2013 IEEE International Conference on Robotics and Automation (ICRA)
Karlsruhe, Germany, May 6-10, 2013

Fast place recognition with plane-based maps

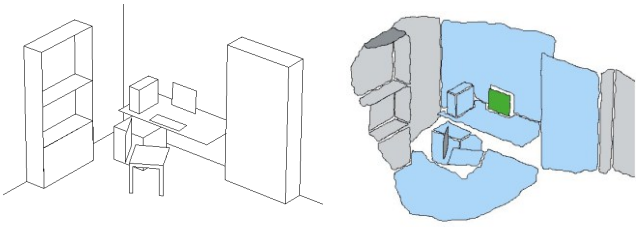
E. Fernández-Moral[1], W. Mayol-Cuevas[2], V. Arévalo[1] and J. González-Jiménez[1]

Abstract— **This paper presents a new method for recognizing places in indoor environments based on the extraction of planar regions from range data provided by a hand-held RGB-D sensor. We propose to build a plane-based map (PbMap) consisting of a set of 3D planar patches described by simple geometric features (normal vector, centroid, area, etc.). This world representation is organized as a graph where the nodes represent the planar patches and the edges connect planes that are close by. This map structure permits to efficiently select subgraphs representing the local neighborhood of observed planes, that will be compared with other subgraphs corresponding to local neighborhoods of planes acquired previously. To find a candidate match between two subgraphs we employ an interpretation tree that permits working with partially observed and missing planes. The candidates from the interpretation tree are further checked out by a rigid registration test, which also gives us the relative pose between the matched places. The experimental results indicate that the proposed approach is an efficient way to solve this problem, working satisfactorily even when there are substantial changes in the scene (lifelong maps).**

I. Introduction

THE ability to recognize a place previously visited is a major problem in mobile robotics since, among other things, it allows to accomplish topological localization and loop closure detection in SLAM. Most of the solutions to this problem have concentrated on exploiting appearance from intensity images [1], [2], [3], [4], [5], [6] and [7]. Though these methods work in many situations, they rely heavily on local visual descriptors and thus fail when these cannot be extracted, e.g. due to little visual texture, changes in illumination or when they are not distinctive enough. Further recent work has extended the concept of local image descriptors to create 3D local or global descriptors [8] and [9] geared for object detection, but these suffer again from distinctiveness or contamination from changes in the environment or occlusion.

In this paper we present an approach to recognize places using range images from an RGB-D camera that overcomes the main drawbacks of local appearance based methods. Specifically, we propose a compact plane-based representation of the scene that we name PbMap (Plane-based Map). This PbMap is organized as an annotated graph where each node is a 3D planar patch (described by simple geometric features: plane's normal, centroid, area, etc.) and the edges connect neighbor patches according to their proximity. These planar patches (or planes, for short) are extracted in real-time from the range video streaming provided by a hand-held sensor. Such planes are integrated into the map in their respective poses, according to the sensor location, which is estimated from the RGB-D observations using a visual odometry algorithm. Although other sources of map creation such as full SLAM are also suitable to use. The use of odometry only for constructing our maps implies that our representation is topological in nature, but we note that place localisation does not require fully consistent maps to work.



a) b)

Fig. 1. a) Example of a typical scene that can be represented with planar patches. b) PbMap of the scene where a local neighborhood of planes is represented, including a reference plane (green), and its closest planes (blue) up to a distance threshold (1 m).

Place recognition in PbMaps is addressed as a problem of matching subgraphs: those subgraphs representing the observed planes are compared with other ones from the PbMap. Such subgraphs are defined by one reference plane together with their closest neighbors, up to a distance threshold (see figure 1). For solving the graph matching problem we rely on an interpretation tree [10] that exploits the geometric characteristics of the planes and their relative positions to generate a set of unary and binary constraints that guide efficiently the search. For gaining in robustness, we introduce a consistency test to finally accept the match given by the search process. This consistency test evaluates the rigid adjustment of the matched planes, providing the adjustment error together with the approximate position of the sensor with respect to the place recognized.

This work was supported by the Spanish Government under the research contract CICYT - DPI-2008-03527.

[1]Dpto. Ingeniería de Sistemas y Automática. Universidad de Málaga. E.T.S. Ingeniería de Informática-Telecomunicación. Campus de Teatinos, 29071, Málaga. E-mail: eduardofernandez@isa.uma.es

[2]Department of Computer Sciences. University of Bristol. Bristol Robotics Lab. Merchant Ventures Building. Wooland Road. Bristol BS8 1UB.

The proposed approach for recognizing a visited place has three main advantages:

1. the description of the scene through a PbMap is very compact, requiring little memory and reducing the computational cost of the required search;
2. it is robust to changes of viewpoint since the scene planes can be detected from very different poses;
3. it tolerates reasonably well changes in the scene, and therefore is adequate for the so called "lifelong maps", i.e. maps that are still valid after the scene changes. This characteristic particularly holds for indoor scenarios, where the most visible and larger planes (i.e. walls, floor, ceiling, bigger furniture, etc.) are normally persistent over time, while other smaller objects (e.g. chairs, a laptop, a backpack) are more likely to be moved or even disappear.

Our approach assumes that there are enough planar patches and thus busy indoor scenes are more amenable for our method.

There exist other methods in the literature which also address the place recognition problem from range data: the work of [11] presents a solution for place recognition which employs distinctive keypoints from 2D lidar observations. This approach is extended to 3D laser point clouds in [12]. The work in [13] also employs range data to extract features that capture important geometric and statistical properties to detect loop closures. Our approach differs from the above ones in different aspects: a) the search for a place is performed using contextual information of nearby planes, and so, it relies on the continuous perception of the scene instead of particular observations of the sensor, b) our method does not require a training step, and c) it describes the scene in a more continuous way with a plane-based representation which is useful beyond place recognition (e.g. scene modeling).

We provide experimental results demonstrating the effectiveness of our method for recognizing and localizing places in a dataset composed of 15 home and work-place scenes: offices, living rooms, kitchens, bathrooms, bedrooms and corridors. In order to test the concept of "lifelong map" we also show how the recognition is affected by the fact that the scene suffers some changes.

Next, we describe the PbMap construction procedure and show how this map is used to search for similar plane configurations in previous subgraphs. The experiments and their results are presented in section V. Finally, we expose the conclusions of our work.

II. PLANE-BASED MAP (PBMAP)

A plane-based map (PbMap) is a representation of the scene as a set of 3D planar patches. It is organized as an annotated, undirected graph *G,* where each node represents a planar patch and the edges connect neighbor planes, that is, an edge connects two patches when the distance between their closest points is under a threshold (see figure 2). Each plane $P_i \in G$ is described by a set of geometric features: the centroid, the area, the elongation, the normal vector, and the principal vector (i.e. dominant direction of the plane). A node also stores a set of points defining the patch's convex hull, which serves to calculate the minimum distance between two patches. All these features are obtained from the plane segmentation and map construction stages: they are set when a plane is initialized and are updated when such plane is re-observed.

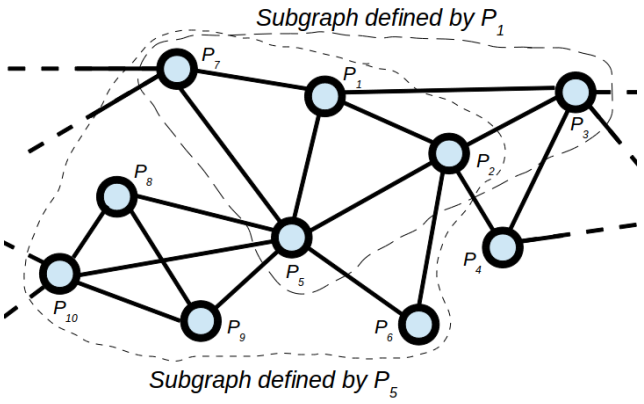


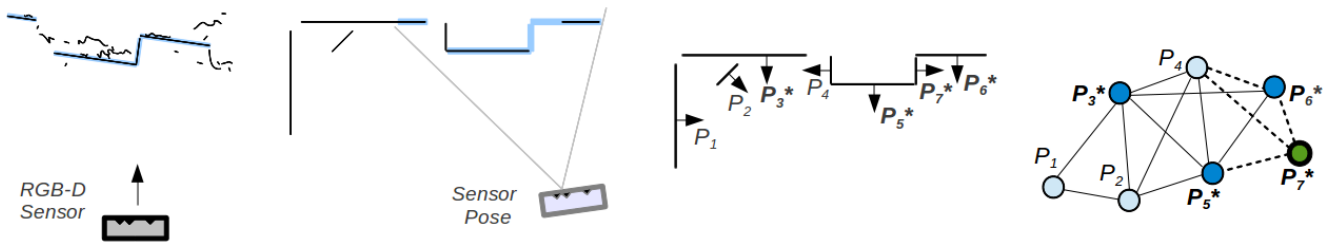
Fig. 2. Example of the graph representation of a PbMap, where the arcs indicate that two planes are close by. Two subgraphs are indicated: the ones generated by the reference planes P_1 and P_5 respectively.

A. Map generation

The planar patches are segmented from 160x120 range images in real time using a region growing technique [14]. This method exploits the spatial organization of the range images to estimate the normal vector for each 3D-point corresponding to each pixel, and then it clusters them to obtain the planar patches. This technique is computationally less expensive than other well-known methods such as RANSAC, which has been used before for concurrent plane extraction in SLAM [15], [16]. After this segmentation stage, a detected planar patch is integrated into the PbMap according to the sensor pose, either by updating an already existing plane or by initializing a new one when it is first observed.

The sensor pose needed to locate the planes in a common frame of reference can be computed in different ways, for example, using visual odometry, as it is the case implemented in this work. Concretely, the odometry method followed here is the one presented in [17]. This method estimates the relative pose between two consecutive RGB-D observations by iteratively maximizing the photoconsistency of both images. The optimization is carried out in a coarse-to-fine scheme that improves efficiency and allows coping with larger differences between poses. The drift of this algorithm along the trajectory is sufficiently small to achieve locally accurate PbMaps.

The PbMap construction procedure is illustrated in figure 3. For every new frame, a subsampled point cloud (160x120) is built relative to the sensor, and planar patches are segmented from it. The segmented patches are then placed



a) b) c) d)

Fig. 3. 2D representation of the map construction scheme. a) RGB-D capture with segmentated planes (blue). b) Current PbMap with segmented planes (blue) superimposed according to the sensor pose. c) PbMap updated: the planes updated are highlighted d) PbMap graph updated: the planes updated are highlighted in blue, the new plane P_7 is marked in green and, the new edges are represented with dashed lines.

in the PbMap according to the sensor pose (figure 3.b). If the new patch overlaps a previous one and their normal vectors coincide they are merged and the parameters of the resulting plane are updated. In other case, a new plane is initialized in the PbMap (figure 3.c). The graph connections of the observed planes are also updated calculating the minimum distance between them and their surrounding planes that are not connected yet (figure 3.d).

III. Place Recognition

The problem addressed here is that of matching local neighborhoods of planes, represented as subgraphs in the PbMap. Concretely, as the PbMap grows and is populated with new planes, the current observed planes are used to define subgraphs (one per observed plane) that are to be matched with other ones previously acquired. The subgraphs are composed of those planes 1-connected with the one considered as reference (e.g. the subgraphs generated by P_1 and P_5 in figure 2). The maximum number of subgraphs in the PbMap is limited by the number of planes, though in practice, this number is smaller, since one particular subgraph can be generated from two –or more– neighbor planes (e.g. the subgraphs generated by P_8 and P_9 in figure 2). Also, when a subgraph is contained in other subgraph, only the largest one is considered for matching a place.

In order to match two subgraphs, S_C, generated from the current range image, and S_M, generated from previous observations, we rely on an interpretation tree [10], which employs geometric restrictions represented as a set of unary and binary constraints. On the one hand, the unary constraints are used to check the correspondence of two single planes based on the comparison of their geometric features. On the other hand, the binary constraints serve to validate that two pairs of connected planes of S_C and S_M, respectively, present the same geometric relationship (e.g. the angle between normal of both pairs are similar up to a given threshold). All such constraints depend on thresholds that have been experimentally determined from different tests carried out in several scenarios. An important advantage of the interpretation tree is that it allows us to recognize places when the planes are partially observed or missing, allowing the system to deal with non-static scenes.

Algorithm 1 Employs an interpretation tree to search recursively for the best match between two subgraphs of planes S_C and S_M, for a given set of matched planes (initially empty).

```
INPUT:   S_C, L_C   // Current subgraph and List of planes of S_C
         S_M, L_M   // Previous subgraph and List of planes of S_M
         matched_planes   // List of matched planes
OUTPUT:  best_combination // Final list of matched planes

best_combination = MatchSubgraphs(L_C, L_M, matched_planes)

  best_combination = matched_planes

  for each plane P_C ∈ L_C do
    for each plane P_M ∈ L_M do

      if EvalUnaryConstraints(P_C, P_M) == F then
        continue
      end if

      for each {P_C', P_M'} ∈ matched_planes do

        // Check if the edges {P_C, P_C'} and {P_M, P_M'} exist
        if {P_C, P_C'} ∈ S_C and {P_M, P_M'} ∈ S_M then

          if EvalBinaryConstraints({P_C, P_C'}, {P_M, P_M'}) == F then
            continue
          end if
        end if
      end for

      // Remove P_C from L_C and P_M from L_M
      new_L_C = L_C - P_C
      new_L_M = L_M – P_M

      new_matched_planes = matched_planes ∪ {P_C, P_M}

      // Search for the best combination of matched planes
      result =MatchSubgraphs(new_L_C, new_L_M, new_matched_planes)

      // Check the length of the resulting list of matched planes
      if SizeOf(result) > SizeOf(best_combination) then
        best_combination = result
      end if
    end for
  end for

  return best_combination
```

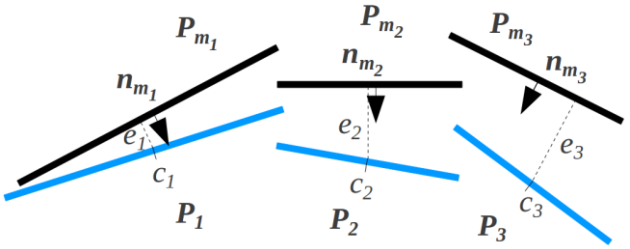


Fig. 4. Consistency test. 2D representation of depth error (the blue segments represent planes of the current subgraph and the black segments correspond to a previous subgrah.

Algorithm 1 describes the recursive function for matching two subgraphs. This function checks all the possible combinations, defined by the graph edges, between planes of the subgraphs S_C and S_B, to find the one with the maximum number of matches. In order to assign a new match between a plane from S_C and a plane from S_B the unary constraints are verified first, and if they are satisfied, the binary constraints are checked with the already matched planes. If all the constraints are satisfied, a match between the planes is accepted and the recursive function is called again with updated arguments. The algorithm finishes when all the possibilities have been explored, returning a list of pairs of corresponding planes.

Despite the large amount of possible combinations for this problem, most of them are rejected in an early stage of the exploration since they do not fulfill the geometric restrictions. In addition, the evaluation of these restrictions requires little computation, since they only do simple operations to compare 3D vectors and scalars. This allows the search process to work at frame rate.

Notice that this process can give rise to several candidate places, one per previous subgraph. From these candidates, we choose the one with the best rigid alignment, which is given by the consistency test described in the next section.

IV. Consistency Test

The consistency test evaluates the rigid correspondence of the matched planes from two subgraphs provided by the interpretation tree. For that, we need to estimate the relative pose in 6D, $\boldsymbol{\mu}$, between the matched places. This is accomplished by minimizing a cost function which measures the adjustment error of each matched plane. Mathematically

$$\hat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu}} \sum_{i=1}^{N} e_i(\boldsymbol{\mu})^2 \qquad (1)$$

where N is the number of matched planes and $e_i(\boldsymbol{\mu})$ represents the adjustment error of the planes $\mathbf{P}_i$ and $\mathbf{P}_{m_i}$ with respect to the rigid transformation defined by $\boldsymbol{\mu}$. This error corresponds to the distance between the centroid of $\mathbf{P}_i$ and the plane $\mathbf{P}_{m_i}$ (refer to figure 3). More precisely, the proposed error function $e_i(\boldsymbol{\mu})$ is given by

$$e_i(\boldsymbol{\mu}) = w_i \, \mathbf{n}_{m_i} \left(\exp(\boldsymbol{\mu}) \mathbf{c}_i - \mathbf{c}_{m_i} \right) \qquad (2)$$

being $\mathbf{n}_{m_i}$ and $\mathbf{c}_{m_i}$ the normal and the centroid of $\mathbf{P}_{m_i}$; $\mathbf{c}_i$ the centroid of $\mathbf{P}_i$, and $\exp(\boldsymbol{\mu})$ the rigid transformation matrix $SE(3)$ represented as the exponential map of the 6D vector $\boldsymbol{\mu}$, which is a minimal parameterization, and w_i a weight defined by

$$w_i = \frac{A_i}{\sum_{j=1}^{N} A_j} \qquad (3)$$

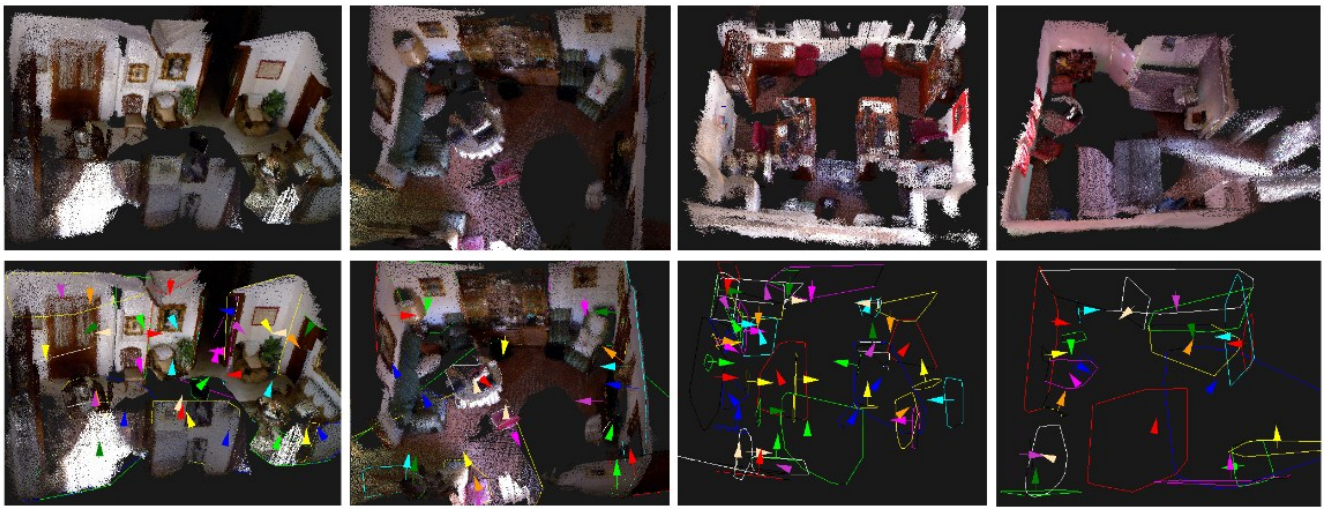
where A_i and A_j are the area of the planes $\mathbf{P}_i$ and $\mathbf{P}_j$ respectively. This weight gives more relevance to the adjustment error of larger planes over smaller ones. We solve this least squares problem using Gauss-Newton optimization for $\boldsymbol{\mu}$. After the relative pose has been calculated, the resulting error is used to evaluate the consistency of the candidate matches.

This consistency test is evaluated for those matched subgraphs given by the place recognition stage, selecting the one that presents the minimum error. And finally, the match is accepted if this error is smaller than a given threshold (0.04 m^2 in our experiments).

V. Experiments

In this section we present the experiments carried out to validate our approach in two different ways: first, we test the effectiveness for recognizing places in 300 tests performed in an environment composed of 15 rooms; second, we evaluate the robustness of our solution to recognize places in non-static scenes, in other words, we evaluate the suitability of the PbMaps to represent scenes that suffers changes continuously (lifelong maps). In these experiments we have employed an Intel Core i7 laptop with 2.2 GHz processor. Our RGB-D camera is a MS Kinect sensor. The visual odometry works independently in one thread while the PbMap is built and explored for previous places in a second thread, this process works at frame rate (30 Hz), where the main load comes from the plane segmentation, which takes around 13 ms per frame.

In the first battery of experiments we explore the scene with a handheld RGB-D sensor, building progressively a PbMap and searching continuously the current place in a set of 15 previously acquired PbMaps corresponding to different rooms (these PbMaps generally capture a 360º coverage of the scene, see figure 5). An additional challenge of this experiment comes from the fact that some PbMaps represent the same type of room. We have repeated 20 exploration sequences with different trajectories for each one of the 15 different scenarios, recording the success and failure rates, together with the average length of the sensor trajectory until a place was detected, or until the scene was fully observed when no place is recognized. Table I shows the recognition rate for these experiments. The first column indicates the percentage of cases where a place was recognized correctly,



a) b) c) d)

Fig. 5. Different scenarios where place recognition has been tested. These pictures correspond to some of the maps created previously in the place recognition tests. The top row correspond to point clouds registered with visual odometry, while the lower row correspond to the PbMaps (the point cloud is also superimposed in a) and b)). These scenarios correspond to: a) Living-room 1, b) Living-room 2, c) Office2 and d) Bedroom 1.

Scenario	Recog Rate	Failure Rate	Av. Path Length (m)
LivingRoom1	100%	0%	5.53
LivingRoom2	100%	0%	3.25
LivingRoom3	100%	0%	2.85
Kitchen1	100%	0%	4.53
Kitchen2	100%	0%	2.24
Kitchen3	90%	0%	3.75
Office1	100%	0%	2.01
Office2	90%	10%	2.61
Office3	90%	10%	3.82
Hall1	100%	0%	1.34
Hall2	80%	10%	2.31
Bedroom1	60%	10%	4.98
Bedroom2	50%	20%	6.25
Bedroom3	55%	20%	5.52
Bathroom	50%	35%	5.60

Table I. Effectiveness of the proposed method in different environments with different exploration trajectories (20 tests for each environment). There are some tests where no place was recognized (neither correctly nor erroneously), as a consequence, the sum of the recognition rate and the failure rate is not 100%.

while the failure rate stands for the percentage of places recognized erroneously. The average length of the path taken until a place is recognized is shown in the third column. This somehow gives an idea of how distinctive neighborhoods of planes are for each different scenario. However note that the length of exploration is not directly related to the recognition rate, since even on scenes with few distinctive subgraphs (e.g. the case of an empty room) can eventually lead to a detection.

An interesting feature of our approach is that it can recognize easily places where there is little appearance information but the geometric configuration of planes is highly descriptive, this can be perceived in our videos http://mapir.isa.uma.es/efernandez/place_recognition. In cases where there are fewer extracted planar patches the recognition rate drops.

A second battery of experiments shows that our PbMap can be used to recognize places that have suffered some changes, but where the main structure of the scene is unchanged. For that, we have evaluated the recognition rate with respect to the amount of change in the scene, which is measured using ICP on the point clouds of the scene. Similarly as in the previous experiments, we evaluate the recognition rate for 20 different trajectories exploring each one of two following scenarios: Office1 and LivingRoom1 (we have chosen these two scenarios because changes are more common in them, see figure 6). The results of these experiments, summarized in Table II, show that the recognition rate remains high for moderate changes in the scene (Ch1 & Ch2, where chairs have been moved, and some objects, like the laptop, have disappeared from the scene, while new objects have appeared). Though as expected, this rate decreases as the change in the scene increases significantly (Ch3 & Ch4, where cardboard boxes have been placed in the scene, occluding previous planes and generating new ones).

a) b) c)

Fig. 6. Lifelong maps in office environment. a) Reference scene, b) Scene with moderate changes (Ch3) c) Scene with significant changes (Ch5).

Office1	Ch0	Ch1	Ch2	Ch3	Ch4
Av. ICP error (mm)	0	0.671	1.215	1.540	3.442
Recognition	100%	100%	95%	90%	80%

LivingRoom1	Ch0	Ch1	Ch2	Ch3	Ch4
Av. ICP error (mm)	0	1.182	2.010	2.942	3.863
Recognition	100%	100%	100%	95%	85%

Table II. Lifelong maps. The ICP fitness score shows the average adjustment error per 3D-point. The recognition shows the percentage of "finds" for 20 different trajectories exploring the scene.

VI. Conclusion

Plane extraction has been used in previous mapping and modelling systems but it has rarely been used to quickly describe a scene in a manner useful for real-time place detection. This article presents a real-time place recognition method for indoor environments using range images from an RGB-D camera. Our approach relies on a plane-based representation of the scene (PbMap). This representation has the advantage of being very compact, and so, permits fast map exploration to detect previously visited places. We introduce an interpretation tree to perform the place search efficiently, together with a consistency test to verify the rigid adjustment of the matched places, which also provides localization. Experiments have demonstrated the potential of our approach to recognize places efficiently, working even for non-static scenes.

In order to study the potential of a purely geometric approach we have not made use of RGB information here to recognize places. Color information, however, could be exploited in several ways to produce a more robust and efficient solution. Our formulation allows this extension easily, e.g. by introducing new constraints in the interpretation tree. This will be one line of future work, in which we aim to evaluate the increase of performance of a combined strategy.

Acknowledgment

We are very grateful to our colleagues Miguel Algaba for providing an implementation for visual odometry and Jose Martínez-Carranza for useful discussions about this work.

References

[1] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos", in *Proceedings of the Intl. Conf. on Computer Vision*, Nice, France, 2003.

[2] I. Ulrich and I. Nourbakhsh, "Appearance-Based Place Recognition for Topological Localization" , In *IEEE International Conference on Robotics and Automation*, 2000.

[3] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. "Context-based vision system for place and object recognition". In Proc. ICCV, 2003.

[4] A. Oliva and A. Torralba. "Building the gist of a scene: The role of global image features in recognition". In *Visual Perception, Progress in Brain Research*, volume 155. Elsevier, 2006.

[5] M. Cummins, and P. Newman, "FAB-MAP: probabilistic localization and mapping in the space of appearance". *The International Journal of Robotics Research,* 27: 647-665, 2008.

[6] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. "Fast and incremental method for loop-closure detection using bags of visual words". IEEE *Transactions on Robotics (T-RO),* 24(5):1027–1037, 2008.

[7] S. Lazebnik, C. Schmid, and J. Ponce. "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories". In *CVPR*, 2006.

[8] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[9] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram," in Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 2010.

[10] Grimson, W. E. L. Object Recognition by Computer - The role of Geometric Constraints. MIT Press, Cambridge, MA. 1990.

[11] M. Bosse and R. Zlot. "Map matching and data association for large-scale 2D laser scan-based SLAM". *International Journal of Robotics Research,* 27(6), June 2008.

[12] M. Bosse and R. Zlot. "Place recognition using regional point descriptors for 3D mapping". In *IEEE Proceedings of the Intl. Conf. on Field and Service Robotics (FSR).* Cambridge, MA. 2010.

[13] K. Granström, T.B. Schön, J.I. Nieto and F.T. Ramos. "Learning to close loops from range data". In *The International Journal of Robotics Research,* 30 (14): 1728-1754. 2011.

[14] D. Holz and S. Behnke, "Fast Range Image Segmentation and Smoothing using Approximate Surface Reconstruction and Region Growing," in Proceedings of the International *Conference on Intelligent Autonomous Systems (IAS),* Jeju Island, Korea, 2012.

[15] Andrew P. Gee, Denis Chekhlov, Andrew Calway, Walterio Mayol-Cuevas, "Discovering Higher Level Structure in Visual SLAM". *IEEE Transactions on Robotics*, 24(5). ISSN 1552-3098, pp. 980–990. October 2008.

[16] J. Martínez-Carranza and A. Calway. "Unifying planar and point mapping in monocular SLAM". In *British Machine Vision Conference (BMVC)*, 2010.

[17] F. Steinbruecker, J. Sturm and D. Cremers. "Real-Time Visual Odometry from Dense RGB-D Images". In *Workshop on Live Dense Reconstruction with Moving Cameras at the Intl. Conf. on Computer Vision (ICCV)*, 2011.