\nu_{ij}}\right)}{\nu_{ij}} & \text{if } (i,j) \in \mathcal{E} \\
1 - \sum_{k=1 \ldots C} w_{ik} & \text{if } i = j \\
0 & \text{otherwise}
\end{cases}
\]  

Let us note that although the fictitious planes do not provide any information to the final consensus, they affect it in the sense that the final value is divided by the total number of agents. The third coordinate of the features plays here a fundamental role. Let us recall that for the fictitious features the third coordinate, usually related with the scale of the point, was also set to zero. Let \( C_f \subseteq C \) the set of agents that have observed \( f \) in their local maps.

**Theorem 3.2 (Average of features).** If \( C_f \neq \emptyset \), then the iteration rule (17) converges, in normalized coordinates, for all the agents in the network to

\[
f_i(t) \to \frac{1}{|C_f|} \sum_{j \in C_f} f_j(0) \text{ as } t \to \infty \quad \forall i \in C.
\]  

**Proof.** By the properties of the consensus rule, the features converge to

\[
f_i(t) \to \frac{1}{C} \sum_{j \in C} f_j(0).
\]  

The third coordinate of the features is either 1 if the agents belongs to \( C_f \) or 0 if it does not. Therefore, this coordinate converges to

\[
|f_i(t)|_3 \to \frac{1}{C} \sum_{j \in C} 1 = \frac{|C_f|}{C}.
\]  

Once the coordinates are normalized, the convergence to the desired value is proven.

Let us note that the consensus is only carried out for the coordinates of the features and not for the whole descriptor. For the SURF descriptors a leader election algorithm is used so that every agent has the same set of SURF descriptors after \( \text{Diam}(G_{\text{comm}}) \) steps.
4. Experimental Results

Several experiments have been carried out in order to evaluate the properties and the behavior of our proposal. We have tested it using different real image data sets that correspond to different locations of man-made environments with plenty of planar regions. The first data set has been recorded indoors (House data set). It consists of 3600 frames from different rooms. The second data set is composed by nine different sequences recorded outdoors in a downtown Zaragoza area (Downtown data set). Figure 2 shows a view of the map where these sequences have been acquired and the topology used in the experiments.

The camera used in all the cases has been a Panasonic Lumix FX-500. In all the cases the camera has moved with 6DOF. For all the images we have used SURF descriptors [31] for matching. The computation of the homographies has been done using DLT+RANSAC algorithm. It is well known that under pure rotations or small motions all the features can be fitted to the same homography. In video sequences with high frame rates this is a common situation. To avoid this problem we have followed the idea of [36] to select key frames among the sequence:

- There are as many images as possible between the key frames $I_k$ and $I_{k+1}$.
- There are at least M matches between the key frames $I_k$ and $I_{k+1}$.
- There are at least N matches between the key frames $I_k$ and $I_{k+2}$.

The results are divided in three sections. In the first experiment (sec. 4.1) we analyze how the triple matching step works and the properties of the extracted planes. In the second experiment (sec. 4.2) we analyze the properties of the individual maps created using the sequences of images from each camera separately. We have compared the resulting graphs obtained for both data sets with graphs made by images [8]. For the latter approach we have stored the SURF descriptors of each image and we have imposed the homography constraint between frames to observe the pros and cons of using images or planes. In the last experiment (sec. 4.3) we have tested the multi-camera distributed approach for the Downtown data set.

4.1. Extraction of the planar regions

The House data set has helped us to test the detection of planes since it has a lot of different planar regions. In Fig. 3 some of the segmented regions are depicted. We observe that although the method detects several planes which are the same in the ceiling, there are no wrong planes segmented.

The nine sequences in the Downtown data set are more challenging because in this case the scene contains dynamic elements such as people and cars. There are also a lot of trees that occlude the building facades and, in some cases, there are also illumination changes when the camera points towards the sun. Figure 4 and 5 depict two examples of two different planes extracted using our method. Even when the extracted planes contain some outlier features the results are still quite good and, what is more important, we observe that the method maintains the planar regions correctly adding new features as they appear.
Figure 4: Planar region extracted from one sequence of the Downtown data set. The algorithm is able to track and grow the plane over 63 different frames with dynamic elements and some illumination changes. The top figure represents the plane with the 482 detected features. The plane has been observed in all the images depicted below.
We have observed that sometimes in a real scenario many dynamic objects may generate planar regions. These planar regions are undesirable in a practical situation. A lower bound on the number of features of a plane after the map generation clears almost all the undesired planes. Other times the algorithm considers as a planar region a set of features belonging to different planes but coplanar between them. Usually in the next steps the algorithm grows this plane considering only the biggest number of real coplanar features. The last problem observed comes from the homology test. The homology depends both on the motion between the images and the parameters of the planes. If the motion is small, it is possible that 2 different planes are fused because the three eigenvalues of the homology will be close to one. The algorithm used to select the key frames [36] mitigates this problem. Since the algorithm tries to skip as many images as possible between two consecutive key-frames, consecutive frames will be, in general, far away from each other in terms of distance. Therefore, the homology will only be close to the identity when the planes must actually be fused.

4.2. Single camera topological map of planar regions

Using the planar regions extracted with our algorithm from the sequences of images, we have computed the associated graphs of planes. We have compared the resulting graphs with the image graphs created following the approach in [8]. Table 1 shows the comparison of the graphs generated using the House data set and Table 2 shows the same comparison for the nine sequences of the Downtown data set.

Table 1: Results for the house sequence

<table>
<thead>
<tr>
<th>Map</th>
<th>Nodes</th>
<th>Edges</th>
<th>Feats</th>
<th>Feats/node</th>
<th>Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Images</td>
<td>140</td>
<td>1728</td>
<td>78124</td>
<td>558</td>
<td>29.0</td>
</tr>
<tr>
<td>Planes</td>
<td>30</td>
<td>78</td>
<td>10832</td>
<td>361</td>
<td>4.0</td>
</tr>
</tbody>
</table>

In both cases the graph made of planes has less nodes and edges than the graph composed by images. The amount of space for storing the information is drastically reduced using our approach (Tables 1 and 2). Notice that in our approach the size of each plane is not bounded and there can be big differences between nodes. In a visual memory made by images all the nodes will have similar size (the features per node can be assumed to be bounded) whereas the graph made of planes may contain very small planar regions with just a few features, and other nodes can represent large planar regions with hundreds of features and many homographies.

If the camera moves too fast or if there is a sequence of images in which there are no planar regions the topological map will be unconnected. To prevent these situations, we have also imposed consecutive planes to be connected in the local maps.

4.3. Multi-Camera Distributed Topological map

The distributed building of a topological map has been done using the maps generated from the Downtown data set. The limited communications between the agents are shown in Fig 2. The diameter of the graph is 4, which means that most part of the algorithms will finish only in four steps.

As previously commented, the triple matching algorithm finds a lot of small planes which are a mix of different outliers (trees, buses, people and noisy features). These planes do not apport real information and it is better to discard them. In order to do so we have set a threshold of 50 features, so that planes containing less features than the threshold are not considered for the distributed global map. As shown in Table 3, after erasing the small planes, only 132 of the 287 planes take part in the distributed process. These planes amount a total of 46981 features.
Table 2: Results for the downtown Zaragoza sequences

<table>
<thead>
<tr>
<th>Agent</th>
<th>Map</th>
<th>Nodes</th>
<th>Edges</th>
<th>Feats</th>
<th>Feats/node</th>
<th>Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Images</td>
<td>300</td>
<td>3966</td>
<td>315830</td>
<td>1052</td>
<td>117.4</td>
</tr>
<tr>
<td></td>
<td>Planes</td>
<td>104</td>
<td>340</td>
<td>15694</td>
<td>150</td>
<td>5.89</td>
</tr>
<tr>
<td>2</td>
<td>Images</td>
<td>78</td>
<td>796</td>
<td>58219</td>
<td>746</td>
<td>21.6</td>
</tr>
<tr>
<td></td>
<td>Planes</td>
<td>17</td>
<td>62</td>
<td>3958</td>
<td>232</td>
<td>2.28</td>
</tr>
<tr>
<td>3</td>
<td>Images</td>
<td>33</td>
<td>230</td>
<td>29777</td>
<td>902</td>
<td>11.1</td>
</tr>
<tr>
<td></td>
<td>Planes</td>
<td>8</td>
<td>28</td>
<td>1084</td>
<td>135</td>
<td>0.4</td>
</tr>
<tr>
<td>4</td>
<td>Images</td>
<td>465</td>
<td>7192</td>
<td>431607</td>
<td>928</td>
<td>160.6</td>
</tr>
<tr>
<td></td>
<td>Planes</td>
<td>92</td>
<td>288</td>
<td>18497</td>
<td>201</td>
<td>6.95</td>
</tr>
<tr>
<td>5</td>
<td>Images</td>
<td>86</td>
<td>820</td>
<td>74123</td>
<td>861</td>
<td>27.5</td>
</tr>
<tr>
<td></td>
<td>Planes</td>
<td>34</td>
<td>110</td>
<td>4104</td>
<td>120</td>
<td>1.54</td>
</tr>
<tr>
<td>6</td>
<td>Images</td>
<td>101</td>
<td>1434</td>
<td>79700</td>
<td>781</td>
<td>29.6</td>
</tr>
<tr>
<td></td>
<td>Planes</td>
<td>11</td>
<td>40</td>
<td>409</td>
<td>361</td>
<td>1.68</td>
</tr>
<tr>
<td>7</td>
<td>Images</td>
<td>32</td>
<td>304</td>
<td>29258</td>
<td>914</td>
<td>10.9</td>
</tr>
<tr>
<td></td>
<td>Planes</td>
<td>6</td>
<td>14</td>
<td>1611</td>
<td>268</td>
<td>0.61</td>
</tr>
<tr>
<td>8</td>
<td>Images</td>
<td>35</td>
<td>230</td>
<td>30434</td>
<td>809</td>
<td>11.3</td>
</tr>
<tr>
<td></td>
<td>Planes</td>
<td>9</td>
<td>18</td>
<td>2021</td>
<td>224</td>
<td>0.77</td>
</tr>
<tr>
<td>9</td>
<td>Images</td>
<td>32</td>
<td>318</td>
<td>29237</td>
<td>913</td>
<td>10.9</td>
</tr>
<tr>
<td></td>
<td>Planes</td>
<td>6</td>
<td>12</td>
<td>1162</td>
<td>193</td>
<td>0.44</td>
</tr>
<tr>
<td>Total</td>
<td>Images</td>
<td>1162</td>
<td>15290</td>
<td>1078185</td>
<td>927</td>
<td>400.9</td>
</tr>
<tr>
<td></td>
<td>Planes</td>
<td>287</td>
<td>912</td>
<td>48540</td>
<td>169</td>
<td>20.20</td>
</tr>
</tbody>
</table>

Initially the agents exchange the information about their maps to create fictitious planes. After four steps, every agent has a map with 132 planes and 46981 features. Then the agents exchange their maps with their neighbors and perform the local data association step. Figure 6 shows one plane seen by three different neighbor agents with the found matches. The local associations delete a total of 11 planes and 340 features, remaining 121 planes and 46351 features (third row in Table 3). We have observed that the small number of associations is mainly due to the different points of view of the trajectories and not because of mismatching. Even so, in the multi-camera mapping it is advisable to use larger RANSAC thresholds.

Table 3: Evolution of the global map’s size

<table>
<thead>
<tr>
<th>Step</th>
<th>Nodes</th>
<th>Feats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial global map</td>
<td>287</td>
<td>48540</td>
</tr>
<tr>
<td>After erasing small planes</td>
<td>132</td>
<td>46981</td>
</tr>
<tr>
<td>Data Association</td>
<td>121</td>
<td>46351</td>
</tr>
</tbody>
</table>

After data association the agents execute the leader election algorithm in order to fix the common references for each plane (Algorithm 4). In the example of the Figure 6 the reference plane is the top one because is the one with the most features.

Finally, the agents execute the consensus rule to reach an average on the features and a consensus on the adjacency matrices of the topological maps. Figure 7 shows the consensus evolution of the three coordinates by the nine agents; we can see that they do not reach the desired average in $f_x$ and $f_y$ because only three of the nine agents have information about the feature but the average considers the value of the nine. However, the normalized coordinates (Fig. 8) converge to the desired value, as stated in Theorem 3.2.

A comparison with the maps made by images has been done in order to analyze the amount of information transmitted through the network. This analysis has been carried out analytically considering the network topology and the information exchanged. An upper bound on the number of messages required to transmit some information to the considered network (flooding) is 14. That means that for the map made of images, the 400.9 megabytes (Table 2) are transmitted 14 times, giving a total of 5612 transmitted megabytes. The breakdown of the transmitted information using topological maps is in Table 4. The results show that the total information transmitted through the network using processed planes is considerably smaller (20%). Also the final global map is better using planar regions, since every match mixes the information reducing the total amount of data, whereas with images the total size is the sum of the local maps.

Finally, we analyze how the connectivity of the graph affects the results. Networks with more links will present a faster convergence of the consensus algorithm, i.e., less
iterations in Figs. 7 and 8, whereas sparse networks will require a larger number of iterations. Other aspect that will be affected is the number of associations, the more links the communication graph presents, the easier will be to match planes and features. In the experiment, even when the network is dense, there are few associations between planes which implies that, in this case, the algorithm would obtain the same results even if many edges were removed.

5. Conclusions

We have presented an algorithm for topological map building using automatically extracted planes. The planar regions are extracted using the information from previous images performing a Plane-Image-Image matching that tracks and grows the planar regions as new areas of the plane become visible. The planes are organized in a graph, built simultaneously to the extraction. The idea of storing planes as information in the top graph rather than whole images (or each image feature set) presents several advantages. It includes some intrinsic information about the scene structure and it reduces the graph size and complexity. This reduction is seized on multi-camera applications. We have proposed how to use distributed consensus techniques to merge the local maps obtained by different agents and to compute a distributed consensus on the global map. The problems using these techniques in our approach: missing information, data association and different reference images, have been exposed and solved also using distributed techniques, proving final convergence to the real average of the data. Experimental results with real images indoors and outdoors show the good performance of the method.

6. Acknowledgments

This work was supported by project DPI2009-08126 and grant AP2007-03282 Ministerio de Educacion y Ciencia. The authors would like to thank A.C. Murillo for her comments on the paper.

7. References

Figure 7: Consensus process for feature coordinates. Initially each agent has a different value of the feature coordinates. The nine agents exchange the information they have with their neighbors. It is observed that after 25 iterations consensus has been achieved and all the agents have the same value of the coordinates.

Figure 8: Consensus process for normalized feature coordinates. Evolution in normalized coordinates of the value of the same feature than in figure 7. Only three agents have a real observation of the feature, which are (358.1072, 292.3148)T, (364.2285, 297.2203)T and (358.9520, 297.1431)T. The rest of the agents have fictitious features with initial value equal to zero. At the beginning the agents with a fictitious coordinate have ∞ values. We can see that eventually the nine agents reach the same value, which corresponds to the average of the measurements of the three agents that observed the feature.


