Dense Multi-Planar Scene Estimation from a Sparse Set of Images

Alberto Argiles, Javier Civera and Luis Montesano

Abstract— **Ego-motion estimation and 3D scene reconstruction from image data has been a long term aim both in the Robotics and Computer Vision communities. Nevertheless, while both visual SLAM and Structure from Motion already provide an accurate ego-motion estimation, visual scene estimation does not offer yet such a satisfactory result; being in most cases limited to a sparse set of salient points. In this paper we propose an algorithm to densify a sparse point-based reconstruction into a dense multi-plane based one, from the only input of a set of sparse images.**

The method starts by recovering a sparse set of 3D salient points and uses them to robustly estimate the dominant planes of the scene. The number of planes is not known in advance and there may exist outliers from the planes in the point cloud. In a second step, the image data and the estimated 3D structure are combined to determine which parts of each plane actually belong to the scene exploiting photoconsistency and geometrical constraints.

Experimental results with real images show that the described approach achieves accurate and dense estimation results in man-made environments. Moreover, the method is able to recover areas without texture, where usually there are no salient points.

I. INTRODUCTION

As an essential component of many robotic systems, automated 3D model building has received considerable attention in the last two decades. Current systems have evolved from 2D maps based on laser measurements [15] to 3D representations that exploit the much richer information coming from cameras. As a result, the Simultaneous Localization and Mapping (SLAM) [4] has now deep connections with Structure from Motion research from the Computer Vision community [10]; which also aims to estimate camera motion and 3D scene being a set of images the only sensorial input.

Most of the times, both Structure from Motion and visual SLAM are based on extracting salient visual features (most commonly points, e.g. SIFT [13] or FAST [17]) and matching them across images, which provides the constraints for camera motion and 3D feature positions estimation. As a result, camera motion can be very accurately estimated; but scene estimation is limited to a sparse set of salient point features. Typically, areas lacking texture –like white walls– cannot be estimated. Also, there is no higher level information about the underlying structure of the environment, which could be exploited by a robot to accomplish its task.

An alternative to this methods is to extract geometric features such as lines or planes using, for example, RANSAC like algorithms. However, this type of approach still does not capture important characteristics of the scene for robot operation. In particular, recovered models are usually continuous (i.e. without considering the presence of gaps or multiple surfaces within the same model).

This paper presents an algorithm that provides a dense reconstruction in environments that are composed of several dominant planes. The planar assumption allows for a compact representation of many man made environments such as indoor scenarios or buildings. The algorithm receives as input a sparse set of partially calibrated images. The specific number of planes in the scene is not known in advance. The method combines state-of-the-art multi-model fitting techniques with geometric photoconsistency constraints to recover large planar patches even on those areas without texture that pose difficulties to salient point-based algorithms. The method has been evaluated on real images obtained with a camera in indoor and urban environments. The experimental results show that by fusing multi-model estimation and photoconsistency it is possible to extract the main planar regions from a set of images.

The paper is organized as follows: section II discusses several references related to our work. In section III, an overview of the method is given together with the notation used along the paper. The next three sections present the three main steps of the algorithm: section IV addresses the initial estimation of 3D sparse points; section V describes the robust multiple model algorithm that extracts planes from the initial sparse 3D structure; and finally, section VI describes the densification process using the planes, photometric image data and visibility constraints. Section VII shows the experimental results using real imagery and section VIII presents the conclusions and future work.

II. RELATED WORK

As mentioned above, related work comes mainly from two different disciplines. On one hand, mapping and SLAM have often provided high level 2D features based on laser measurements such as segments [2]. In the 3D case, there are some works that tried to construct a dense reconstruction based on planes from laser measurements [14]. As they are using laser sensing, all these works have as starting point a dense 3D point cloud. 3D visual sensing, at its current state, is only able to provide a sparse point cloud limited to textured areas; being necessary a more sophisticated processing.

Very recently, there has been some effort to upgrade the usual sparse point-based visual SLAM scene estimation into a higher geometric level. For example, [19] presents a line-based reconstruction and [9] incorporates planes, both of

A. Argiles, J. Civera, and L. Montesano are with Robotics, Perception and Real-Time Group, Aragon Institute of Engineering Research (I3A), Universidad de Zaragoza, 50018 Zaragoza, Spain alargi@gmail.com, jcivera@unizar.es, montesano@unizar.es

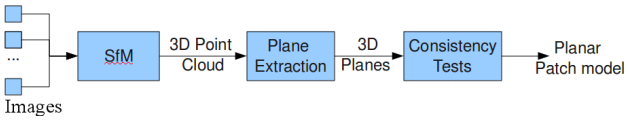

Fig. 1. Overview of the approach.

them based on a EKF monocular SLAM. [12] incorporates edgelets and points in his SLAM framework based on optimization over keyframes.

On the other hand, the computer vision community has also studied the problem of dense reconstruction from sparse images, specially for architectural purposes. Most of these works take as starting point a sparse 3D structure estimation and image registration; very similarly to what we do. For instance, [5] combines such a reconstruction with a multi-view stereo algorithm [7] and a depth grid map integration algorithm for Manhattan worlds.

Perhaps one of the most successful methods is the so-called Clustering Views for Multi-view Stereo (CMVS) [6]. Starting from salient points, it uses small patches around them to densify the reconstruction and provides a dense set of 3D points with texture. The method has proven to be very powerful to densely estimate large scenes from large sets of images. In contrast with the other cited works, this is a textured cloud of points without structure. [16] is a promising work also estimating an accurate 3D point cloud from an initial sparse one, but using optical flow equations. The efficiency is its most remarkable feature, being able to run in real-time.

A different trend of methods try to extract high level features such as planes from points extracted from images. The main difficulty here is that the scene contains an unknown number of models. Multi-model RANSAC [11] applies single-model standard RANSAC sequentially. Points marked as outliers are used in following iterations to discover new models. Sequential RANSAC has been succesfully applied in [8] as a first stage for planar surface extraction from stereo data. Nevertheless, this algorithms have two limitations: 1) they are dependant on the order in the plane extraction; and 2) the number of planes have to be known in advance.

Parallel implementations of Multi-model RANSAC, like [23], overcome these two limitations. In this paper we are using one of these parallel implementations, called J-Linkage algorithm [20]. This algorithm combines sampling models from the cloud of points with an agglomerative clustering of these hypotheses. Results have shown better performance than RANSAC or Hough based algorithms.

The multiple-model estimation algorithms described in the previous paragraph do not consider discontinuities in the fitted models and, therefore, fail to provide an accurate description of the environment. [3], [21] are two very recent approaches combining the J-Linkage technique with geometric and photometric tests, being both of them very similar to the algorithm presented in this paper.

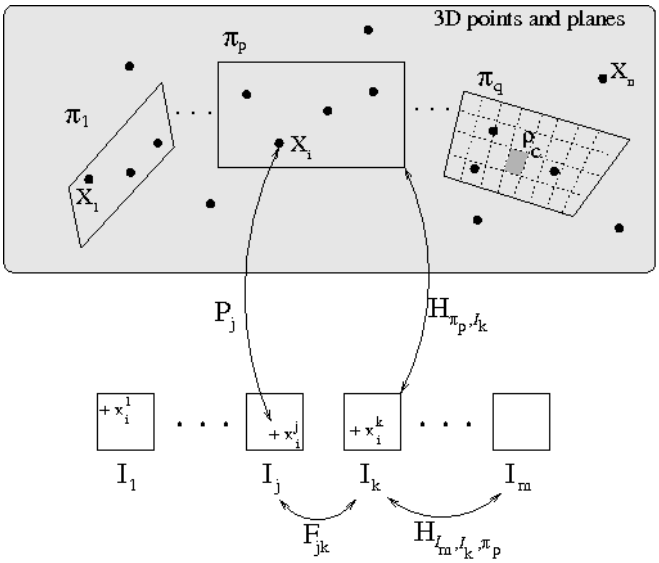

Fig. 2. Notation convention for the main entities appearing in the paper.

III. GENERAL APPROACH AND NOTATION

In this section, we provide a general overview of the proposed method and introduce the notation used along the rest of the paper. Figure 1 describes the three main steps of the method, while Fig. 2 depicts the relations between the main entities involved in the planar reconstruction process.

The algorithm takes as input a set of m images, $< I_1, \ldots, I_j, \ldots, I_k, \ldots, I_m >$, of the scene. The first step is to recover a set of n 3D salient point features $\mathbf{X} = \begin{pmatrix}\mathbf{X}_1^\top & \ldots & \mathbf{X}_i^\top & \ldots & \mathbf{X}_n^\top\end{pmatrix}^\top$. $\mathbf{X}_i$ stands for the 3D position of the point i referred to a global reference frame W. In general, we will use homogeneous coordinates rather than Euclidean ones to describe points and planes, since they fit better the projective nature of camera sensing.

In the second step, the interest points $\mathbf{X}$ are used to extract a set of q dominant planes $\pi = \begin{pmatrix}\pi_1^\top & \ldots & \pi_p^\top & \ldots & \pi_q^\top\end{pmatrix}^\top$. The number of planes is not known in advance by the algorithm and there may be points $\mathbf{X}_i$ that do not belong to any dominant plane π_p.

Once the main planes of the scene have been recovered, the last step uses photoconsistency and visibility constraints to determine which parts of the plane are actually supported by the images of the scene and to provide a dense representation of the scene. To achieve this, each plane is divided in a set of rectangular cells, $\mathcal{C} = \{\rho_1, \ldots, \rho_c, \ldots, \rho_r\}$. The constraints are checked cell-wise and the final planar model is composed only by those that satisfy both constraints.

The described process exploits several projective models and geometric properties (see Fig. 2). The projection of each 3D salient point will appear in one or more of the images. We denote as $\mathbf{x}_i^j$ the projection of the 3D point $\mathbf{X}_i$ on the image I_j. We are modeling such projection with the standard pinhole camera model, following $\mathbf{x}_i^j = \mathbf{P}_j\mathbf{X}_i$ where $\mathbf{P}_j$ stands for the projection matrix. The projection matrix includes the internal camera calibration matrix $\mathbf{K}_j$ and the

rotation and translation $\mathbf{R}_{C_j}^W$ and $\mathbf{t}_{C_j}^W$ with respect to the global reference frame W.

$$\mathbf{P}_j \;=\; \mathbf{K}_j \;(\mathbf{I}_{3\times 3}\;\mathbf{0}_{3\times 1}) \begin{pmatrix} \mathbf{R}_{C_j}^W & \mathbf{t}_{C_j}^W \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} . \quad (1)$$

The internal calibration matrix $\mathbf{K}_j$ includes the focal length $(f_{x,j}\ f_{y,j})^\top$ and the principal point $(x_{0,j}\ y_{0,j})^\top$ both in terms of pixel dimensions and in the horizontal direction x and vertical direction y in the image

$$\mathbf{K}_j \;=\; \begin{pmatrix} f_{x,j} & 0 & x_{0,j} \\ 0 & f_{y,j} & y_{0,j} \\ 0 & 0 & 1 \end{pmatrix} . \quad (2)$$

Radial distortion is also considered, so image points are undistorted previous to the application of projective models.

The Fundamental Matrix encapsulates the intrinsic geometry between two generic views I_j and I_k is written as $\mathbf{F}_{j,k}$. It is the main relation used to extract the sparse 3D points and to compute the camera locations. Plane-image and image-image homographies, encoding the projective mapping between planes, will be used in the last step of the algorithm. $\mathbf{H}_{\pi_p,I_j}$ stands for the homography transforming a point belonging to the plane π_p expressed in 2D plane coordinates into its position within image I_j. $\mathbf{H}_{I_j,I_k,\pi_p}$ stands for the homography induced by the plane π_p, that transform a 2D point belonging to the plane π_p imaged in I_j to its image position in I_k.

IV. SPARSE POINT-BASED SCENE RECONSTRUCTION

The objective of this first stage of the algorithm is to estimate, from a set of m images, a sparse 3D reconstruction of salient points and the camera motion between images. The process starts by extracting salient points for each image and matching them between pairs of images. SIFT features [13] are used here, due to their good invariance properties to projective transformations. The correspondence search is sped up by using approximate nearest neighbor [1].

The resulting matches are then used to compute the Fundamental matrix between pairs of images $\mathbf{F}_{jk}$, which encodes the motion from camera j to camera k and the internal calibration of both cameras. Each pair of points adds a constraint on the Fundamental matrix as follows

$$\mathbf{x}_i^{j\top} \mathbf{F}_{jk} \mathbf{x}_i^k = 0 \ ; \quad (3)$$

where $\mathbf{x}_i^j$ is the point i in image j, $\mathbf{x}_i^k$ is the point i in image k. The eight-point algorithm [10] is used to estimate this fundamental matrix, using RANSAC for robustness against outliers. A prior guess for the focal length of the camera is extracted from the EXIF tag of each image.

The resulting motion estimations -camera position $\mathbf{t}_j^W$ and rotation $\mathbf{R}_j^W$ for each image j–, internal camera calibration – focal length f_j for each image j– and 3D point estimation $\mathbf{X}_i$ are used as initial seed for a Bundle Adjustment [22]. Bundle Adjustment outputs the maximum likelihood estimation by minimizing the reprojection error

$$\Delta \mathbf{x}_i^j = \mathbf{x}_i^j - \mathbf{h}\left(\mathbf{t}_j^W, \mathbf{R}_j^W, \mathbf{K}_j, \mathbf{X}_i\right) \ , \quad (4)$$

where $\mathbf{h}$ is the function modelling the projection of point $\mathbf{X}_i$ in image I_j. This part of the algorithm makes use of the software package Bundler [18], a state-of-the-art implementation of the algorithm described above.

V. ROBUST MULTIPLE-PLANE ESTIMATION

The 3D point cloud $\mathbf{X}$ already contains sparse information about the structure of the environment. This section describes how to recover from this 3D point cloud the set of dominant planes within the scene. As mentioned before, the main challenge here is the unknown number of planes and the presence of outliers.

The robust estimation of multiple planes π_m from noisy 3D points $\mathbf{X}_i$ is performed using J-Linkage [20]. As mentioned in Section II, this algorithm has shown better performance than other approaches to multi-model extraction. The algorithm basically works in two steps. First, a large number of plane hypothesis are created by sampling random minimal subsets of data points, very similar to RANSAC algorithms. For each plane hypothesis, its consensus set is computed. The consensus set is composed by the 3D points whose distance to the plane hypothesis is under a predefined threshold ϵ. Similarly, each point has a preference set, i.e. the set of sampled planes with distance below ϵ.

In a second step, differently from RANSAC, an agglomerative clustering technique is used to cluster plane hypotheses, starting from all sampled hypotheses, and extract the planes from the scene. The key of the algorithm is the distance used in the clustering process. Each cluster hypothesis has a preference set equal to the the intersection of the preference sets of all the models within the cluster. The similarity between clusters is computed based on the the Jaccard distance between the preference set of clusters. The use of this distance ensures that within each cluster, there is a model voted by all the points belonging to this cluster; and that there is no model that can share all the points of two different clusters (see [20] for details). As usual, the parameters of the resulting plane for each cluster are estimated using least squares with all the points associated to the cluster. We used the open-source implementation provided by the authors.

VI. CONSISTENCY TESTS

A. Photoconsistency

The last step of the method aims to find the actual structure of the recovered planes. A simple solution would be to recover the convex hull of the points associated to each plane. This approach has two important limitations. First, it fails to consider structure outside the convex hull of salient points and, therefore, cannot incorporate areas without texture to the plane. Second, it cannot deal with holes or separated multiple surfaces lying on a single plane. In order to provide a dense representation that overcomes

Algorithm 1 Photoconsistency

INPUT: X, π **,** $< I_1, \cdots, I_m >$

for each π_i **do**
  $Divide\ \pi_i\ into\ t \times t\ cells\ \mathcal{C} = \{\rho_1,\ \ldots,\ \rho_c,\ \ldots,\ \rho_r\}$
  $\mathcal{S}_i = \{\rho_c \mid \exists \mathbf{X_j} \in \rho_c\}$
  $\mathcal{E} = neighbors(\mathcal{S}_i)$
  while $\mathcal{E} \neq \emptyset$ **do**
    $Select\ \rho_c\ from\ \mathcal{E}$
    $f_r = \mathbf{H}_{\pi_i, I_r} \rho_c \quad \%\ Reference\ Image$
    for $I_j \neq I_r$ **do**
      $f_j = \mathbf{H}_{I_j, I_r, \pi_i} \mathbf{H}_{\pi_i, I_j} \rho_c$
    end for
    if $\neg \exists (jk) \mid crosscorr(f_j, f_k) < \lambda$ **then**
      $\mathcal{S}_i = \rho_c \cup \mathcal{S}_i$
      $\mathcal{E} = neighbors(\rho_c) \cup \mathcal{E}$
    end if
  end while
end for

OUTPUT: $\mathcal{S} = \{\mathcal{S}_i\}$

Algorithm 2 Visibility

INPUT: X, $\pi, \mathcal{S}$

for all $\mathcal{S}_i$ **do**
  for all I_j **do**
    $\%Compute\ image\ with\ depths$
    $d_{ij} = dist(\mathbf{t}^W_{C_j}, \mathcal{S}_i)$
  end for
end for
for all $\mathcal{S}_i$ **do**
  $\mathcal{V}_i = \emptyset$
  for all $\rho_c \in \mathcal{S}_i$ **do**
    if $\exists j, j', j \neq j' \mid i = \arg\min_k d_{kj}(\mathbf{H}_{\pi_i, I_j} \rho_c) \wedge i = \arg\min_k d_{kj'}(\mathbf{H}_{\pi_i, I_j} \rho_c)$ **then**
      $\mathcal{V}_i = \rho_c \cup \mathcal{V}_i$
    end if
  end for
end for

OUTPUT: Visible cells $\mathcal{V} = \{\mathcal{V}_i\}$

these limitatios, each plane is divided into a set of small cells $\mathcal{C} = \{\rho_1,\ \ldots,\ \rho_c,\ \ldots,\ \rho_r\}$. Each cell is further composed by $t \times t$ pixels, $\rho_c = \{\rho_{c_{11}}, \cdots, \rho_{c_{tt}}\}$ where $\rho_{c_{ij}} = (u\ v\ 1)^T$ represent the homogeneous coordinates of the pixel within the plane.

Photoconsistency is used to decide whether the cell actually belongs to the world structure. Using an idea very similar to [6], the process starts from those cells containing 3D interest points and then proceeds to examine neighboring patches with a 8-neighbor strategy. For each plane, the process finishes when no new neighboring patches pass the photoconsistency check.

The consistency check is geometric and photometric. The pixels of plane cell ρ_c are projected to a reference image I_r through the homography $\mathbf{H}_{\pi, I_r}$ (see Fig. 2). Then, we use the image to image homography $\mathbf{H}_{I_j, I_r, \pi}$ to project the corresponding region of each image I_j to the reference image. If the plane cell actually belongs to the 3D plane in the scene, all the image patches will be equal. We use the normalized correlation to compare pairs of patches. A cell is marked as photoconsistent if the correlation of all pairs of patches are over a certain threshold λ. Otherwise, the cell is marked as non-photoconsistent and is not included in the final model. Algorithm 1 provides an overview of the photoconsistency test described in this section.

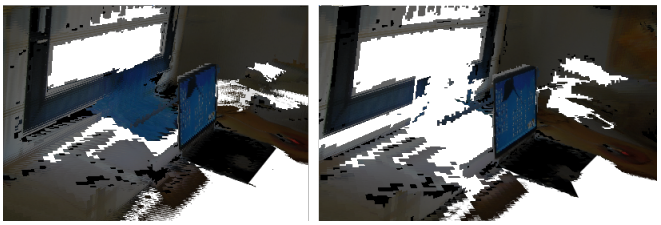

Fig. 3. Example of a real reconstruction without (left) and with (right) visibility constraints. Notice that the use of visibility constraints correctly removes the wrongly estimated surface that projects the laptop texture onto the table behind it.

For visualization purposes, during the photoconsistency test, we generate the cell texture from the corresponding image patches. A standard median filter is applied for each pixel of the cell.

B. Visibility constraints

The photoconsistency test does not take into account any visibility constraints. When two planes overlap, photoconsistency is prone to copy texture from the closest plane into the back one. This happens, for instance, when the image region lacks texture (see the left image in figure 3). Note how the blue monitor area has been (wrongly) projected into the table. To avoid this problem, we use the visibility constraints imposed by the 3D structure of the scene. The final result is shown in the right image of figure 3.

The visibility constraint has to be checked for all photoconsistent cells of each 3D plane. Using the homography $\mathbf{H}_{\pi_i, I_j}$, the photoconsistent cells $\mathcal{S}_i$ of plane π_i are projected onto each image I_j to create a depth image d_{ij} with the same size as image I_j containing at every pixel the distance from the 3D cells in $\mathcal{S}_i$ to the camera j according to its location $\mathbf{t}^W_{C_j}$.

Having these depth images for every plane, we look for cells occluded by other cells by comparing the corresponding distances to each camera. In general, multiple cameras will provide different results for the visibility constraint due to different occlusions according to their positions in the scene. Since examined patches have to be photoconsistent and visible in at least two images, we kept all patches which are visible at least from two cameras assuming these cameras have correctly reconstructed the patch. Algorithm 2 summarizes the visibility test proposed in this section.

VII. EXPERIMENTAL RESULTS

To evaluate the method, we have acquired three sets of images of environments where planar structures dominate.

(a) (b) (c)

Fig. 4. Four representative images out of (a) the 12 images of dataset 1 –office–; (b) the 16 images of the dataset 2 –apartment facades– ;and (c) the 14 images of the dataset 3 –industrial facades–.

Dataset 1 images an office scene. Figure 4(a) shows four pictures out of the 12 that are the input to our algorithm. The images are 2048×1536 and were taken with a standard consumer digital camera. The second and third dataset have 16 and 14 images respectively and correspond to outdoor scenes: dataset 2 was taken in an apartment building and dataset 3 in an industrial building. Figures 4(b) and (c) show four images of each dataset. In both cases the image resolution is 1632×1224 and were taken with the same consumer camera. During the description of the method, we have introduced a certain number of parameters that have to be tuned. Bundler was configured to extract as many SIFT points as possible, so as to increase the number of 3D points for the plane extraction. The rest of the parameters were those by default in the software package. For the J-Linkage parameters, 50000 plane hypotheses were generated before clustering and the distance threshold ϵ was set to 0.1. The size of the cells balances precision of the reconstructed model and computational load. We used a 8×8 cells in all our experiments. The cross-correlation threshold λ for the photoconsistency test was 0.6.

We analyze first the results for the office environment. Figure 5 shows the resulting scene estimated for the first dataset, a side view in Fig. 5(a) and a front view in Fig. 5(b). The texture for each planar cell has been drawn onto the cell to ease the visualization. Visual inspection shows the accuracy of the 3D reconstruction obtained, and how it reflects the dominant planar structure of the room. It is also noticeable that the curtain wall behind the laptop screen was entirely recontructed, even though this area is poorly textured.

For this first result, we will go through all the steps of the proposed algorithm with more detail. The Bundle Adjustment described in section IV estimates the relative motion of the 12 images and a sparse reconstruction composed of 2113 points. A 3D view of the reconstructed points is shown in Fig. 6. Notice that the sparse reconstruction contains many points in highly textured areas, mostly in the laptop screen. Nevertheless, areas lacking texture like white walls are not estimated by these standard methods based on salient point matching.

It is also worth remarking here again the main motivation for this work: a sparse structure estimation like the one in figure 6 would be rather useless for a robotic application –

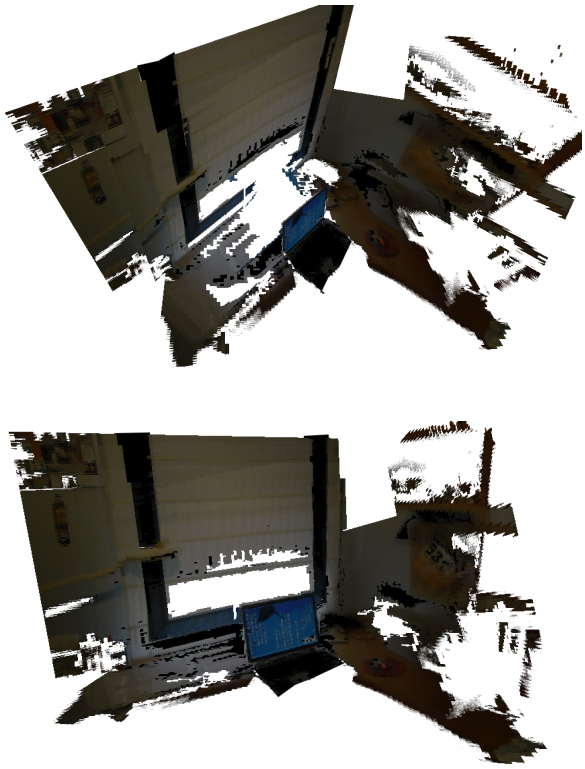

Fig. 5. Two views of the 3D dense multiplanar estimation of the scene corresponding to dataset 1 –office–. Figure is best seen in color.

e.g., navigation–; because we do not know how is the 3D scene between this sparse points. If we look again at figure 5, it is clear that this dense one should be preferred.

Figure 7 shows the 6 planes extracted from the set of 2113 points: one corresponding to the desktop plane, one to the laptop screen, two for the bookshelves wall (one for the bookshelves plane and one for the wall plane) and two for the curtain wall (one for the curtain and one for the wall).

Although the reconstruction at this step has a higher geometric level than the point-based one in figure 6; it can be seen that it does not offer yet a usable model of the scene. Looking closely at figure 7, it can be noticed that the boundaries for the planar surfaces are not defined. Checking the photometric and geometric constraints, as explained in section VI, allows to check which parts of each plane are in

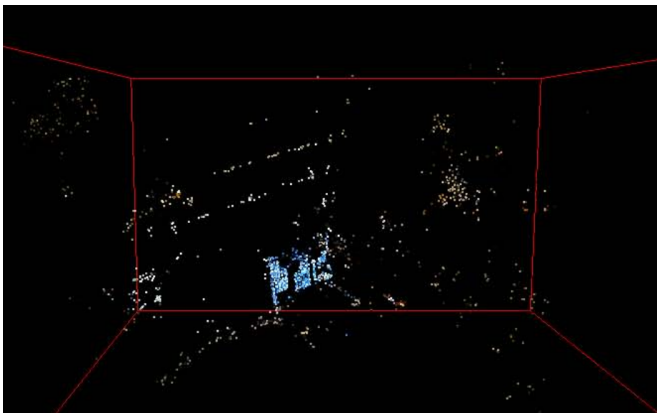

Fig. 6. Sparse 3D reconstruction of salient points corresponding to the office dataset.

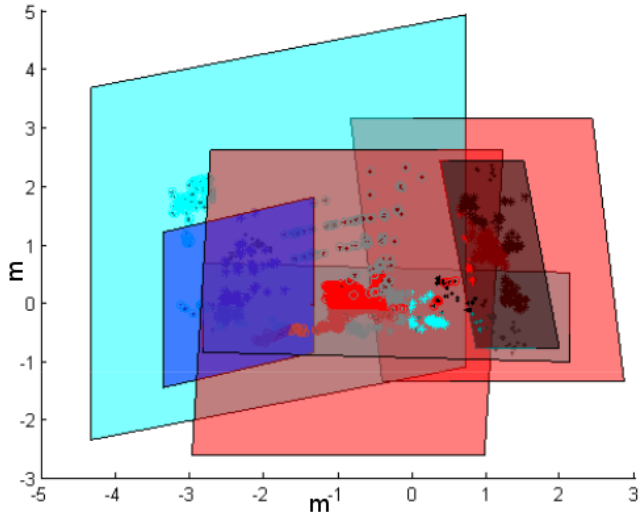

Fig. 7. Planes extracted by J-Linkage from the sparse 3D point cloud in figure 6 –office dataset.

fact seen by the cameras providing the final result as shown in figure 5.

Results for dataset 2 –apartment facades– are shown in Fig. 8. The number of 3D salient points extracted during the sparse reconstruction was 5076. From these points, the clustering algorithm produced 4 planes, corresponding to the different facades of the building including a plane for the balconies. Figure 8(a) shows the 3D points together with the recovered planes. The model failed to detect the ground due to the absence of salient points. Although it is not easy to realize it in the figure, the plane for the balconies is not correctly aligned with the wall. This is due to the fact that balcony points are aligned along two lines. The clustering algorithm included some other points which resulted in a bad plane orientation.

In the case of dataset 3 (Fig. 4(c), industrial facades), bundler extracted 2738 3D salient points from which the J-Linkage algorithm recovered 4 planes corresponding to the facades of the building. Figure 9 (a) shows the planes together with the sparse 3D points. Again, the ground did not contain enough salient points to extract a plane. The scene contains a glass corridor (see Figure 4(c)) between the walls that was not extracted. Due to the different reflections in each image, the photoconsistency test did not fill the back plane with any texture. This is the reason why the densification only worked on the top part of the buildings.

As a summary of the results, the densification process based on dominant planes is able to recover the main structure of the environment. The method is able to expand over non textured areas. Despite the use of a robust multi-plane fitting algorithm, results show that the method is still sensitive to outliers, specially for planes based on sets of points along a narrow area such as the balconies of dataset 3 or the shelves of dataset 1. Also, in all cases the reconstruction at the boundaries of the scene has lower quality due to the smaller number of images and points in those areas.

VIII. CONCLUSIONS AND FUTURE WORK

This paper describes an algorithm for dense 3D reconstruction of planar environments from the only input of a sparse set of images without any extra knowledge. First, a point-based sparse reconstruction is done using standard Structure from Motion algorithms: SIFT features extraction and matching, robust fundamental matrix estimation between pairs of images using RANSAC and non-linear optimization using all the images. After that, and based on this sparse 3D point reconstruction, a robust algorithm is used to extract the dominant planes of the scene. Finally, homographies are used to analyze the planes and to determine which parts are photoconsistent and visible from the cameras and hence belong to the real scene.

The experimental results show that the algorithm provides good performance in man-made environments: accurate and dense reconstructions can be obtained, even in cases where there are large untextured areas.

Several lines for future work arise from the results of this paper, but two of them are most remarkable. First, it would be interesting to code an efficient version of the presented algorithm for a computational cost analysis. The authors believe that none of the steps in the presented algorithm should be expensive: Pairwise Structure from Motion plus Bundle Adjustment have been adapted very recently to work sequentially in real-time at 30 frames per second [12]. Neither testing hypothesis for multiple models nor checking textures via homography should present a high computational complexity or cost for a local map estimation like the one presented in the experiments of this paper.

Secondly, although the planar assumption may roughly model indoors or man-made environments, such scenes are not entirely formed by planes. Having as starting point the planar-based reconstruction presented in this paper it would be very interesting to complete the non-planar parts using other approaches; for example Multi-view Stereo algorithms [7]. Compared to the plain use of Multi-View Stereo algorithms over the whole image area, introducing the extra knowledge about the planarity of most of the scene using

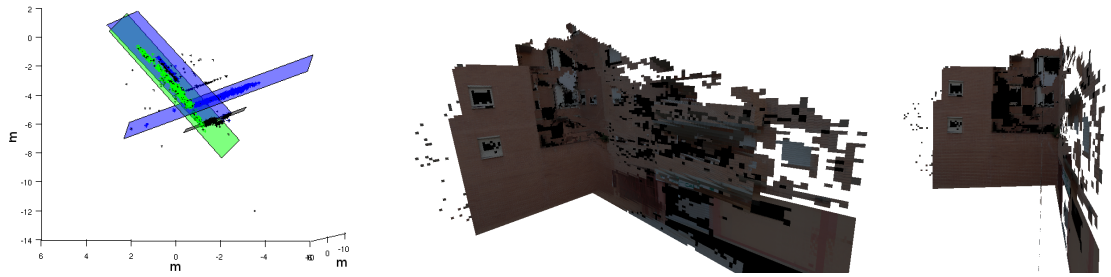

Fig. 8. Dataset 2 –apartment facades– results: a) recovered planes, b) 3D reconstructed model and c) other view of the reconstruction. Figure is best seen in color.

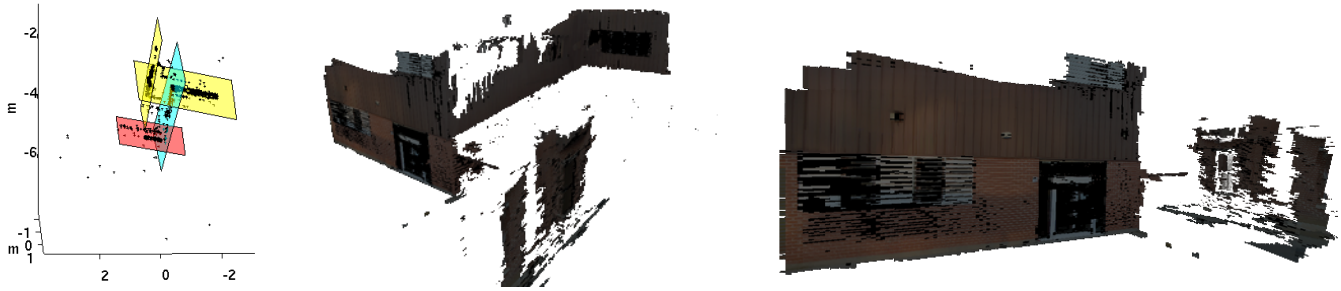

Fig. 9. Dataset 3 –industrial facades– results: a) recovered planes, b) 3D reconstructed model and c) other view of the reconstruction. Figure is best seen in color.

the algorithm presented in the paper could reduce the cost of Multi-View Stereo.

IX. ACKNOWLEDGMENTS

This work was partly supported by the Spanish DPI2009-07130 (Dirección General de Investigación of Spain) and European RoboEarth FP7-248942 projects. The authors would like to thank J. M. M. Montiel for fruitful discussions.

REFERENCES

[1] S. Arya, D. Mount, N. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.

[2] J. Castellanos and J. Tardos. *Mobile robot localization and map building: A multisensor fusion approach*. Springer Netherlands, 1999.

[3] A. B. D. Fouhey, D. Scharstein. Multiple Plane Detection in Image Pairs Using J-Linkage. In *20th International Conference on Pattern Recognition (ICPR 2010)*, pages 336–339, 2010.

[4] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robotics and Automation Magazine*, 13(2):99–110, 2006.

[5] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Reconstructing building interiors from images. In *Proceedings of the 12th IEEE International Conference on Computer Vision*, pages 80–87, 2009.

[6] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Towards Internet-scale Multi-view Stereo . In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[7] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pages 1–8, 2007.

[8] D. Gallup, J. Frahm, and M. Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*, pages 1418–1425, 2010.

[9] A. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas. Discovering Higher Level Structure in Visual SLAM. *IEEE Transactions on Robotics*, 24(5):980–990, 2008.

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2004.

[11] Y. Kanazawa and H. Kawakami. Detection of planar regions with uncalibrated stereo using distributions of feature points. In *British Machine Vision Conference*, pages 247–256, 2004.

[12] G. Klein and D. Murray. Improving the Agility of Keyframe-Based SLAM. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, pages 802–815. Springer, 2008.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[14] M. Martín Nevado, G. Gómez, et al. Obtaining 3D models of indoor environments with a mobile robot by estimating local surface directions. *Robotics and Autonomous Systems*, 48(2-3):131–143, 2004.

[15] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the National conference on Artificial Intelligence*, pages 593–598, 2002.

[16] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*, pages 1498–1505, 2010.

[17] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. *Proceedings of the 9th European Conference on Computer Vision*, pages 430–443, 2006.

[18] N. Snavely, S. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[19] J. Sola, T. Vidal-Calleja, and M. Devy. Undelayed initialization of line segments in Monocular SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent RObots and Systems*, pages 1553–1558, 2009.

[20] R. Toldo and A. Fusiello. Robust multiple structures estimation with J-Linkage. *Proceedings of the 10th European Conference on Computer Vision*, pages 537–547, 2008.

[21] R. Toldo and A. Fusiello. Photo-consistent Planar Patches from Unstructured Cloud of Points. *Proceedings of the 11th European Conference on Computer Vision*, pages 589–602, 2010.

[22] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – A modern synthesis. In *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[23] M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multiransac algorithm and its application to detect planar homographies. In *IEEE International Conference on Image Processing*, Sep 2005.