The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Topological maps based on graphs of planar regions

E. Montijano and C. Sagues
DIIS - I3A, University of Zaragoza, Spain
Email: {emonti, csagues} @unizar.es

Abstract— **Topological visual maps contain different abstraction levels of information that can be used by robots to carry out different activities. We propose here a new hierarchical structure in which landmarks extracted from conventional images are grouped creating a graph of planar regions. The new hierarchy improves previous approaches based on images reducing both, the size of the graph and its complexity. In order to segment and group the planar regions of a sequence of images a new approach based on the simultaneous matching of two images and the previously extracted planar regions is proposed. We also consider multi-plane restrictions so that the method is robust to the appearance of new planes. The paper presents two contributions. First the triple matching approach to extract all the planes seen in the set of images and second a new topological map construction based on a graph of planar regions which can be used by mobile robots to localize and move in the environment. Experiments with real images in both indoor and outdoor environments show good performance of our proposal.**

Index Terms - homography, topological maps, planar regions.

I. Introduction

In robotics, vision sensors are usually chosen due to its low cost and the big amount of information they provide compared with other sensors. In this context, images stored in a visual memory have been extensively used. However, the big amount of data managed with this kind of sensors makes necessary to incorporate additional levels of organization. Topological maps are an interesting way to organize the visual information and are also very useful in the field of robotics. When the size of a map is big enough, an organization in different abstraction levels will make a robot to lose less time in the search of the corresponding images while it is performing its tasks. The information also must be useful for the robot in order to self localize and move in the environment with accuracy. Moreover, if a robot needs to cooperate with a human or simply receive orders from him it should be able to understand some basic human concepts such us walls, rooms or buildings. A multiple layer hierarchical representation can help to accomplish these two different tasks. As we will see, the combination of visual landmarks with a higher conceptual organization of this landmarks in planar regions can benefit robotic systems in many different ways.

Organization of the information in hierarchical structures is not a trivial question and several aspects must be considered. Murillo et al. present in [10] a hierarchical technique to localize a mobile robot using omnidirectional images. In a first step the robot detects the room where it is located using a pyramidal matching and in a second stage it recovers its metric localization using two reference images by computing the 1D trifocal tensor. In this approach the images of different rooms are manually sorted. Automatic sorting of the reference images is a desirable characteristic for the organization of the visual information. Zivkovic et al. [17] describe a method in which images are automatically sorted in a graph where each node represents an image and each edge represents a geometric relation between the two images defined by the epipolar geometry. The graph is later clustered into non intersected regions representing convex spaces. This kind of representation is very intuitive and has a good performance, but there is still room for improvements. We find two drawbacks in the use of maps made only by images. First, in hierarchical maps created with images there will be a lot of redundant information. Many features will be seen in several images so that the map will take up a lot of unnecessary space. The second disadvantage is the density of the graph. In one room most part of the images will have connections with all the other images in the same room. This implies that working with an image graph will be computationally expensive. We propose here an algorithm to extract and arrange all the planes contained in the scene in a similar graph, and, as we will show, with this approach the size of the graph is reduced in space and complexity.

There are several works in the literature that assume the presence of planes in the environment to solve different robotic problems. In these cases a representation based on planar regions can also help. For example [2], [4] and [8] use planes and homographies to control robots with high precision. In [11] and [3] graphs of images are used to move a robot through different routes using also homographies. In all these approaches the structure of the scene is not required and only sets of coplanar features are used. Taking this into account we have chosen not to compute or store any additional information about the scene but just extract and group sets of coplanar features in the image domain, which also makes the approach faster. Moreover, other tasks like structure reconstruction [13] or camera calibration [9] can also benefit from the knowledge of the existence of planes in the scene and can make use of the information provided.

This work was supported by the projects DPI2006-07928,DPI2009-08126, IST-1-045062-URUS-STP.

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

However, in order to build a topological map with one layer representing planar regions it is still necessary to detect all the planes that are visible in the images. The idea of segmenting all the planes that appear in a set of images is not new and there are several works in the literature that propose different algorithms and approaches to make it possible. A method for reconstructing the planar regions visible from two oriented images from sparse correspondences is presented in [6]. Zelnik-Manor and Irani [16] have shown that the homographies of multiple planes across multiple views rely in a 4-dimensional linear subspace and the new constraints can improve the estimations of the homographies. In [14] a method for detecting planes in images with a voting procedure is proposed. The method requires an initial estimation of the camera calibration and the motion between the two images, which is not always possible. All the previous approaches can benefit from the use of some knowledge of the planes extracted in previous images. Our approach uses this information to improve the quality of the computed homographies. Using a triple set, plane-image-image, for feature matching and homography computation we track the planes over the sequence and also grow them up as new features appear. Homology constraints allow us to detect new planes and also give us a geometric criterion to relate the planes in a hierarchical structure. The experimental results with real images show the good performance of this new algorithm.

Two contributions are presented in this paper. The first one is the new approach for detecting all the planes in a set of images and a set of coplanar features for robust matching and homography estimation. The second contribution is the new organization of visual information in a graph of planar regions where there exists a geometric relation between the different planes of the scene.

II. Segmentation of planes

We start with a set of ordered images $\mathcal{I} = \{\mathcal{I}_1, .., \mathcal{I}_n\}$, which correspond to a sequence captured with a camera onboard a robot or a 6DOF camera in hand. We intend to extract and organize the set of all the planes $\boldsymbol{\pi} = \{\pi^1, .., \pi^m\}$ in the scene from the set of images $\mathcal{I}$. For an easy understanding of the paper, subscript indices will correspond to images in the set whereas superscript indices will correspond to planar regions, for example π_i^k will represent the features of the k^{th} planar region seen in the i^{th} image.

The only assumption made (apart from the presence of the planar regions) is the rigidity of the scene. There is no knowledge about neither the internal parameters of the camera, represented by the calibration matrix $\mathbf{K}$, nor about the motion between consecutive frames, $\mathbf{R}$ and $\mathbf{t}$. If one plane, π^k is visible in two images of the scene, $\mathcal{I}_i$ and $\mathcal{I}_j$, it is possible to compute a projective mapping (inter-image homography), $\mathbf{H}_{ij}^k$, that relates the points belonging to the plane, $\pi_i^k = \mathbf{H}_{ij}^k \pi_j^k$. This homography is defined up to a scale factor and has the form $\mathbf{H}_{ij}^k = \mathbf{K}(\mathbf{R}_{ij} - (1/d_j^k)\mathbf{t}_{ij}(\mathbf{n}_j^k)^T)\mathbf{K}^{-1}$, with d_j^k and $\mathbf{n}_j^k$ the distance and normal of the k^{th} plane in the j^{th} frame respectively. The homography can be estimated from four correspondences without prior knowledge about the scene or the calibration [7] and we will exploit this to extract the different sets of coplanar features. In order to make the process more robust, RANSAC [5] has been used.

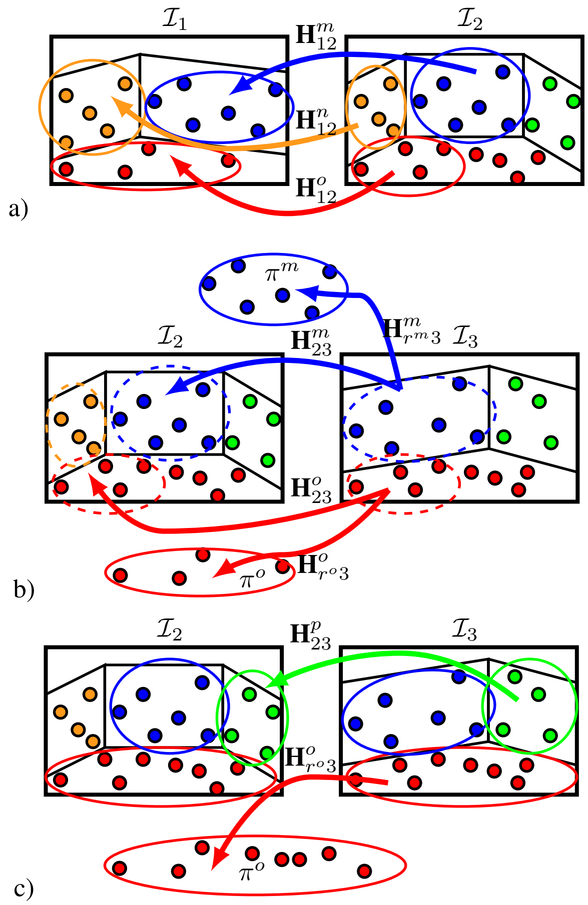

Fig. 1. Scheme of the plane segmentation. a) Extraction of the initial planes. b) Triple match Plane-Image-Image for speedy and robust homography computation. c) Addition of new points to the existing planes and detection of new planar regions within the remaining matches. The plane n is closed because there are not matches belonging to it between $\mathcal{I}_2$ and $\mathcal{I}_3$.

In images where there are several planes the DLT+RANSAC approach may fail due to bad election of the random samples used to compute the homographies. In addition, a high number of samples is required to have a chance that the four points of a sample belong to the same planar region. In order to avoid this problem only sets of four non collinear points that are close to each other are chosen as combinations for the algorithm. Closer points will have a bigger chance to be coplanar than points that rely in the image far from each other.

A. Triple matching Plane-Image-Image

The two initial frames of the sequence, $\mathcal{I}_1$ and $\mathcal{I}_2$ are picked up and all the planes seen in both images (Fig. 1-a) are extracted. Without loss of generality we explain the process for one plane π^m, being this process the same for every other plane (and independent of the number of planes). The homography between the images, $\mathbf{H}_{12}^m$, and all the features that belong to π^m, expressed in the reference image of the plane, $\mathcal{I}_{r^m}$ are stored. The reference image of a plane is defined as the first image of the sequence where the plane has been seen. The identifiers of the features for every image where the plane is observed are also stored so that the search for these features in the future will be automatic. This last information is stored only during the matching step, once all the process has been finished it is erased, since it is no longer necessary.

With the initial plane extracted the next image in the list is picked up, $\mathcal{I}_3$, and the common matches with $\mathcal{I}_2$ are found. From all the set of matches only those that belong to π^m are chosen, searching a new homography among this subset with RANSAC. The number of combinations required to have a probability, p, for one combination to be composed all by inliers is

$$k = \frac{log(1-p)}{log(1-w^4)}, \tag{1}$$

where w is the probability that one feature is an inlier in the plane. By taking the combinations only among a subset that represents a planar region we are increasing w, consequently reducing k, and therefore computing the homography in less time. The homography with respect to the points in the reference image of the plane, $\mathbf{H}^m_{r^m 3}$, is also computed so that the voting procedure is more robust, enforcing every feature to support both homographies instead just one (Fig. 1-b). The new homographies are then used to localize new features that belong to the plane. The new features are added to the plane using $\mathbf{H}^m_{r^m 3}$. Once all the matches belonging to the existing planes have been sorted we try to find new planes between the remaining matches, adding new planes to the set (Fig 1-c). The process is recursively repeated for all the remaining images.

When in one image no feature adjusts to an existing plane it means that this plane has gone out the field of view. We define this plane as *closed* and it is no longer considered in the matching process, which helps to speed up the process. The list of identifiers of the features in every image can also be used to keep a feature voting system among the images. When one plane is closed, all the features that have not received enough votes compared with the number of images where the plane has been visible are discarded. So, the method deletes possible outliers that may have passed the previous constraints. Later we will discuss about the planes that have been seen more than once so that they appear repeated in the set of planes.

B. Constraints between planes

Additional constraints can be imposed in order to obtain more reliable results in the segmentation. Since most part of the images will contain two or more planar regions, multi-plane constraints can be a very useful tool to improve the results. 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 π^m and π^n are both visible in $\mathcal{I}_i$ and $\mathcal{I}_j$. The homology is obtained by multiplying one of the homographies with the inverse of the other one, $\boldsymbol{H}^{mn}_{ij} = (\mathbf{H}^m_{ij})^{-1}\mathbf{H}^n_{ij}$. The homology has some properties that can be useful for our purpose. Using the Sherman-Morrison formula [15], as in [16], the homology matrix can be decomposed in $\boldsymbol{H}^{mn}_{ij} = \mathbf{I} + \mathbf{v}\mathbf{p}^T$, where

$$\mathbf{v} = (v_1, v_2, v_3)^T = \mathbf{K}\frac{\mathbf{R}^{-1}_{ij}\mathbf{t}_{ij}}{1 + \frac{(\mathbf{n}^m_j)^T}{d^m_j}\mathbf{R}^{-1}_{ij}\mathbf{t}_{ij}} \tag{2}$$

is a view dependent vector and

$$\mathbf{p} = (p_1, p_2, p_3)^T = (\frac{(\mathbf{n}^m_j)^T}{d^m_j} - \frac{(\mathbf{n}^n_j)^T}{d^n_j})\mathbf{K}^{-1} \tag{3}$$

is a plane dependent vector. The homology can be used to separate real planes from false and repeated ones. This is done using its eigenvalues. The eigenvalues of a correct homology must have the form $(1, 1, 1 + v_1p_1 + v_2p_2 + v_3p_3)$. This knowledge is used by the proposed algorithm to classify the extracted planes. 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 we have an homography that is not describing a real plane. In this second case the new plane is ignored. The more planes visible in two images the more homology tests will be computed and the results will be better. 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.

III. Creation of the topological maps of planar regions

The idea of sorting the planes extracted from the reference images in a hierarchical structure to connect them has a great interest. An easy and formal way to do this is using a graph. One graph, $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, can be represented by a finite non empty set of nodes $\mathcal{N}$ and a set of edges $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ that connect the elements of the set of nodes. The set of edges is usually represented by its adjacency matrix, $\mathbf{A}$, a $|\mathcal{N}|$x$|\mathcal{N}|$ matrix, where the element $\mathbf{A}(m,n)$ shows the relation that exists between the nodes m and n of the set of nodes. In our case each node represents a plane of the scene whereas an edge $\mathcal{E}(m,n)$ between two nodes shows if the corresponding planes are co-visible or not. Two planes π^m and π^n are co-visible if and only if it is possible to compute an homology ($\boldsymbol{H}^{mn}$) between them in the set of reference images, which means that there are at least two consecutive images with enough features of both planes. The idea of co-visibility has a great interest in navigation tasks. If the robot has localized one plane of the set it will know which other planes it might see next to it when it moves, and so the space searched during the navigation will be reduced. In the following, instead of considering the set of edges we will work with the adjacency matrix, $\mathbf{A}$. We choose this representation because some operations in the graph, such as adding a new plane to the graph or merging two planes can be done easily by multiplying different matrices. With

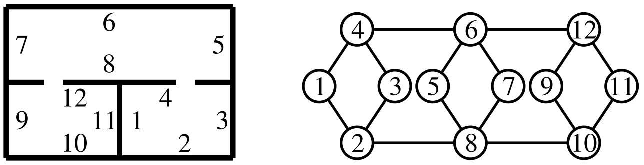

Fig. 2. Example of a 2D map with the planes numbered (left) and the topological graph of planes associated (right)

this representation if the planes m and n are co-visible then $\mathbf{A}(m,n)=1$, whereas it will be equal to 0 otherwise. An example of a topological map made by planes can be seen in Fig. 2.

Due to the definition of the nodes and the edges, all the created graphs will have two properties. Firstly, all the graphs will be undirected graphs which means that the adjacency matrix will be symmetric because if we can compute any homology, $\boldsymbol{H}^{mn}$, its inverse also exists. The second property is the absence of self loops in the graph, since it is not possible to compute an homology with only one homography.

In order to create the graph, the set of reference images is explored. For every image the method looks for the planes in it. Those which were not yet in the graph are added. The chosen information to represent one plane is the following:

- The identifiers of the images where the plane has been seen (the images are not necessary but the identifiers are required in order to determine where two planes have been seen together). One of the images is chosen as the reference for the plane.
- The set of features that belong to the plane. The coordinates of each feature are expressed in the image chosen as reference for the plane.
- The homographies that transform the features from any image where the plane has been seen to the reference image of the plane.

The node n is formally added to the graph by

$$\begin{cases} \mathbf{A} = [\mathbf{I}_{\mathcal{N}} \mid \mathbf{0}]^T \ \mathbf{A} \ [\mathbf{I}_{\mathcal{N}} \mid \mathbf{0}], \\ \mathcal{N} = \mathcal{N} \cup \{\pi^n\}, \end{cases} \tag{4}$$

with $\mathbf{I}_{\mathcal{N}}$ the identity matrix of $|\mathcal{N}| \times |\mathcal{N}|$ dimensions and $\mathbf{0}$ a null vector of dimension $|\mathcal{N}|$. Finally for all those planes which have been detected together in two images (it exists a correct homology between them) the algorithm sets the edges connecting them to 1:

$$\mathbf{A}(m,n) = \mathbf{A}(n,m) = 1 \Leftrightarrow \exists\, \mathcal{I}_i \in \mathcal{I} \mid \exists\, \boldsymbol{H}^{mn}_{i-1i} \tag{5}$$

Let us notice that this process can be done simultaneously with the extraction of planes from the images.

A. Fusion of planes

It may happen that in the sequence of images one plane leaves the field of view and later enter again in it (loop closing). If this happens the proposed segmentation algorithm will find twice the same plane so the graph will contain two nodes which actually are the same one. In order to solve this problem, once all the planes have been extracted and stored in the graph a fusion algorithm is run to merge the repeated planes. The method matches the features of every couple of planes $(\boldsymbol{\pi}^m, \boldsymbol{\pi}^n)$ which are not co-visible, $\mathbf{A}(m,n)=0$, and tries to compute a robust homography between them using DLT+RANSAC. If that 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 consist in the creation of a new graph $\mathcal{G}' = (\mathcal{N}', \mathbf{A}')$ erasing from $\mathcal{N}$ the repeated node. The edges that contain the erased node must be trespassed to the remaining nodes in such a way that for those nodes l such that $\mathbf{A}(n,l)=1$ then $\mathbf{A}(m,l)=1$, and for those that $\mathbf{A}(l,n)=1$ then $\mathbf{A}(l,m)=1$. This is formally described with the following expressions

$$\begin{aligned} \mathcal{N}' &= \mathcal{N} \setminus \{\pi^n\} \\ \mathbf{A}' &= \mathbf{I}_n(\mathbf{A} \vee \mathbf{P}_{mn}\mathbf{A} \vee \mathbf{A}\mathbf{P}_{mn})\mathbf{I}_n^T, \end{aligned} \tag{6}$$

where $\mathbf{P}_{mn}$ is a permutation matrix of the rows m and n and $\mathbf{I}_n$ is an identity matrix where the n^{th} row has been deleted. The symbol $\vee$ 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 $\{0,1\}$.

B. Overview of the method

All the method can be summarized in the Algorithm 1

Algorithm 1 Overview of the algorithm

```
Extract planes from $\mathcal{I}_1$ and $\mathcal{I}_2$
Create $\mathcal{G}$ with the initial planes
$\mathbf{A}(m,n) = \mathbf{A}(n,m) = 1\ \forall m \neq n$
for $\mathcal{I}_i$ $i = 3..n$ do
    Match features in $\mathcal{I}_{i-1}$ and $\mathcal{I}_i$
    for all opened $\pi^m \in \mathcal{N}$ do
        Select the matches that belong to $\pi^m$
        Compute $\mathbf{H}^m_{i-1i}$ and $\mathbf{H}^m_{r^m i}$ with DLT+RANSAC
        Add new features to $\pi^m$ from image $i$ using $\mathbf{H}^m_{r^m i}$
    end for
    Search for new planes in the remaining matches
    Add the new planes to $\mathcal{G}$ (eq. (4))
    Modify $\mathbf{A}$ with the new homologies (eq. (5))
end for
for all $\pi^m, \pi^n \in \mathcal{N}$ do
    if $\mathbf{A}(m,n) = 0$ then
        Try fusion of $\pi^m$ and $\pi^n$ (eq. (6))
    end if
end for
```

IV. Experimental Results

Several experiments have been carried out in order to analyze the properties and the behavior of our proposal. We have tested the methods using different data sets composed by real images. The data sets correspond to different locations where the proposals may be useful due to the plenty of planar regions.

The first data set has been recorded inside of one of the authors home (House data set), so it is a realistic scenario.

The camera used has been a *Panasonic Lumix FX-500* camera freely moved with 6DOF and 3600 frames have been stored from different rooms. The second data set has been recorded outside the building where we work (Ada Byron data set) moving a robot with a *Canon VCC4* camera onboard. The set contains 900 images captured along a planar path on the floor. In all the cases we have used SURF features [1] for the matching and the computation of the homographies. If the camera frame rate is high enough the motion between the images will be small and will be closer to a rotation motion. When the camera motion is a pure rotation all the features can be fitted in the same homography and the plane segmentation will fail. To avoid this problem we have followed the idea of [12] to select key frames among the sequence:

- There are as many images as possible between the key frames $\mathcal{I}_i$ and $\mathcal{I}_{i-1}$.
- There are at least M matches between the key frames $\mathcal{I}_i$ and $\mathcal{I}_{i-1}$.
- There are at least N matches between the key frames $\mathcal{I}_i$ and $\mathcal{I}_{i-2}$.

With this selection there is a higher chance to avoid pure rotations and then to obtain better results. We have divided the results in two subsections, one concerning the segmentation of the planar regions and the other deals with the construction of the graph of planes. An additional subsection shows how the new organization improves robot global localization in the environment, even when using images captured with a different camera than the one used for recording the reference sequence.

A. Segmentation of planar regions

We have tested our approach to segment the planar regions in both data sets getting good results. The House sequence has helped us to test the plane detection 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 Ada Byron sequence has less planar regions than the house sequence but the planes are seen in more images and the initial observed region of the planes usually has no overlapping with the final observed region of the same plane. With this sequence we have tested how the method tracks and grows the number of features of the planes. Fig. 4 shows two extracted facades. The results are not as precise as in the other sequence, but this is caused by the relative distance to the planes with respect to the distance between consecutive images. Even when there are more outliers the results are still quite good and, what is more important, we observe that the method follows the planar regions correctly adding new features as they appear.

B. Topological map of planar regions

Using the planar regions extracted with our algorithm from the data sets, we have computed the associated graph of planes using the technique proposed in section III. We have compared the resulting graphs with the graphs made by images [17] imposing the homography constraint between frames to observe the pros and cons of using one approach or the other. Tables I and II show the comparison for the House sequence and the Ada Byron sequence respectively. 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 was drastically reduced using our approach. Finally, the time required for computing the graph of planes ($\simeq$ 300 sec) is quite smaller than the time required for the graph of images ($\simeq$ 600 sec). This reduction is caused because in order to construct the graph of images it is necessary to compare every image with each other (quadratic cost in the number of images) whereas for the construction

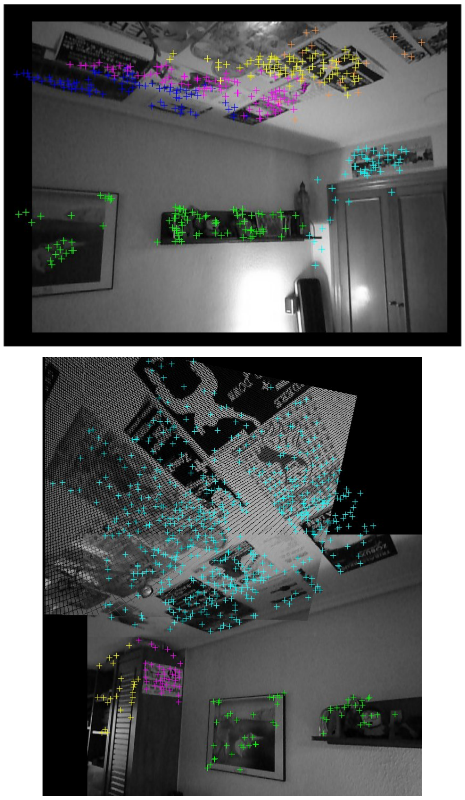

Fig. 3. Planar regions extracted from the House sequence. Each region is represented with a different color. In the bottom figure we have added points corresponding to the ceiling transformed with the computed homography.

TABLE I
RESULTS FOR THE HOUSE SEQUENCE

	Nodes	Edges	Features	Feats/node	Size (MB)
Image Graph	140	1728	78124	558	29.0
Plane Graph	30	78	10832	361	4.0

TABLE II
RESULTS FOR THE ADA BYRON SEQUENCE

	Nodes	Edges	Features	Feats/node	Size (MB)
Image Graph	181	9062	117987	651	43.9
Plane Graph	22	88	4589	208	1.7

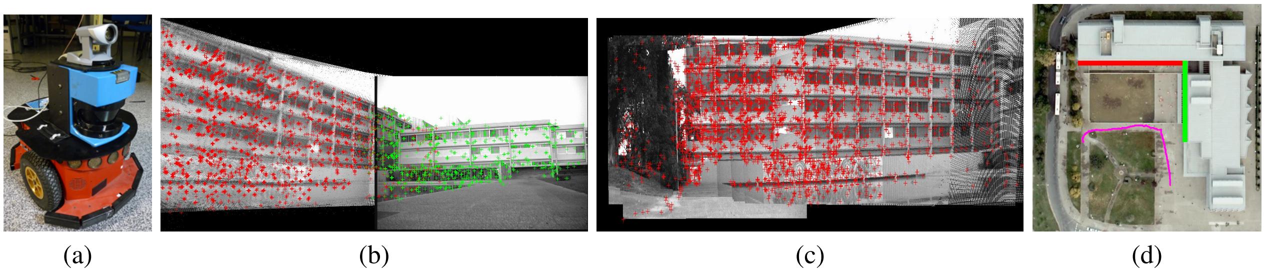

Fig. 4. (a) Robot used in the experiment. (b) Planar regions extracted from the Ada Byron sequence. (c) One planar region seen in the reference image (20 images are overlapped). (d) Trajectory followed by the robot (magenta line) computed using a structure from motion method.

TABLE III

RESULTS LOCALIZING THE ROBOT

	Mean time used in the localization
Images Graph	62 sec
Planes Graph	5 sec

of the graph of planes we just compare each image with the previous and the next one in the sequence (linear cost). If one plane appears twice or more it is merged in the final fusion stage. The only drawback of the new approach is that the size of each node 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 thousands of features and many homographies.

C. Localizing the robot

One last experiment was performed to measure the time used to localize the robot within the environment (the kidnapped robot problem). We have taken some images of the Ada Byron building using the *Panasonic Lumix FX-500* camera (the reference sequence was taken with the *Canon*) and we have searched among the set of planes/images selecting the most similar. The results are shown in Table III. The time invested searching in the set of planar regions was smaller than the time invested in the set of images whereas the accuracy remains the same (in all the cases the plane/image selected from the reference set contained high overlapping with the captured image). Since in this work we are not considering any metric information about the scene the localization algorithm only finds the planar region with more matches in the current image. In order to have a metric comparison of the accuracy in the localization metric information about the parameters of the planes should be computed.

V. CONCLUSIONS

We have presented an algorithm for topological map building using planar regions. The planar regions are extracted from a set of ordered images taking advantage of the information about planes in previous images performing a triple matching Plane-Image-Image that allows to track and grow the plane regions as new areas of the plane become visible. Extracted planar regions are used to build a graph, which can be built simultaneously to the extraction. The idea of using planes instead of images presents several advantages. It reduces the size of the graph and also its complexity. In addition, the knowledge of the planes in the scene can improve the robot localization as well as other tasks. Experimental results with real images indoors and outdoors show the good performance of the method.

REFERENCES

[1] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, 2006.

[2] J. Chen, W.E. Dixon, M. Dawson, and M. McIntyre. Homography-based visual servo tracking control of a wheeled mobile robot. *IEEE Transactions on Robotics*, 22(2):406–415, 2006.

[3] J. Courbon, Y. Mezouar, and P. Martinet. Indoor navigation of a non-holonomic mobile robot using a visual memory. *Autonomous Robots*, 25(3):253–266, 2008.

[4] Y. Fang, D.M. Dawson, W.E. Dixon, and M.S. de Queiroz. Homography-based visual servoing of wheeled mobile robots. *41st IEEE Conference on Decision and Control,*, 3:2866–2871 vol.3, 2002.

[5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, 1981.

[6] F. Fraundorfer, K. Schindler, and H. Bischof. Piecewise planar scene reconstruction from sparse correspondences. *Image and vision computing*, 24:395–406, 2006.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2000.

[8] G. Lopez-Nicolas, C. Sagues, and J.J. Guerrero. Homography-based visual control of nonholonomic vehicles. *IEEE International Conference on Robotics and Automation*, pages 1703–1708, 2007.

[9] J.F. Menudet, J.M. Becker, T.Fournel, and C.Mennessier. Plane-based camera self-calibration by metric rectification of images. *Image and Vision Computing*, 2007.

[10] A. C. Murillo, C. Sagüés, J. J. Guerrero, T. Goedemé, T. Tuytelaars, and L. Van Gool. From omnidirectional images to hierarchical localization. *Robotics and Autonomous Systems*, 55(5):372–382, 2007.

[11] A. Remazeilles and F. Chaumette. Image-based robot navigation from an image memory. *Robot. Auton. Syst.*, 55(4):345–356, 2007.

[12] E. Royer, M. Lhuillier, M. Dhome, and J.M. Lavest. Monocular vision for mobile robot localization and autonomous navigation. *Int. J. Comput. Vision*, 74(3):237–260, 2007.

[13] C. Sagüés, A.C. Murillo, F. Escudero, and J.J. Guerrero. From lines to epipoles through planes in two views. *Pattern Recognition*, 39(3):384–393, 2006.

[14] G. Silveira, E. Malis, and P. Rives. Real-time robust detection of planar regions in a pair of images. In *IROS*, pages 49–54, 2006.

[15] S. A. Teulosky, W. T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The art of Scientific Computing*. Cambridge University Press, Cambridge, 2002.

[16] L. Zelnik-Manor and M. Irani. Multiview constraints on homographies. *IEEE Tra. on Pattern Analysis and Machine Intelligence*, 24(2):214–222, 2002.

[17] Z. Zivkovic, O. Booij, and Ben Kröse. From images to rooms. *Robot. Auton. Syst.*, 55(5):411–418, 2007.