Scene structure registration for localization and mapping

Eduardo Fernández-Moral, Patrick Rives, Vicente Arévalo, Javier González-Jiménez

Scene structure registration for localization and mapping

Eduardo Fernández-Moral[a,*], Patrick Rives[a], Vicente Arévalo[b], Javier González-Jiménez[b]

[a] *INRIA Sophia Antipolis - Méditerranée, Lagadic team. 2004 route des Lucioles - BP 93, 06902 Sophia Antipolis, France.*
[b] *Universidad de Malaga, Mapir Group. E.T.S. Ingeniería de Informática-Telecomunicación. Campus de Teatinos, 29071, Málaga, Spain.*

Abstract

Image registration, and more generally scene registration, needs to be solved in mobile robotics for a number of tasks including localization, mapping, object recognition, visual odometry and loop-closure. This paper presents a flexible strategy to register scenes based on its planar structure, which can be used with different sensors that acquire 3D data like LIDAR, time-of-flight cameras, RGB-D sensors and stereo vision. The proposed strategy is based on the segmentation of the planar surfaces from the scene, and its representation using a graph which stores the geometric relationships between neighbouring planar patches. Uncertainty information from the planar patches is exploited in a hierarchical fashion to improve both the robustness and the efficiency of registration. Quick registration is achieved in indoor structured scenarios, offering advantages like a compact representation, and flexibility to adapt to different environments and sensors. Our method is validated with different sensors: a hand-held RGB-D camera and an omni-directional RGB-D sensor; and for different applications: from visual-range odometry to loop closure and SLAM.

Keywords: Scene registration, scene recognition, localization, mapping, planar segmentation

1. Introduction

Image registration has been a major problem in computer vision over the past decades. It implies searching an image in a database of previously acquired images to find one (or several) that fulfil some degree of similarity, e.g. an image of the same scene from a similar viewpoint. This problem is interesting in mobile robotics for topological mapping, re-localization, loop closure and object identification. Scene registration can be seen as a generalization of the above problem where the representation to match is not necessarily defined by a single image (i.e. the information may come from different images and/or sensors), attempting to exploit all information available to pursue higher performance and flexibility.

This paper addresses the problem of scene registration from 3D data using a compact description which encodes geometric information (and photometric information if it is available) about the scene's planar surfaces. This article extends a previous work which relies on the segmentation of planar patches to build a Plane-based Map (named PbMap, see figure 1) [1]. Such solution is extended here using a probabilistic framework to take into account the uncertainty model of the sensor. This approach is specially interesting for depth devices like range cameras or stereo vision, which deliver data in an organized way (i.e. neighbouring pixels correspond to close-by 3D points) so that it allows us to segment efficiently planar patches, referred as planes for short.

The key idea in this article is that even a small set of neighbouring planar patches (e.g. 4-10) encode enough information to recognize and register a scene. This strategy contrasts with previous methods that make use of local or global descriptors [2]. A relevant difference is that our method exploits a connected description of the scene and thus, it is less dependent on the particular field of view of the sensors, offering a piecewise continuous description that supports multi-sensor and multi-frame ob-

*Corresponding author: eduardo.fernandez-moral@inria.fr

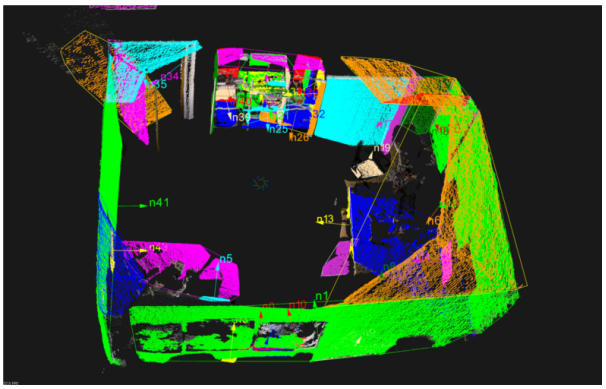

Figure 1: Colored planar patches of the PbMap representation of an office environment.

servations. We present a set of experiments using RGB-D cameras (Asus Xtion Pro Live -XPL-), both with a single sensor waved by hand, and a rig of sensors which provide omnidirectional RGB-D images mounted on a wheeled robot [3]. These experiments demonstrate the potential of our technique for localization, loop closure, odometry and SLAM. Our registration technique has been implemented in C++ and is freely available as a module of PCL.

1.1. Related works

Most solutions for scene registration make use of intensity information from regular cameras. Such sensors have the advantage to provide detailed information at a low cost. Scene registration from images generally requires invariance to changes of illumination and viewpoint, and robustness to visual aliasing and lack of texture among others [4]. The methods proposed in the literature are traditionally classified into feature-based or dense, depending on whether they are based on local features or on global appearance. Local features like point descriptors (e.g. SIFT, SURF, ORB, etc.) are commonly applied to camera tracking (visual odometry). On the other hand, dense methods minimize the photometric difference between two images with respect to a warping function which is estimated iteratively, not requiring any previous step to extract and match salient points. However, dense methods require an initial estimate of the image warping, plus they need to make assumptions about the scene structure when 3D information is not available (monocular).

Scene recognition is a problem related to scene registration, where the former does not require to find the alignment between the scenes. Thus, scene recognition is related to topological localization while scene registration is related to metric localisation. Image indexation is a particular case of scene recognition for which the data to retrieve are images [2]. In robotics, the most common approach to solve this problem is the bag-of-words (BoW) method [5, 6], which creates a dictionary of features and employs a voting method to retrieve the location, is traditionally used for re-localisation, loop closure or topological localisation and mapping. The authors of [7] presented an alternative approach by creating a compact global descriptor which is related to the shape of the scene and the geometric relationships in it. This last is generally faster, but less suited if the relative pose between the images have to be recovered.

The use of 3D data to solve the scene registration problem is increasingly popular in mobile robotics due to several affordable range sensors that have appeared in the last years, like these of PrimeSense. Before that, registration of 3D shapes has been researched mainly for object recognition [8], where a limited amount of data, usually a point cloud of a well delimited object, is registered using Iterative-Closest-Point (ICP). This kind of solution cannot be directly applied for re-localization since first, it requires an initial estimation of the relative pose between the point clouds; and second, the point cloud description of the scene cannot be easily delimited to a set of corresponding points within a large reference map, being difficult to find a reference and a target point cloud with almost same content. Another approach which makes use of depth information (and also photometry) is direct registration [9, 10], but it has the same limitations as ICP since it requires an initial estimation for the registration, and such estimation is not available in re-localisation kind of problems.

The technique we propose here is an extension of our previous work in [1], which can be seen as a combined geometric-photometric global descriptor where the scene is described by a set of neighbouring planar patches which capture information about the scene's global set-up. A recent work which is based on matching a set of planar patches and line segments is described in [11], however it is restricted to match images, in a similar fashion to the SLAM solution of [12], and does not exploit the fact that

planar patches can be used to build a piecewise continuous representation using a graph. In contrast to previous registration approaches, our method is not limited to image registration, and therefore it can be applied to different sensors and can use information of several images or video sequences. This advantage is crucial in situations where a single image does not provide enough information to register the scene, but whose information is useful to complete previous observations (a situation which is quite common in mobile robotics). In this sense, our work is more related to visual place recognition approaches which exploit the information of a sequence of images to boost the recognition accuracy [13, 14]. The performance of such solutions is highly related to the size of the sequence describing the scene to match. This concept is similar to the size of the set of neighbour planes describing the scene. However, this last has the advantage that it is directly related to the size of the scene, and it abstracts from other aspects like sensor field of view, or proximity of the frames in the sequence. Thus, our solution frees the user from tuning sensor and robot parameters, requiring only to specify the size of the scene to be matched through the number of planes.

1.2. Contribution

The main contributions of this paper are:

- We extend our previous Plane-based Map description and registration technique to incorporate uncertainty information in a hierarchical structure to improve robustness and efficiency.
- We generalize this solution so that it can take different sources of data, from different kinds of range cameras to stereo vision. A validation is presented with different sensors and for different applications.
- An implementation is made available as a module of PCL [15], including a tutorial with some practical examples.

2. PbMap: a representation of the scene's planar structure

A plane-based map (PbMap) is a representation of the scene as a set of 3D planar patches, which are described by a series of geometric and radiometric attributes [16]. It can be built from different sources of data (several range images from different sensors) if we know the relative poses of the different observations. In this case, the overlapping patches are merged together so that each planar surface in the scene is represented by a single patch.

2.1. Planar patch segmentation and parametrization

In order to obtain the planar patches the depth images are segmented with a region growing approach which is implemented in the Point Cloud Library (PCL) [17]. This method exploits the spatial organization of the range images to efficiently estimate the normal vector for the 3D-point corresponding to each pixel, and then it clusters them with region growing to obtain the planar patches. This technique is computationally less expensive than other well-known methods such as RANSAC.

A planar patch is represented by its normal vector $\mathbf{n}$, with $\|\mathbf{n}\| = 1$, and the distance d from the infinite plane to the optical center of the camera. In this way, a 3D point $\mathbf{p} = \rho \cdot \mathbf{m}$ given by the range ρ and the ray direction $\mathbf{m}$ (bearing direction), which lies on the plane fulfils the equation

$$\mathbf{n} \cdot \mathbf{p} + d = 0 \tag{1}$$

The plane parameters and their covariances are estimated following [18], assuming that the bearing directions are accurate, so that the noise only affects the range measurements ρ_i. We assume that $\rho_i \sim N(\hat{\rho}_i, \sigma_i)$, where $\hat{\rho}_i = d/(\mathbf{n} \cdot \mathbf{m}_i)$ is the true range of the i-th measurement. The standard deviation σ_i is generally a function that depends quadratically on the range $\sigma_i = k\rho_i^2$ [19]. This value (or rather a conservative upper limit) may be provided by the sensor constructor, or can be inferred after an statistical analysis of the sensor noise. Then, the covariance of the i-th point is defined by $\mathrm{C}_{p_i} = \sigma_i^2 \cdot \mathbf{m}_i \cdot \mathbf{m}_i^T$. And from this, we can define a weight for each point

$$w_{p_i} = \frac{1}{trace(\mathrm{C}_{p_i})} = \frac{1}{\sigma_i^2} \tag{2}$$

and the weighted mass center of the planar patch

$$\mathbf{c} = \frac{\sum_{i=1}^{N} w_{p_i} \cdot \rho_i \cdot \mathbf{m}_i}{\sum_{i=1}^{N} w_{p_i}} \tag{3}$$

The optimal $\mathbf{n}$ is the eigenvector corresponding to the smallest eigenvalue of the matrix

$$\mathrm{M} = \sum_{i=1}^{N} w_{p_i} \cdot (\rho_i \cdot \mathbf{m}_i - \mathbf{c})(\rho_i \cdot \mathbf{m}_i - \mathbf{c})^T \tag{4}$$

and the optimal d is given by

$$d = \mathbf{n} \cdot \mathbf{c} \tag{5}$$

The covariance of the plane parameters Σ is calculated as inverse of the Moore-Penrose generalized inverse of H

$$\Sigma = (-\mathrm{H})^{+} \tag{6}$$

$$\mathrm{H} = \begin{bmatrix} \mathrm{H}_{\mathbf{nn}} & \mathrm{H}_{\mathbf{n}d} \\ \mathrm{H}_{\mathbf{n}d}^T & \mathrm{H}_{dd} \end{bmatrix} \tag{7}$$

where

$$\mathrm{H}_{dd} = \sum_{i=1}^{N} w_{p_i}, \quad \mathrm{H}_{\mathbf{n}d} = \mathrm{H}_{dd}\mathbf{c} \tag{8}$$

$$\mathrm{H}_{\mathbf{nn}} = \mathrm{M} - \mathrm{H}_{dd}\ \mathbf{c}\ \mathbf{c}^T - (\mathbf{n}^T\ \mathrm{M}\ \mathbf{n})\mathrm{I}_3 \tag{9}$$

2.2. Planar patch uncertainty

The covariance of the plane parameters defined above provides information about the patch's uncertainty. Such information is useful to register two sets of planar patches in order to select the most informative patches to match. This differs from our previous work [1] where the number of points supporting each plane was used instead as a measure of information. By using the uncertainty provided by the covariance we take into account the model of uncertainty of the sensor and thus, the different noise level of the planes, which is generally higher for planes further away and when they are observed with narrower incident angles. Thus, a matched plane with higher uncertainty will contribute less to the scene's matching score.

Another relevant point is to match planes that are in a variety of orientations with respect to a given reference system (either the reference or the target scene). This is necessary for recovering the relative pose between the matched scenes which requires to match at least three planes with linearly independent normal vectors, and also to avoid wrong matches due to geometric aliasing, e.g. as it can happen in a staircase. A nice strategy to balance the two factors above is presented in [11] with the *information content factor*. This factor was defined according to both the uncertainty and the distribution of normal vectors of the set of planar patches, resulting in the following value of scene's planar information

$$Y = \sum_{i=1}^{N} \mathbf{n}_i \cdot \mathbf{n}_i^T \cdot \mu_i \tag{10}$$

where $\mathbf{n}_i$ is the normal vector of the $i-$th plane and μ_i is the number of 3D points supporting the plane (inliers) representing the inverse of the plane's uncertainty. Then, the contribution of a given plane to the total information can be computed as $\mathbf{n}_i^T \cdot Y \cdot \mathbf{n}_i$, and then, the weight w_i assigned to the $i-$th plane to measure its contribution in the direction of its normal vector is defined as

$$w_i = \frac{\mu_i}{\mathbf{n}_i^T \cdot Y \cdot \mathbf{n}_i} \tag{11}$$

In this paper we employ a variation of the *information content factor* which takes into account the sensor's uncertainty model. The idea is to introduce the computed plane covariance (eq. 6) into the above factor instead of naively using the number of supporting points as a measure of uncertainty. Thus, in this work the value of μ_i is given by the second smallest eigenvalue of the information matrix H. The second smallest eigenvalue is chosen because H is rank deficient ($rank(\mathrm{H}) = 3 < 4$) since the normal vector is overparametrized and thus, the smallest eigenvalue should be zero. By choosing the second smallest eigenvalue we adopt a conservative measure of the plane's total uncertainty.

2.3. Formal definition of a PbMap

A PbMap is organized as an annotated, undirected graph G, where each node represents a planar patch and the edges connect neighbour patches (see figure 2). Such neighbourhood depends upon both distance and visibility conditions. Thus, an edge will connect two patches when:

- the distance between their closest points is under a threshold, and
- the patches are co-visible in at least one observation.

Besides the geometric parameters defined in the plane segmentation section, the information of the planar patches is complemented with a series of geometric and radiometric attributes which are used later for registration. Each plane $\mathrm{P}_i \in \mathrm{G}$ is described by:

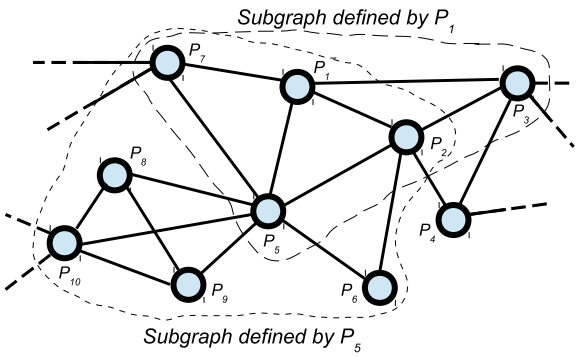

Figure 2: Example of the graph representation of a PbMap, where the arcs indicate that two planes are neighbour. Two subgraphs are indicated: the ones generated by the reference planes P_1 and P_5, respectively.

- $\mathbf{n}_i$ the normal vector,
- $\mathbf{c}_i$ the centroid,
- d_i the distance to the reference frame,
- Σ_i the covariance matrix of $\{\mathbf{n}_i, d_i\}$,
- L_i a list of points defining the patch's convex hull,
- $\mathbf{v}_i$ the principal vector,
- e_i the elongation,
- a_i the area, and
- $\mathbf{r}_i$ the dominant colour.

$\mathbf{n}_i$, $\mathbf{c}_i$, d_i, and Σ_i are computed as detailed in section 2.1. The convex hull L_i is efficiently computed from the patch's contour points (which are provided by the region growing segmentation) in 2D by forcing the contour points to fulfil the plane equation (eq. 1). The rest of geometric attributes are computed from the convex hull. The principal vector $\mathbf{v}_i$ is the eigenvector corresponding to the largest eigenvalue of the covariance matrix M' (M' is a matrix similar to M (eq. 4), but it is computed from the convex hull's weighted vertices to allow efficient update of merged patches). The elongation e_i is computed as the ratio between the two largest eigenvalues of M', and a_i is the area of the convex hull.

When radiometric information is available it is used through the patch's dominant colour $\mathbf{r}_i$, which is computed following [20]. This feature encodes the main colour of the patch extracted using a mean shift algorithm in normalized rgb. This feature has demonstrated to have a similar performance as the hue histogram [21] in typical indoor environments. The information about the consistency of the dominant colour is also stored, so that patches which have several predominant colours are marked as not reliable and thus, this property is not used for matching planes.

Note that the uncertainty of a planar patch is only tied to the parameters of the plane equation. The reason is that these are the only parameters which are invariant to the viewing conditions. The rest of parameters defining a patch are used for pruning the search space as it is explained in section 4, but they are not used to evaluate the quality of the match since our scene registration approach is designed to work even from very different viewpoints.

3. PbMap construction

After segmentation each 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 obtained in different ways. The most desirable solution in mobile robotics is to use extrinsic calibration to find the relative poses between the different sensors, and then to use all the sensors available to find the most likely movement of the robot. When the only information available comes from our range or RGB-D sensors, which is the case shown in our experiments, the current pose may be obtained from PbMap registration if a sufficient number of planes is observed, as it is done with omnidirectional RGB-D images (section 5.2). If this situation cannot be guaranteed, as it happens for a hand-held RGB-D camera, then visual-range odometry may be used [22, 23], as in the experiments in section 5.1.

The PbMap construction procedure is illustrated in figure 3. First, the planar patches are segmented from the sensor observation (figure 3.a), the segmented patches are then placed in the PbMap according to the sensor pose (figure 3.b). If the new patch overlaps a previous one, that is, they have the same plane parameters $\nu = \{\mathbf{n}, d\}$ up to

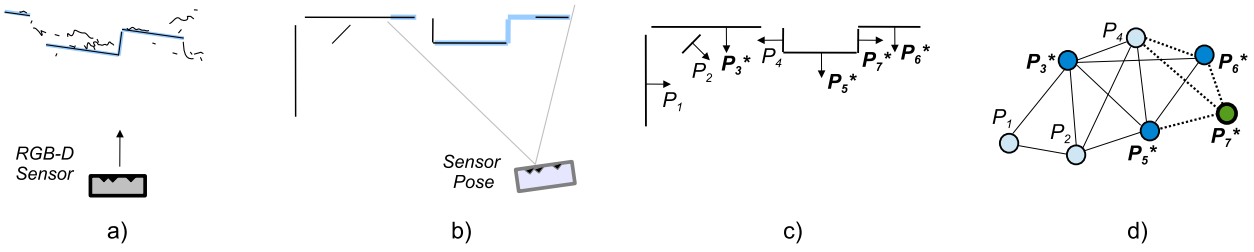

Figure 3: 2D representation of the map construction scheme. a) RGB-D capture with segmentated planes (blue). b) Current PbMap with segmented planes (blue) superimposed according to the sensor pose. c) PbMap updated: the planes updated are highlighted d) PbMap graph updated: the planes updated are highlighted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

a given threshold, and their convex hulls intersect, then 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 at every new observation by calculating the minimum distance between the current planes in view and the surrounding planes (figure 3.d). An example of a PbMap built from a short RGB-D video sequence in a home environment is shown in figure 4 where we can distinguish the different planes segmented.

In order to merge two patches P_a and P_b which correspond to the same surface, their covariances are taken into account to extract the fused patch P_f, resulting in

$$\Sigma_f^{-1} = \Sigma_a^{-1} + \Sigma_b^{-1} \tag{12}$$

$$\{\mathbf{n}, d\}_f = \Sigma_f(\Sigma_a^{-1}\nu_a + \Sigma_b^{-1}\nu_b) \tag{13}$$

Finally, the resulting normal vector must be normalized. Notice that the plane observations will generally come from different viewing poses. The plane parameters and their covariances must be previously placed in the same coordinate system, the mathematical details for this transformation can be found in [18]. The convex hull is updated by placing the two convex hulls in the same reference system, forcing the points to lie on the plane using the merged plane parameters, and recomputing the convex hull of the merged points. The rest of geometric parameters are derived from the convex hull. The resulting dominant colour is recomputed from weighted average of the two dominant colours, with the weights given the patch' areas. In case that a reliable dominant colour could not be extracted for at least one of the patches, then the resulting dominant colour is also marked as not reliable (see [20]).

4. Registration approach

The identification of a place using PbMaps is based on registering a set of neighbour planes that are represented by a graph. This process requires addressing three main issues: what and where to search, how to perform data association, and how to verify such data association. These questions are tackled below separately: first, the scope and the size of the subgraphs that are to be compared is chosen by selecting groups of k-connected patches; second, an interpretation tree is build upon geometric and radiometric constraints to find the association of planes with the highest score between the two subgraphs; and finally, the matched planes are aligned rigidly, providing an error measurement and the relative rigid transformation between the matched places. Notice that for the case of image registration, the issue of selecting a search scope is trivial since the reference an target scenes may be selected by considering all the planes in each image.

4.1. Search scope

The first issue in PbMap registration implies that we have to select a set of planes (or subgraph) which defines a place as a distinctive entity. The key to select a subgraph from the multiple combinations that are possible in a PbMap lies in the graph connections, as they link highly related planes in terms of distance and co-visibility. Thus, a subgraph is selected by choosing one or several reference planes and its neighbours up to level k in the graph, being k the graph distance (number of connections of the shortest path). The reference planes of the current view are those which are visible in the current frame. On the other hand, different subgraphs can be proposed to match

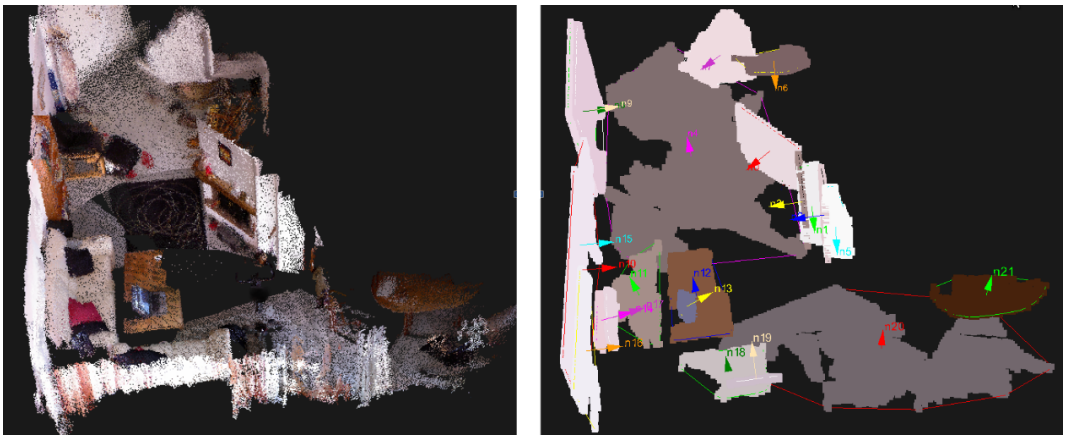

Figure 4: Plane based representation of a living room. The coloured planes at the right have been extracted from the point cloud at the left.

by making as many subgraphs as planes in the map, so that a candidate graph for re-localisation is composed of a reference plane and its neighbours up to level k. This strategy permits to describe a place in a piecewise continuous fashion, so that different subgraphs can be possible around a local area, providing flexibility to recognize places that are partially observed.

Notice that the sizes of the subgraphs defined by the number k affect both the descriptiveness and the computational cost of the graph search, which grows exponentially with the graph size. We note in our experiments that graphs composed of around 10 planes represent a good balance of descriptiveness *vs.* search cost. Besides the computational cost, another reason to work with small graphs is that large ones will generally be composed of several observations, which may accumulate drift and therefore reduce the accuracy of the representation.

The number of possible subgraphs in a PbMap (that may be selected for loop closure) grows linearly with the map size, that is, the maximum number of subgraphs in the PbMap is limited by the number of planes. Thus, in order to achieve a scalable solution for place recognition or loop closure we just need to guarantee that the subgraphs to be matched have a limited size, which is an intrinsic feature of our proposal.

4.2. Graph matching

The problem addressed here is that of matching local neighbourhoods of planes, represented as subgraphs in the PbMap. Thus, we aim to solve a graph matching problem allowing for inexact matching to be robust to occlusions and viewpoint changes. Several alternatives are found in the literature for this problem, from tree search to continuous optimization or spectral methods [24]. Here, we employ a tree search strategy because it is easy to implement and it is extremely fast to apply when the subgraphs to be compared have a limited size. We rely on an interpretation tree [25], which employs weak 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 and radiometric features. On the other hand, the binary constraints serve to validate that two pairs of connected planes present the same geometric relationship. An important advantage of this strategy is that it allows us to recognize places when the planes are partially observed or missing (inexact matching), resulting in high robustness to changes of viewpoint.

4.2.1. Unary constraints

The unary constraints presented here are designed to reject incorrect matches of two planes, and thus, to prune the branches of the interpretation tree to speed-up the search process. These are *weak* constraints, meaning that

the uncertainty about the plane parameters is high, so the thresholds are very relaxed to avoid rejecting a correct match. In other words, a unary constraint should validate that two planes are distinct when their geometric or radiometric characteristics are too different, but they lack information to confirm that two observations belong to the same plane, since even different planar patches can have the same characteristics.

Three unary constraints have been used here, which perform direct comparisons of the plane's area, elongation, and dominant colour if available. For example, the area constraint checks that the ratio between the areas of two observed planes are under certain bounds

$$\frac{1}{th_{area}} < \frac{a_{P_i}}{a_{P_j}} < th_{area} \tag{14}$$

and similar constraints are applied for the elongation and for the dominant colour.

In order to determine appropriate thresholds for such constraints, we analyse their performance in a dataset containing 1000 observations of plane surfaces from different scenarios, spanning diverse viewing conditions (changing viewpoint and illumination, partial occlusion, etc.). We have manually classified these planes, so that the correspondences of all plane observations are known. Then, we analyse the classification results of our constraints in terms of the sensitivity (ratio of actual positives which are correctly identified) and the specificity (ratio of negatives which are correctly rejected), for a set of different thresholds. The result of this experiment are shown with a ROC curve, which shows the sensitivity with respect to the specificity for a set of thresholds, see figure 5. As expected, the curves show that higher values of specificity correspond to smaller values of sensitivity and viceversa. Note that the nearer the curve is to the optimum point (1,1) the better the classification of the weak constraint. From this graph we can see that the colour is the most discriminative constraint, followed by the area and elongation constraints respectively.

The thresholds for each constraint are determined consistently by choosing a minimum sensitivity of 99%. We notice that those planes that are incorrectly rejected by a unary constraint correspond to planes which have been partially observed. The fact that some planes might be rejected incorrectly is not critical to recognize a place since not all of the planes are required to be matched. The

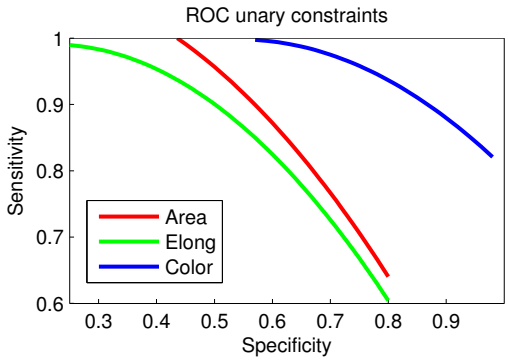

Figure 5: Comparison of the different unary constraints by their ROC curves (sensitivity *vs.* specificity).

thresholds obtained here depend on the amount and variety of the training samples used. But since most indoor scenes have planes of similar sizes and with similar configurations, such thresholds must be valid for most home and office environments. Besides, we have observed that big variations in the thresholds do not affect too much the results for place recognition.

4.2.2. Binary constraints

The binary constraints impose geometric restrictions about the relative position of two pairs of neighbour planes (e.g. the angle between the normal vectors of both pairs must be similar, up to a given threshold, to match the planes). These constraints are responsible to provide robustness in our graph matching technique by enforcing the consistency of the matched scene. Three binary constraints are imposed to each pair of planes in a matched subgraph. First, the angle difference between the two pairs being compared should be similar. This is

$$\left|\arccos(\mathbf{n}_i^r \cdot \mathbf{n}_j^r) - \arccos(\mathbf{n}_{ii}^t \cdot \mathbf{n}_{jj}^t)\right| < th_{angle} \tag{15}$$

where $\mathbf{n}_i^r$ and $\mathbf{n}_j^r$ are the normal vectors of a pair of nearby planes from the subgraph S_r, and similarly $\mathbf{n}_{ii}^t$ and $\mathbf{n}_{jj}^t$ are the normal vectors of a pair of planes from the subgraph S_t.

Also, the distances between the centroids of the pairs of planes must be bounded

$$\left|(\mathbf{c}_j^r - \mathbf{c}_i^r) - (\mathbf{c}_{ii}^t - \mathbf{c}_{jj}^t)\right| < th_{dist} \tag{16}$$

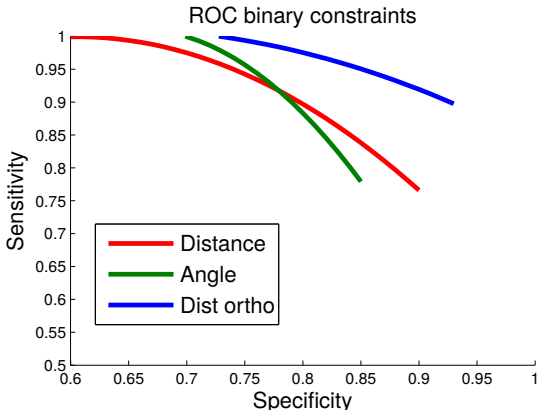

Figure 6: Comparison of the different binary constraints by their ROC curves (sensitivity *vs.* specificity).

The other binary constraint takes into account the orthogonal distance from one plane to the centroid of its neighbour. This distance must be similar when the two pair of planes are correctly matched,

$$\left|\mathbf{n}_i^r \cdot (\mathbf{c}_j^r - \mathbf{c}_i^r) - \mathbf{n}_{ii}^t \cdot (\mathbf{c}_{jj}^t - \mathbf{c}_{ii}^t)\right| < th_{ortho} \qquad (17)$$

Other constraints have been tested employing the distance between planes, and the direction of the principal vectors, however, these constraints did not improve significantly the search since they are highly sensitive to the partial observation of planes.

Similarly as for the unary constraints, the influence in classification performance of these constraints is analysed for a range of thresholds, and the final thresholds are selected by setting a minimum sensitivity of 99%. The estimated ROC curves showing the balance between sensitivity and specificity are shown in figure 6.

4.2.3. Interpretation tree

Algorithm 1 describes the recursive procedure for matching two subgraphs. It checks all the possible combinations defined by the nodes and edges of the subgraphs S_r and S_t, to find the one with the best score. In order to assign a new match between a plane from S_r and a plane from S_t the unary constraints are verified first (their result is stored in a look-up table to speed up the search), 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 the updated arguments. The algorithm finishes when all the possibilities have been explored, returning a list of corresponding of pairs planes with the highest score.

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 fulfil the unary and/or binary restrictions. In addition, the evaluation of these restrictions is very fast, since they only do simple operations to compare 3D vectors and scalars. The cost of this process depends linearly on the number of edges in the subgraphs, and the number of edges depends on the chosen distance defining neighbour planes and on the number of k-levels defining the subgraph. Thus, if the size of the subgraphs is limited the cost of the search depends linearly with the size of the PbMap where the robot tries to re-localise itself. This allows the search process to work at frame rate when the number of edges in the subgraphs is bounded (e.g. 10^{25} edges per subgraph, such a number of edges can be obtained by setting a large connectivity $k = 4$ with a distance threshold for neighbour planes of 1 m). By considering smaller, more reasonable connectivity levels $k = \{1, 2\}$ to define distinctive contexts of planes, this process performs in the order of microseconds.

4.3. Localization and rigid consistency

A consistency test is proposed here to evaluate the rigid correspondence of the matched planes of two subgraphs provided by the interpretation tree. This technique requires that at least 3 linearly independent (non parallel) planes are matched to estimate the relative pose between them. This is accomplished by minimizing a cost function which measures the adjustment error of each matched plane. Mathematically

$$\hat{\epsilon} = \underset{\epsilon}{\operatorname{argmin}} \sum_{i=1}^{N} e_i(\epsilon)^2 \qquad (18)$$

where N is the number of matched planes and $e_i(\epsilon)$ represents the adjustment error of a pair of planes P_{r_i} and P_{t_i} with respect to the rigid transformation defined by ϵ. This error corresponds to the distance from the centroid of P_i to its matched plane P_{m_i} (refer to figure 7). Hence, the proposed error function $e_i(\epsilon)$ is given by

$$e_i(\epsilon) = w_i \, \mathbf{n}_{m_i}(\exp(\epsilon)\mathbf{c}_i - \mathbf{c}_{m_i}) \qquad (19)$$

INPUT:
$S_r \leftarrow$ Reference subgraph
$S_t \leftarrow$ Target subgraph
$matched \leftarrow$ Matched planes in the branch (initially empty)
INPUT and OUTPUT:
$best\_match \leftarrow$ Final list of matched planes (initially empty)

$best\_match = MatchSubgraphs(best\_match, S_r, S_t, matched)$

1: **if** $Score(best\_match) > Score(matched) + sumScore(S_r)$ **then**
2: **return** $best\_match$
3: **end if**
4: **for each plane** $P_C \in S_r$ **do**
5: **for each plane** $P_M \in S_t$ **do**
6: **if** $EvalUnaryConsts(P_C, P_M) == False$ **then**
7: **continue**
8: **end if**
9: **for each** $\{P'_C, P'_M\} \in matched$ **do**
10: // Check if the edges $\{P_C, P'_C\}$ & $\{P_M, P'_M\}$ exist
11: **if** $\{P_C, P'_C\} \in S_C$ & $\{P_M, P'_M\} \in S_M$ **then**
12: **if** $EvalBinaryConsts(\{P_C, P'_C\}, \{P_M, P'_M\}) == False$ **then**
13: **continue**
14: **end if**
15: **end if**
16: **end for**

17: // Remove P_C from S_r and P_M from S_t
18: $new\_S_r = S_r - P_C$
19: $new\_S_t = S_t - P_M$
20: $new\_matched = matched \cup \{P_C, P_M\}$

21: // Explore this branch further down
22: $best\_match = MatchSubgraphs(best\_match, new\_S_r, new\_S_t, new\_matched)$
23: **end for**
24: **end for**

25: // Check the score of the new match
26: **if** $Score(new\_matched) > Score(best\_match)$ **then**
27: $best\_match = new\_matched$
28: **end if**

29: **return** $best\_match$

Algorithm 1: MatchSubgraphs.

being $\mathbf{n}_{m_i}$ the normal vector and $\mathbf{c}_{m_i}$ the centroid of P_{m_i}; $\mathbf{c}_i$ is the centroid of P_i, and $\exp(\epsilon)$ is the rigid transformation matrix in $\mathbb{SE}(3)$ represented as the exponential map of the 6D vector ϵ, which is a minimal parametrization for the relative pose, and w_i is the weight defined in eq. 11.

We solve this non-linear least squares problem using Gauss-Newton optimization. Notice that other scene alignment methods can be applied, like ICP or direct registration if the depth images or the point clouds are stored. The election of the one presented here is motivated because it does not require extra information apart from that already present in the PbMap, and it is simple and fast to compute.

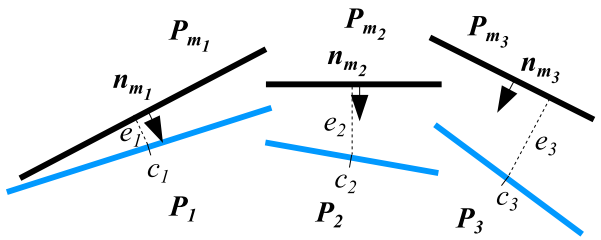

Figure 7: Consistency test. 2D representation of the depth error (the blue segments represent planes of the current subgraph and the black segments correspond to a previous subgraph).

After the above method has converged and the relative pose has been calculated, the resulting error can be used to evaluate the consistency of several candidate matches if there are several ones with high scores. For the case in which the PbMap is used to register range or RGB-D images, direct registration [10] is applied as a final check, which also results in higher accuracy of the registration as it employs all the information available in the images instead of only the planar patches.

5. Experimental validation

We present a set of experiments using RGB-D cameras Asus Xtion Pro Live (Asus XPL), and employing also a rig of such sensors to provide omnidirectional RGB-D images [3]. First, an experiment of place recognition is presented in different home and office environments to demonstrate the ability to recognize and register scenes from planar information using a hand-held Asus XPL. Also some results are presented in the context of life long mapping using PbMaps. A second set of experiments is presented in which loop closure is evaluated with both probabilistic and non-probabilistic PbMap registration. Scene tracking with PbMaps is also compared with ICP and direct image registration to demonstrate that

scene structure registration is also useful for odometry, in addition to re-localization and loop closure.

5.1. Scene recognition with a hand-held RGB-D sensor

This section presents the experiments carried out to validate our approach for place recognition with a hand-held RGB-D sensor. In this first set of experiments, the effectiveness for recognizing places is evaluated with 300 tests performed in an environment composed of 15 rooms. These experiments are divided in two subsections depending on the input data: range only or RGB-D, where the advantage of adding radiometric information to a plane-based geometric description is evaluated. The experiments are performed using RGB-D video sequences where we only use the depth images for the case of range only, so that the results of both can be compared. Then, 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 suffer changes continuously (this second experiment is performed using only range images). In these experiments we have employed an Intel Core i7 laptop with 2.2 GHz processor.

In the first battery of experiments we explore the scene with a hand-held RGB-D sensor, building progressively a PbMap while at the same time, the system searches for places visited previously. In order to build the PbMap, the pose of each frame is estimated with a method for dense visual odometry (also called direct registration) [10]. 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. While the scene is explored and the PbMap is built, the current place is continuously searched in a set of 15 previously acquired PbMaps corresponding to different rooms of office and home scenarios (these PbMaps generally capture a 360° coverage of the scene, see figure 8). An additional challenge of this experiment comes from the fact that some PbMaps represent the same type of room. This is an important issue for solutions based on bag-of-words since features are normally repeated in scenes of the same kind. In the case of PbMaps, this can also be problematic as some scenes share a similar layout.

Table 1: Effectiveness of the proposed method in different environments with different exploration trajectories (20 tests for each environment).

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

We have repeated 20 exploration sequences with different trajectories for each one of the 15 different scenarios. The success and failure rates for place recognition have been recorded, together with the average length of the sensor trajectory until a place was detected, or until the scene was fully observed when no place was recognized. Table 1 shows the recognition rate for these experiments. The first column indicates the percentage of cases where a place was recognized correctly, while the failure rate stands for the percentage of places recognized erroneously. 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%. 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 the local neighbourhoods of planes are for each different scenario. Nevertheless, note that the length of exploration is not directly related to the recognition rate, since even scenes with few distinctive subgraphs (e.g. the case of an empty room) can eventually be matched. An interesting feature of our approach is that it can recognize easily places where there is little appearance infor-

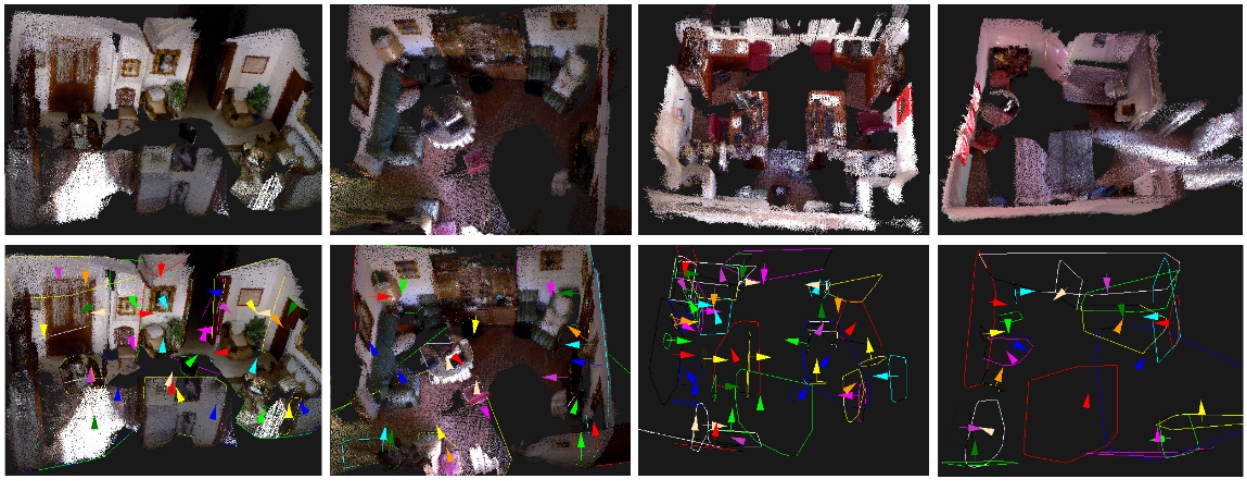

Figure 8: Different scenarios where place recognition has been tested. These pictures show the point clouds of some of the maps created previously, showing their PbMap right below of each scenario.

mation, but where the geometric configuration of planes is highly descriptive, this can be perceived in the video `http://youtu.be/uujqNm_WEIo`. In cases where there are fewer extracted planar patches the recognition rate drops.

The same experiment of the previous subsection, in which we explore 15 different scenes with 300 independent sequences is carried out here adding colour information. Regarding place recognition, or more concretely graph matching, we perceive two relevant improvements: first, the search is more efficient, and second, it is more robust to incorrect matches. The performance improvement is illustrated with an experiment which shows the number of constraints checked (which is directly proportional to the time required for searching a place) with respect to the subgraphs size, with and without the use of the colour descriptor. Figure 9 shows the average time of the search with respect to the number of planes being evaluated. We observe that performing the search using the proposed colour descriptor is around 6 times faster. Such a rate varies from 2 to 10 depending on the radiometric characteristics of the planar surfaces of the particular environment. This constitutes a significant increase of efficiency over the pure-geometric solution.

The radiometric information in PbMaps allows to distinguish different places with similar geometric layout but

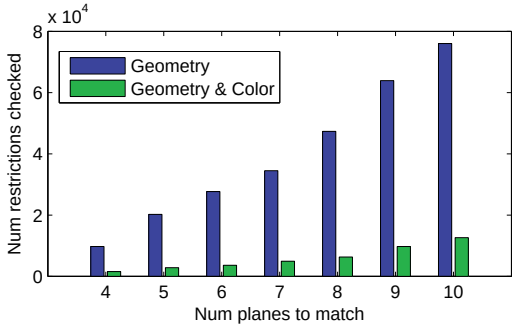

Figure 9: Performance of the place recognition process (in terms of the number of restrictions checked until matching with respect to the size of the subgraph to match) for both: only geometry and colour and geometry in PbMaps. The computing time is directly proportional to the number of restrictions checked.

different colour. That was the case in two bedroom environments of the previous experiment, where colour information helps to differentiate one from another. The results show that apart of the improvement on efficiency, the solution employing colour is more robust to incorrect matching. This is shown in table 2, where a significant reduction in the number of mismatched scenes is achieved.

A second battery of experiments shows that PbMaps can be used to recognize places that have suffered some changes, but where the main structure of the scene is un-

(a) (b) (c)

Figure 10: Lifelong maps in office environment. a) Reference scene (Ch0), b) Scene with moderate changes (Ch3), c) Scene with significant changes (Ch5).

Table 2: Robustness to wrong recognition by using colour information.

Scenario	*Failure rate (depth)*	*Failure rate (depth+colour)*
Office2	10%	0%
Office3	10%	0%
Hall2	10%	0%
Bedroom1	10%	5%
Bedroom2	20%	10%
Bedroom3	20%	5%
Bathroom	35%	30%

Table 3: Lifelong maps. The recognition shows the percentage of "finds" for 20 different trajectories exploring the scene.

Office1	*Ch0*	*Ch1*	*Ch2*	*Ch3*	*Ch4*
ICP res. (mm)	0	0.671	1.215	1.540	3.442
Recognition	100%	100%	95%	90%	80%
LivingRoom1	*Ch0*	*Ch1*	*Ch2*	*Ch3*	*Ch4*
ICP res. (mm)	0	1.182	2.010	2.942	3.863
Recognition	100%	100%	100%	95%	85%

changed. For that, we have evaluated the recognition rate with respect to the amount of change in the scene, which is measured as the average residual of the Iterative Closest Point (ICP) [8] on the point clouds built from the depth images. 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 10). The results of these experiments are summarized in Table 3, showing that the recognition rate remains high for moderate changes in the scene (Ch1 & Ch2, where chairs have been moved, and some objects like a laptop, have disappeared from the scene, while new objects have also 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 more previous planes and generating new ones).

5.2. PbMap registration with omnidirectional RGB-D

This section presents some preliminary experiments to validate our SLAM system. These experiments are carried out with a wheeled robot moving in 2D (see figure 11). Figure The PbMap registration can still work in 3D, with 6 degrees of freedom. In our experiments we employ a reduced resolution of the omnidirectional RGB-D images with 960×160 pixels (see figure **??**), since higher resolutions do not affect significantly the plane segmentation results and they have a higher computational cost. The depth images captured by the sensor are corrected as explained in [3], such correction takes around 2 ms per omnidirectional image. Several sequences are taken exploring different home and office environments, where the robot is remotely guided by a human at a maximum speed of 1 m/s.

In order to compare the performance of the previous PbMap registration approach with the new probabilistic variant, we check the amount of correct and wrong loop closure detections in a dataset of 1500 images taken in an

Figure 11: Robot with the omnidirectional RGB-D sensor.

(a)

(b)

Figure 12: a) Omnidirectional RGB-D camera rig and b) an image captured with it.

office environment using the platform shown in figure 11. The sequence was taken in a way that subsequent images are separated around 20 cm. We compare each image with all the others in the sequence without providing any information about the image ordering in the sequence, to simulate a loop closure scenario. Finally, the loop detection is verified by direct registration of the images. Thus an image will generally match those nearby in the sequence, plus other short sequences if the place is revisited. This experiment reproduces well the conditions of loop closure due to the limited traversable space of the robot in indoor scenarios and omnidirectional field of view of our sensor. The results of this experiment are shown in table 4 showing that the probabilistic approach for PbMap registration improves the number of correct detections, and more significantly, it reduces the number of wrong ones. Both alternatives have similar computation times.

Table 4: Evaluation of loop closure with both probabilistic and non-probabilistic PbMaps.

	Regular	*Probabilistic*	*Improv.*
Correct detections	7831	8165	**4.3** %
Wrong detections	389	196	**49.6** %

The performance of PbMab registration is compared with other registration approaches like ICP and direct registration. PbMap registration requires the segmentation of planar surfaces from the spherical RGB-D images, being such segmentation the most demanding task for registration. This stage is also parallelized to exploit our multicore processor to segment the planes of the spherical image in less than 20 ms. PbMap matching requires much less computation, in the order of microseconds. On the other hand, both ICP and direct registration also need a previous preparation to compute the spherical point cloud and the spherical images, respectively, before computing the matching. Table 5 presents the average computation time of these three methods for spherical RGB-D image registration, calculated from 1000 consecutive registrations (odometry). For that, both ICP and direct registration are performed using a pyramid of scales for robustness and efficiency. In this table, we can see how the registration based on PbMap is two orders of magnitude faster than the other two alternatives.

Besides the low computational burden, another important advantage of our registration technique with respect to classic approaches like ICP or direct registration is that we do not require any initial estimation. Thus, we can

Table 5: Average RGB-D sphere registration performance of different methods (in seconds).

	PbMap	*ICP*	*Dense*
PbMap construction (s)	0.019	-	-
Sphere construction (s)	-	0.010	0.093
Matching (s)	10^{-6}	1.53	2.12
Total Registration (s)	**0.019**	**1.54**	**2.22**

register images taken further away, while ICP and direct registration are limited to close-by frames unless a good initial estimation is provided. This fact is also illustrated in table 6, which shows the average maximum Euclidean distance between the registered frames of the previous sequence. For that, each frame is registered with all the preceding frames until tracking is lost, selecting the last registered frame as the furthest one.

Table 6: Average of the maximum distance for registration with different methods.

	PbMap	*ICP*	*Dense*
Registration dist. (m)	3.4	0.39	0.43

The registration of RGB-D images through PbMap permits to perform odometry estimation of the robot trajectory efficiently. This is done simply by registering the current frame to the previous one (see the video at `www.youtube.com/watch?v=8hzj6qhqpaA`). Figure 13 shows the trajectory followed by our sensor in one of our exploration sequences in a home environment together with the point clouds from each spherical image superimposed. The consistency of the resulting map indicates that each sphere is registered correctly with respect to the previous one, though yet, we can appreciate the drift in the trajectory which comes as a consequence of the open loop approach. This qualitative experiment shows that despite the compact information extracted for fast registration of the spherical images, the accuracy of registration is still good for many applications. More results and videos showing uses of PbMap for localization and mapping with omnidirectional RGB-D can be found in `https://sites.google.com/site/efernandezmoral/projects/rgbd360`.

6. Conclusions

A methodology for fast scene registration based on planar patches (PbMap) has been proposed for indoor, structured environments. The registration process is tackled with an interpretation tree, which matches efficiently local neighbourhoods of planes exploiting their uncertainty information, together with a set of weak constraints that prune the match space. We provide experimental results demonstrating the effectiveness of our approach for recognizing and registering places in a dataset composed of 20 home and office scenes: living rooms, kitchens, bathrooms, bedrooms, offices and corridors. A further study on the weak constraint parameters may be done to improve the performance of the solution, but we note that such study will always rely on the sample datasets, so that the gain in performance offered may not adapt to different scenarios.

Also, our method has the advantage to adapt to dynamic environments where humans or other elements are constantly moving, since the large planar surfaces taken into account for registration are generally static. Home and office environments are a good example for that, where people interact with the scene changing their positions and those of some objects like chairs, but where the scene structure remains unchanged. In order to test this idea, we performed an experiment to measure how the recognition performance is affected by the fact that objects can be moved by the users. The results confirm the intuition, though a deeper study must be carried out to evaluate the applicability of our representation for such a problem, which is left for future research.

While our registration method is mainly based on the observation of the scene's geometry by range sensors, a future improvement may be obtained by combining it with a method based on bag-of-words to increase the registration robustness when intensity information is available. Another open issue from this research is how to use the compact description of a PbMap for semantic inference, which can provide extra capabilities in mobile robotics and better communication interfaces human-robot [26]. Since the PbMap's compact description is useful to match scenes, it is reasonable that they can be useful to identify classes of scenes (e.g. kitchens, bedrooms, etc.) what is interesting for example in the context of domestic service robotics.

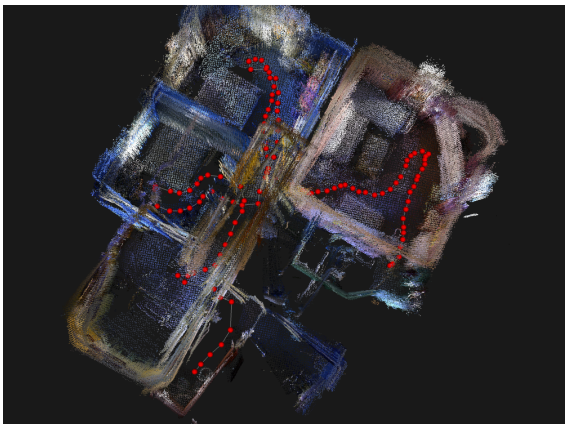

Figure 13: Trajectory of the sensor in a home environment composed of different rooms (the path is about 36 m).

7. ACKNOWLEDGMENT

This work has been carried out in collaboration between INRIA Sophia-Antipolis and Universidad de Málaga, and has been co-funded by the INRIA's post-doctoral fellowship and the Spanish Government under the research contract CICYT-DPI2011-25483.

References

[1] E. Fernández-Moral, W. Mayol-Cuevas, V. Arévalo, J. González-Jiménez, Fast place recognition with plane-based maps, in: International Conference on Robotics and Automation (ICRA 2013), IEEE, 2013.

[2] E. Garcia-Fidalgo, A. Ortiz, Vision-based topological mapping and localization methods: A survey, Robotics and Autonomous Systems 64 (2015) 1–20.

[3] E. Fernández-Moral, J. González-Jiménez, P. Rives, V. Arévalo, Extrinsic calibration of a set of range cameras in 5 seconds without pattern, in: Intelligent Robots and Systems (IROS), in IEEE/RSJ International Conference on, 2014.

[4] B. Zitova, J. Flusser, Image registration methods: a survey, Image and vision computing 21 (11) (2003) 977–1000.

[5] M. Cummins, P. Newman, Fab-map: Probabilistic localization and mapping in the space of appearance, The International Journal of Robotics Research 27 (6) (2008) 647–665.

[6] D. Galvez-Lopez, J. D. Tardos, Bags of binary words for fast place recognition in image sequences, IEEE Transactions on Robotics 28 (5) (2012) 1188–1197. doi:10.1109/TRO.2012.2197158.

[7] A. Oliva, A. Torralba, Building the gist of a scene: The role of global image features in recognition, Progress in brain research 155 (2006) 23.

[8] P. J. Besl, N. D. McKay, Method for registration of 3-d shapes, in: Robotics-DL tentative, International Society for Optics and Photonics, 1992, pp. 586–606.

[9] T. Tykkala, C. Audras, A. I. Comport, Direct iterative closest point for real-time visual odometry, in: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE, 2011, pp. 2050–2056.

[10] C. Audras, A. Comport, M. Meilland, P. Rives, Real-time dense appearance-based slam for rgb-d sensors, in: Australasian Conf. on Robotics and Automation, 2011.

[11] R. Cupec, E. K. Nyarko, D. Filko, A. Kitanov, I. Petrović, Place recognition based on matching of planar surfaces and line segments, International journal of robotics research 278 (2015) 3649.

[12] J. Weingarten, R. Siegwart, 3D SLAM using planar segments, in: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, 2006, pp. 3062–3067.

[13] M. J. Milford, G. F. Wyeth, Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights, in: Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE, 2012, pp. 1643–1649.

[14] M. Milford, Vision-based place recognition: how low can you go?, The International Journal of Robotics Research 32 (7) (2013) 766–789.

[15] R. B. Rusu, S. Cousins, 3d is here: Point cloud library (pcl), in: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, 2011, pp. 1–4.

[16] E. Fernández-Moral, Contributions to metric-topological localization and mapping in mobile robotics, Ph.D. thesis, Universidad de Málaga (2014).

[17] D. Holz, S. Behnke, Fast range image segmentation and smoothing using approximate surface reconstruction and region growing, in: Intelligent Autonomous Systems 12, Springer, 2013, pp. 61–73.

[18] K. Pathak, A. Birk, N. Vaskevicius, J. Poppinga, Fast registration based on noisy planes with unknown correspondences for 3-d mapping, IEEE Transactions on Robotics 26 (3) (2010) 424–441.

[19] K. Khoshelham, S. O. Elberink, Accuracy and resolution of kinect depth data for indoor mapping applications, Sensors 12 (2) (2012) 1437–1454.

[20] E. Fernández-Moral, V. Arévalo, J. González-Jiménez, A compact planar patch descriptor based on color, in: International Conference on Informatics in Control, Automation and Robotics (ICINCO), 2014.

[21] K. Pathak, N. Vaskevicius, F. Bungiu, A. Birk, Utilizing color information in 3d scan-registration using planar-patches matching, in: Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on, 2012, pp. 371–376.

[22] C. Kerl, J. Sturm, D. Cremers, Robust odometry estimation for rgb-d cameras, in: IEEE Int. Conf. on Robotics and Automation (ICRA), 2013.

[23] T. Gokhool, M. Meilland, P. Rives, E. Fernández-Moral, A Dense Map Building Approach from Spherical RGBD Images, in: International Conference on Computer Vision Theory and Applications (VISAPP 2014), Lisbon, Portugal, 2014.

[24] P. Hansen, N. Mladenović, J. A. M. Pérez, Variable neighbourhood search: methods and applications, 4OR 6 (4) (2008) 319–360.

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

[26] J. R. Ruiz-Sarmiento, C. Galindo, J. González-Jiménez, Joint categorization of objects and rooms for mobile robots, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015.