# Visual Registration Method For A Low Cost Robot

David Aldavert
Ricardo Toledo
Computer Vision Center (CVC)
Dept. Ciències de la Computació
Universitat Autònoma de Barcelona (UAB)
08193, Bellaterra, Spain
Email: aldavert@cvc.uab.cat, ricard@cvc.uab.cat

Arnau Ramisa
Ramon López de Mántaras
Computer Vision Center (CVC)
Artificial Intelligence Research Institute (IIIA-CSIC)
Campus UAB
08193, Bellaterra, Spain
Email: aramisa@csic.iiia.es, mantaras@iiia.csic.es

***Abstract*— An autonomous mobile robot must face the correspondence or data association problem in order to carry out tasks like place recognition or unknown environment mapping. In order to put into correspondence two maps, most correspondence methods first extract early features from robot sensor data, then matches between features are searched and finally the transformation that relates the maps is estimated from such matches. However, finding explicit matches between features is a challenging and computationally expensive task. In this paper, we propose a new method to align obstacle maps without searching explicit matches between features. The maps are obtained from a stereo pair. Then, we use a vocabulary tree approach to identify putative corresponding maps followed by a Newton minimization algorithm to find the transformation that relates both maps. The proposed method is evaluated on a typical office dataset showing good performance.**

## I. Introduction

An autonomous mobile robot that navigates through an unknown environment often has to carry out tasks such as closing-loop detection, estimate motion from robot sensors or build a map using some SLAM algorithm. To solve such problems we must face the correspondence (or data association) problem, i.e. the problem of determining if sensor measurements taken at different locations or at different time correspond to the same physical object in the world.

This problem is usually approached extracting primitives from sensor measurements and searching correspondences between them. From such correspondences an estimation of the robot motion and its uncertainty is obtained. In [1], Cox extracts points from laser scans and uses them as primitives. Then point primitives are matched to lines from a map given *a priori*. In [4], Lu and Milios propose the IDC (Iterative Dual Correspondence) which is a more general approach that matches points to points. As Cox's algorithm performs better in structured environments and IDC in unstructured environments, Gutmann combines both methods in [3]. The IDC is a variant of the ICP (Iterative Closest Point) algorithm [5] applied to laser range scans. The ICP is also used to align robot measurements, specially when using 3D range data [6, 7].

Computationally, the search of explicit correspondences is the most expensive step. The performace is poor because for each primitive of a set a test against all the primitives from the other set must be done. Therefore, other methods attempted to avoid this step aligning sensor measurements without finding direct correspondences between primitives. In [2], Weiss and Puttkamer build histograms of sensor measurements and search the parameters that best align both scans using a correlation measure. This method is designed to work in very structured environments, so, when applied in unstructured environments the results tend to be poor. In [9], Biber and Straßer presented the Normal Distributions Transform, which is a more general approach to align scans obtained from a laser range scanner. This method divides the space into cells forming a grid. Then, to each cell, they assign a normal distribution, which locally models the probability of measuring an obstacle. Finally, the Newton's algorithm is used to align a laser scan input to the probability distribution.

The methods commented above use range information which is not discriminative enough to directly find correct correspondences between primitives, so that, such methods iteratively search the matching primitive. Using image data, robust local invariant features can provide primitives that are distinctive enough to search matches directly without using an iterative approach [12, 13]. However, there are situations where image local invariant features cannot be used to describe the world. For example, in poorly textured environments, the number of putative matches usually is not enough to ensure that the estimated robot motion is correct. In environments with repetitive textures, the amount of false correspondences rises rapidly and the transformation that relates both scans cannot be estimated reliably. These two problems are common in indoor or urban environments.

In this paper, we present a method to align local maps using stereo image data. The maps are obtained from different locations and the alignment is done without establishing direct correspondences between map primitives. First, local obstacle maps are obtained by scanning the environment with a stereo head. Then, using a *bag of features* [25, 24] inspired approach, signatures of obstacle maps are built with robust invariant features extracted from stereo images. Such map signatures are used as a fast method to determine if two maps are likely to be related or not. Finally, a Newton minimization algorithm is used to iteratively determine robot motion. Our minimization algorithm searches explicit correspondences be-

tween primitives, which is a computationally expensive step. However, as our obstacle space is discrete and obstacles are restricted into the ground plane, the matching step can be greatly speed up. Moreover, color image information is added to the probabilistic map in order to increase the convergence ratio and the robustness of the alignment estimation.

The paper is structured as follows: In section II methods used to build local obstacle maps and to obtain map signatures are presented. In section III the method used to align different obstacle maps is described. The experiments set-up and results are shown in section IV. Finally in section V there is a discussion of the results and an overview of future work.

## II. Local stereo maps

The payload of our robot cannot currently afford an extra laser range finder sensor, anyway, a stereo head mounted on a pan-tilt unit is used to obtain depth information. Since the field of view of the cameras is small, the robot pans the stereo head to obtain several views and create a obstacle map covering a wider area. Unlike range sensors, stereo camera pairs cannot directly obtain depth information and a dense stereo algorithm [14] is required. However, stereo images also provide illumination, color and texture information, which can be directly added to the obstacles. Besides, image information is used to obtain a signature that identifies the obstacle map by extracting robust invariant features from the images with a *bag of features* inspired approach.

### A. Obstacle maps

Obstacle maps are represented by a 2D occupancy grid in the X-Z world plane where each cell represents the probability that an obstacle is present. Obstacles are detected using a correlation based algorithm that uses the SAD (Sum of Absolute Differences) function as similarity measure which, for a relatively small resolution can obtain a dense stereo map in real time [8]. In our approach, several expensive refinements of the method described in [8], such as the left-right consistency check, are removed in order to reduce the computational cost of the algorithm. To remove possible inconsistencies due to occlusions, the resulting disparity map is segmented using the watershed algorithm [15] and small disparity regions are further removed from it.

Once the dense stereo map is obtained, image points are transformed from pixel coordinates to image plane coordinates so that points can be reprojected to 3D space by simply using a noise free triangulation operation. Let $m_l = [x_l, y_l]$ and $m_r = [x_r, y_r]$ be a corresponding pair of points in image plane coordinates. The 3D coordinates can be computed as follows:

$$X = \frac{bx_l}{x_l - x_r} \qquad Y = \frac{by_l}{x_l - x_r} \qquad Z = \frac{bf}{x_l - x_r} \ . \qquad (1)$$

where $b$ is the baseline and $f$ is the focal length of the camera [23]. The resulting 3D world points that are within a height range, say $[Y_1, Y_2]$, are reprojected to a 2D occupancy grid in the X-Z world plane. Cells without the minimum support to be considered an obstacle, and isolated cells are removed from the 2D occupancy grid.

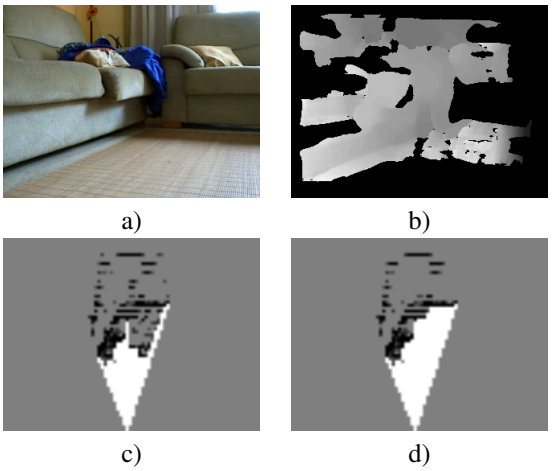



Fig. 1. a) Original right stereo pair image. b) Dense disparity map filtered with watershed segmentation. c) Occupancy grid obtained without filtering small regions. d) Occupancy grid obtained after filtering small disparity regions.

The occupancy grid and the subsequent filtering reduce the effects of the inconsistencies in the dense stereo algorithm. Therefore a perfect stereo reconstruction is not required and we can gain speed avoiding refinements like the left-right consistency check. Figure 1 shows how a local map is built: First, a dense disparity map (Fig. 1.b) is obtained from stereo image pairs (Fig. 1.a). Although the disparity maps have gaps in poorly textured regions, obstacles are found in the occupancy grid (Fig. 1.c). Filtering small disparity map regions, a more accurate occupancy grid can be obtained (Fig. 1.d).

The local map is built by making a scan from $-60°$ degrees to $60°$ degrees and taking stereo head measurement at steps of $10°$ degrees. The rotation error from the pan & tilt unit servo motors is quite small (about $0.5°$ degrees) compared to the obstacle map resolution and to the stereo depth estimation error, therefore, the location of the stereo head cameras at each scan step can be estimated *a priori*. Once stereo cameras location at each step is known, cells of the local map that are seen from each location are also known. As the measurements are taken at steps of $10°$ degrees and the stereo cameras field of view is about $42°$ degrees, several cells can be seen from several stereo head locations. Therefore, as the uncertainty of depth estimation decreases for points that are near to the horizontal central image point, each cell is assigned to the stereo pair that minimizes this uncertainty [23].

### B. Maps signature

Easily, a robot mapping a fairly large environment can store up to thousands of scans. Therefore, finding correspondences for a new scan in the database using only the alignment method can be computationally expensive. In order to filter most of the unrelated scans, a visual appearance based signature is extracted for each newly acquired scan and used to select the

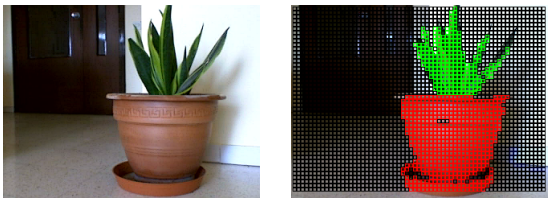
most similar instances of the database.

The signature used is based on the bag of words document retrieval methods, that represent the subject of a document by the frequency in which certain words appear in the text. Recently these approach has been adapted to visual object recognition by different authors [24][25] using local descriptors computed on image features as *visual words*. First a clustering algorithm e.g. k-means is used to sample the descriptor space, that usually is extremely big, to a more tractable size of thousands of codewords that can be represented in a signature histogram. Finally a classifier is trained with signatures from different classes. In this work we have used a technique inspired by the approach of Nistér and Stewénius [20] because it has very efficient temporal and spatial complexity.

Instead of a normal *k*-means, Nistér and Stewénius use its hierarchical version to build a codebook tree with branch factor *k* from a database of local descriptors extracted from training images. This representation allows to classify new descriptors in logarithmic time instead of linear. Another advantage of this approach is the inverted files mechanism, which accelerates the comparison of a test scan with the database stored in the memory of the robot.

In the work of Nistér and Stewénius the MSER [22] covariant region detector and the SIFT [26] descriptor are used. In our experiments we have evaluated the performance of three types of descriptors: Shape Context, Steerable Filters and SIFT [11], computed on regions detected by five state-of-the-art region detectors: Harris Affine, Hessian Affine, Harris Laplace, Hessian Laplace [10], MSER and SURF [21]. However, indoor environment are usually poorly textured and few regions are detected. Since, the number of features is the most influential parameter governing the performance [27], combinations of different detectors with complementary properties have been also evaluated. For instance, Harris-Laplace detects corner-like regions and Hessian-Laplace detects blobs like regions are combined, but Hessian-Laplace and SURF detectors are not tested together as both detects blob like regions. The evaluation results are explained in section IV.

### C. Color obstacle maps



a) b)

Fig. 2. Original image (a) is segmented obtaining (b).

Instead of using only depth information to build local environment maps, color information is added to each 2D occupancy grid cell in order to improve alignment results. Essentially, the 2D occupancy grid is divided into 4 layers, three layers for colors red, green and blue and one for grayish obstacles. In order to achieve a certain degree of invariance to illumination changes, we have trained a *Support Vector Machine* (SVM) [28] for color image segmentation.

The input vector of the SVM is built as follows: the image is transformed to the Hue-Saturation-Lightness (HSL) color space and a color descriptor is built from a region around each pixel. The descriptor is a histogram of six bins. Each pixel of the region votes in a bin determined by its *hue* and weighted by its *saturation*. The output of the SVM determines to which layers the obstacle pixels contribute. To train the SVM, a dataset of 100 images acquired with the robot cameras has been manually annotated.

In Fig. 2, a segmentation example is shown. Black pixels correspond to regions assigned to the grayish layer and red and green pixels corresponds to the objects assigned to the layer red and green respectively.

Once the occupancy grids are built, if a color layer cell has not enough support (the value of the bin is lower than the $1\%$ of the sum of all the map bins), the cell value is set to zero and its contribution is added to the grayish layer.

## III. Map alignment

In this section we present the method used to align different local maps. The first step uses the signature of the map to filter maps into the database and obtain candidates to be aligned. Then, for the selected maps a Newton based iterative algorithm is used to find the registration parameters.

### A. Signature comparison

When a new scan is acquired by the robot, its appearance signature described in section II-B is computed and compared to the ones stored in the memory of the robot. Next the *k* most similar scans of the database are selected and registered using the alignment method described in the next subsection. In our experiments we have used the Euclidean distance as a similarity measure between the signature histograms given that it is widely used in the literature [27, 25, 24, 20].

### B. Iterative map alignment

In order to align a new map with a map in the database, we need to find the transformation that relates both maps. As the robot moves in an indoor environment, we can assume a planar ground, then, to align a query map against a the database map we need to estimate a 2D rigid transformation:

$$M = \begin{bmatrix} cos\beta & -sin\beta & t_x \\ sin\beta & cos\beta & t_z \end{bmatrix} . \quad (2)$$

where $\beta$ is the rotation between the to maps in the $Y$ axis and $t_x$ and $t_z$ are the translation between the to maps in the $X$ and $Z$ axis respectively. To find the parameters $p = [\beta, t_x, t_z]^\top$ of equation 2, an iterative method which at each step reduces the distance between the obstacles of the maps query map and the database map is used. This algorithm tries to find the best parameters $p$ that minimizes the following function:

$$sc(p) = \sum_{c=1}^{4} \sum_{j=1}^{M} \sum_{i=1}^{N} e^{-\mathbf{x}_i^{c\top} C_i^{c-1} \mathbf{y}_j^c} . \quad (3)$$

where M and N are respectively the number of obstacles in the query and database map, $\mathbf{y}_j^c$ is a vector with the location coordinates of the $j$-th obstacle with color $c$ of the query map, $\mathbf{x}_i^c$ is a vector with the location coordinates of the $i$-th obstacle with color $c$ of the database map and $C_i^c$ is the covariance matrix modelling location uncertainty of $\mathbf{x}_i^c$:

$$C_i = RJ \begin{bmatrix} \sigma_{l_x} & 0 \\ 0 & \sigma_{r_x} \end{bmatrix} J^\top R^\top \ . \tag{4}$$

where $\sigma_{l_x}$ and $\sigma_{r_x}$ are the pixel localisation error, which is determined by camera calibration error statistics and $J$ is the Jacobian matrix that maps error from image coordinates to space coordinates:

$$J = \begin{bmatrix} \frac{-bx_r}{d^2} & \frac{bx_l}{d^2} \\ \frac{-bf}{d^2} & \frac{bf}{d^2} \end{bmatrix} \ . \tag{5}$$

where $d = x_l - x_r$ is the disparity between $x_l$ and $x_r$ expressed in image plane coordinates, $b$ is the baseline and $f$ is the focal length of the camera. The rotation matrix $R$ is expressed as follows:

$$R = \begin{bmatrix} cos\beta & -sin\beta \\ sin\beta & cos\beta \end{bmatrix} \ . \tag{6}$$

where $\beta$ is the pan unit rotation angle.

Then, to align two obstacle maps, the following method is proposed:

1) Initialise the motion parameters to zero or by an estimation obtained from the robot odometry.
2) Apply the parameters of the transformation to the set of points $S$ corresponding to the location of the obstacles in the query map.
3) From eq. 3 a score value is obtained.
4) Estimate new parameters values by optimizing the score using a Newton minimization algorithm.
5) While the convergence criterion is not meet, go to 2.

Given that eq. 3 is non-linear, to find the parameters $p$ that maximize eq. 3 the Newton's algorithm is used. This method is similar to other computer vision methods used for registration of image information obtained from different sensors [17] or aligning images related by an affine or projective transformation [18, 16]. The Newton's algorithm iteratively finds the parameters $p$ that maximize eq. 3. At each iteration it solves the following eq.:

$$\triangle p = -H^{-1} g \tag{7}$$

where $g$ is the gradient of eq. 3 with elements:

$$g_i = \frac{\partial sc(p)}{\partial p_i} \tag{8}$$

and $H$ is the Hessian of eq. 3 with elements:

$$H_{ij} = \frac{\partial sc(p)}{\partial p_i \partial p_j} \tag{9}$$

Then, the parameters are updated using the following eq.:

$$p = p + \triangle p \ . \tag{10}$$

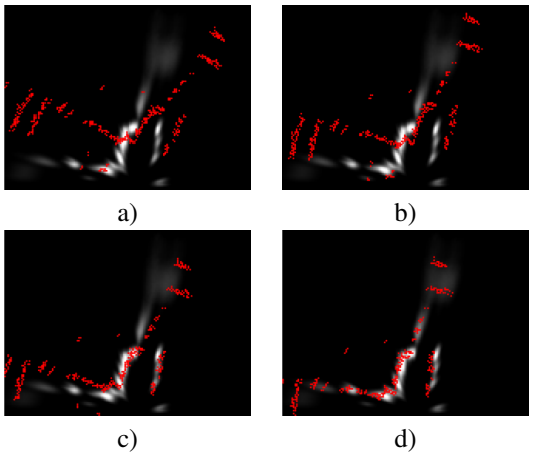


Fig. 3. Alignment of two local maps after: a) first iteration, b) 10 iterations, c) 20 iterations and d) final solution after 42 iterations.

Equations 7 and 10 are iterated until the estimate of $p$ converges. For each obstacle $\mathbf{y}$ of the query obstacle map, the elements of the gradient are, by the chain rule:

$$g_i = \frac{\partial sc(p)}{\partial p_i} = \frac{\partial sc(p)}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial p_i} \tag{11}$$

where the partial derivate of $sc(p)$ respect $\mathbf{y}$ is the gradient of the eq. 3 and the partial derivate of $\mathbf{y}$ respect $p_i$ are given by the Jacobian of the alignment transformation. As robot motion in a indoor or urban environments can be modelled as a 2D rigid transformation, i.e. translation in the $x$ and $z$ axis and rotation in the $y$ axis, the Jacobian is:

$$\frac{\partial W}{\partial p} = \begin{bmatrix} -x_i sin\alpha - y_i cos\alpha & 1 & 0 \\ x_i cos\alpha - y_i sin\alpha & 0 & 1 \end{bmatrix} \ . \tag{12}$$

The Hessian matrix $H$ is given by:

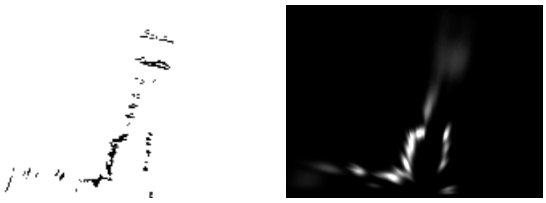
$$H = \sum_j \left[ \frac{\partial sc(p)}{\partial \mathbf{y}} \frac{\partial W}{\partial p} \right]^\top \left[ \frac{\partial sc(p)}{\partial \mathbf{y}} \frac{\partial W}{\partial p} \right] \tag{13}$$

Finally, the algorithm is iterated until a max number of iterations is reached or the update of the parameters fulfill the condition $\| \ \triangle p < \varepsilon \ \|$. Robot motion estimation is obtained from parameters vector $p$ and the uncertainty of such estimation, i.e. the covariance matrix, is obtained from the inverse of the Hessian matrix. The value of Eq. 3 is used together with the number of iterations spent by the alignment process to determine the feasibility of the obtained parameters. Figure 3 depicts an example where obstacles of the query map gradually converges to the obstacles of one of the database map.

### C. Optimizing the minimization process

As seen in section II, obstacles are detected using a 2D occupancy grid, so that, the location of the detected obstacles is discrete over the $X - Z$ plane. Therefore, for each location $\mathbf{y} = (i, j)$ of a 2D occupancy grid, we can calculate *a priori* its score:

$$SM(\mathbf{y}, c) = \sum_i e^{-\mathbf{x}_i^c C_i^{c\,-1} \mathbf{y}} \ . \tag{14}$$



a) b)

Fig. 4. a) Obstacles 2D occupancy grid. b) Probability distribution built from the 2D occupancy grid.

Where $y$ is the coordinates vector of the $j$ obstacle in channel $c$ of the query map. Then, eq. 3 can be reduced to:

$$sc(p) = \sum_{c=1}^{4} \sum_{j=1}^{M} SM(y_j, c) \tag{15}$$

Therefore, at each step of the minimization process we have to look up the value of eq. 14 avoiding to calculate the exponentials of eq. 3. Although this calculation of eq. 14 is computationally expensive, it only has to be done once and it greatly speeds up the matching algorithm. Figure 4 shows how the values of the matrix $SM(y_j, c)$ (Fig. 4.b) is formed from a 2D occupancy grid (Fig. 4.a).

## IV. Experimental Results

In this section, we analyze the results obtained with our alignment algorithm. To perform the experiments, a database of 50 panoramas has been acquired in a typical indoor environment. A testing environment with poor salient visual features was chosen in order to test the reliability of our method. The, we manually annotated the relations and the alignment parameters between the panoramas to create the ground truth. The ground truth has 236 correct relations out of the 2450 possible local map relationships.

Fig. 5. Robot used in the experiments.

Data has been acquired using a mobile robot platform built at our department (see Figure 5). It is based on a Lynxmotion 4WD3 robot kit and it has been designed to be as cheap as possible. All the experiments are executed on the robot's on-board computer, which is a VIA Mini-ITX EPIX PE computer with a VIA C3 1 GHz CPU, and stereo images are obtained from two Philips SPC900NC webcams with a resolution of $320 \times 240$ pixels. Stereo measurements are stored in a 2D occupancy grid that has 160 cell width per 120 cell height. Each cell represents a square with a side length of 0.05 meters, so that, the local map has a width of 8 meters and a depth of 6 meters.

### A. Map identification

As explained in Section II-B, for a query map, the $k$ most similar database maps are selected using a fast visual appearance method. This way the more computationally expensive alignment step does not have to be applied to every database map. To find the best parameters for our vocabulary tree, we have trained several trees using different depths and branch factors. Taking into account both, performance and computational cost, we have selected a branch factor of 5 and a depth of 6, so that the resulting tree has 15.625 leaves.

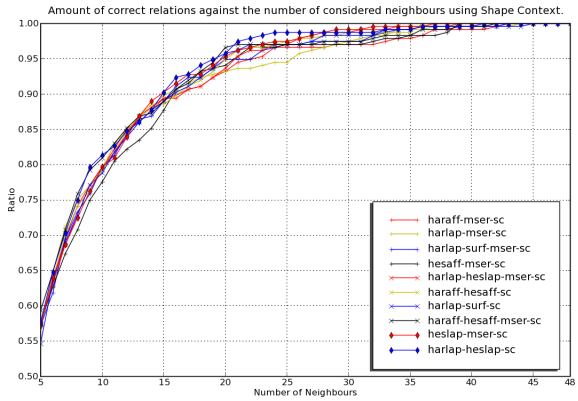
As has been mentioned in Section II-B, the performance of a bag of features inspired approach is typically improved with more detected features. Therefore it is interesting to combine complementary feature detectors. However, the computational cost of combining all possible feature detectors is prohibitive in our approach. Instead we have evaluated all the possible combination in order to find the one that maximizes performance while minimizing the number of applied detectors. With a similar idea in mind, we have compared three region descriptors with significantly different dimensionality. With a similar performance it is favorable to chose the descriptor with lowest dimensionality. Figure 6.a shows the average ratio of correct map relations (vertical axis) that are included between the $k$ most similar appearance signatures (abscissa axis) using the shape context descriptor. Figure 6.b shows the same results but using the SIFT descriptor. Results of the tests using steerable filters are omitted for space reasons because they performed worse than the other two descriptors.

We require that $90\%$ of $k$ local maps selected by the vocabulary tree are truly related to the query local map. This ratio is achieved by the combination of detectors Hessian-Affine and Harris-Affine using the SIFT descriptor at $k$ equal to $14$. Another interesting option would be the combination of Harris-Laplace and Hessian-Laplace detectors with Shape Context descriptor, that achieves $90\%$ with $k$ equal to $15$. Even though an extra map alignment has to be done, the computational cost of the Laplace version of the detectors is much lower than its Affine counterpart. Besides the Shape Context has a markedly lower dimensionality than SIFT, therefore it needs less computational effort and is more scalable, so that it might be an interesting option for mapping larger environments.

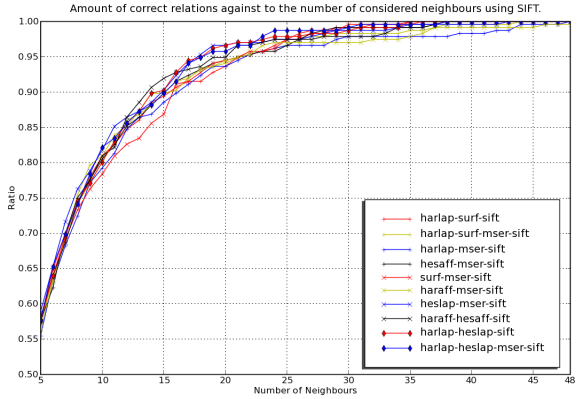
### B. Map alignment

To evaluate the performance of the alignment algorithm, we have selected a set of 10 significant samples of indoor scenes, e.g. corridor, dinning room, office, etc. . . and for each scene a set of local maps with different amounts of overlapping are built.

Figures 7 and 8 show the mean ratio of correctly aligned maps. From Figures 7 it is shown that the method can deal




a)




b)

Fig. 6. Average ratio of correct relations according to ground truth among the $k$ most similar appearance signatures using a) the Shape Context descriptor and b) the SIFT descriptor.

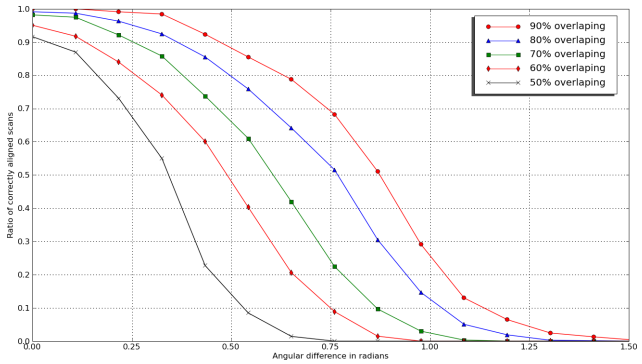



Fig. 7. Ratio of correctly aligned scans for different starting rotation values and different amount of overlapping between the aligned maps.

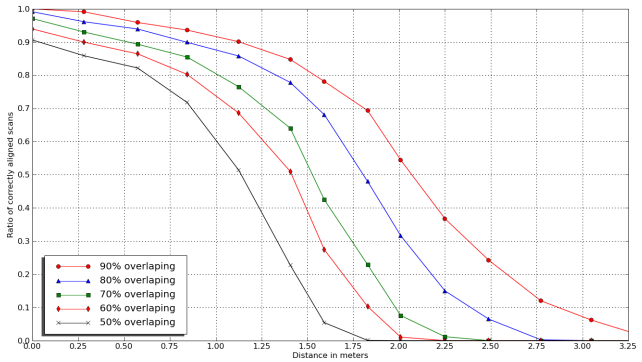



Fig. 8. Ratio of correctly aligned scans for different starting translation values and different amount of overlapping between the aligned maps.

with rotations up to 45 degrees quite well. This is because our testing environment is highly structured, so that, for rotations greater than 45 degrees the method usually falls into a local minima. From Figure 8 it is shown that for translations up to 1.5 meters the behavior of the method is acceptable taking into account that the size of our testing environment is relatively small (e.g. rooms are no longer than 3 meters). Results could be improved applying common techniques to avoid local minima such as random restart, but this increases the computational complexity of the alignment algorithm.

### C. Performance Evaluation

Finally, we evaluate the performance of the whole system and we compare it against direct matching. Direct matching computes the alignment between the panoramas estimating correspondences between robust features. In this experiment, the Harris-Affine and Hessian-Affine covariant features detectors and the SIFT descriptor have been used. First, putative matches between features of the query map and a database map are searched using the same technique as [26]. Then, features are projected to the X-Z plane using stereo information and the RANSAC algorithm [23] is used to reject possible false matches and find the 2D rigid transformation between the two maps. The ratio between correct matches and total putative matches is used to reject false map relations.

Figure 9 compares the performance of the proposed method and the direct matching method. We have evaluated four different variants of the proposed method: using both color and vocabulary tree (CIter + VT), vocabulary tree without color (Iter + VT), only color information (CIter) and the iterative alignment algorithm alone (Iter). As can be seen, direct matching performs significantly worse than all the variations of the proposed method. Using the vocabulary tree increases the recall until reaching the limit imposed by the number of selected neighbors (90.0%), as explained in section IV-A, while using color slightly increases the performance of the method. The most noticeable effects of the vocabulary tree is the raise of the precision thanks to the filtering of most false relations between maps.

Finally, regarding time complexity, the alignment method requires about 5 ms per iteration and 14.7 iterations are required in average to align two maps. Therefore, the mean time elapsed in the alignment step is about 73 ms. Classifying a map on the vocabulary tree takes about 40 ms, and given that 15 database maps are selected by the vocabulary tree, 1.1 seconds are required to find the possible relations between a query map and the database maps.

## V. Conclusion

The first contribution of this paper is a method to build local maps from information acquired by a stereo head. The

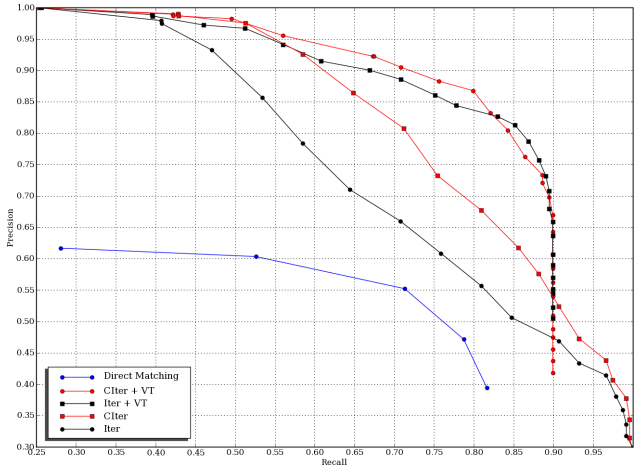



Fig. 9. Precision vs. Recall for each evaluated method.

local map provides information about the distribution of the obstacles in the X-Z plane and also stores color information. Next, we proposed a method that uses a Newton minimization algorithm to align these local maps. To avoid aligning unrelated maps with similar geometrical layout but different visual appearance, we use a *vocabulary tree* approach. Both methods avoid the expensive step of searching implicit feature correspondences. The obtained results shows that the *vocabulary tree* effectively filters unrelated maps and, combined with the alignment method, up to $87.3\%$ of the relations of our data set are correctly detected keeping a good precision. This method performs well in environments with few visual salient features, where methods based on feature matching tend to fail.

Future work includes the testing of our method using larger data sets including different types of environments. Finally, we want to implement a faster version of this schema trying to reach a real time mapping performance.

## ACKNOLEGMENT

This work has been partially funded by TIN 2006-15308-C02-02 project grant of the Ministry of Education of Spain, the CSD2007-00018 Consolider Ingenio 2010, the FI grant and the BE grant from the AGAUR, the European Social Fund, the 2005/SGR/00093 project, supported by the Generalitat de Catalunya, the MIDCBR project grant TIN 200615140C0301, TIN 200615308C0202 and FEDER funds.

## REFERENCES

[1] Cox, I.J.: Blanchean experiment in guidance and navigation of an autonomous robot vehicle. IEEE Transactions on Robotics and Automation, 7(2) (1991), 193203.

[2] Weiss, G., von Puttkamer, E.: A map based on laserscans without geometric interpretation. Intel. Aut. Systems, Karlsruhe, Germany, March 27-30, (1995), 403-407.

[3] Gutmann, J.-S., Schlegel, C.: AMOS: comparison of scan matching approaches for self-localization in indoor environments. Eurobot, p. 61, (1996).

[4] Lu, F., Milios, E.: Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. J. of Intelligent and Robotic Systems 18, 3 (Mar. 1997), 249-275.

[5] Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. Int. J. of Computer Vision 13, 2 (Oct. 1994), 119-152.

[6] Nüchter, A., Surmann, H., Lingemann, K., Hertzberg, J., Thurn, S.: 6D SLAM with an Application in autonomous mine mapping. In Proc. of the IEEE Int. Conf. of Robotics and Automation, New Orleans, USA, (April 2004), 1998-2003.

[7] Nieto, J., Bailey, T., Nebot, E.: Recursive scan-matching SLAM. Robotics and Autonomous Systems, 55, 1 (Jan. 2007), 39-49.

[8] Mühlmann, K., Maier, D., Hesser, J., and Männer, R.: Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation. Int. Journal of Computer Vision 47, 1-3 (Apr. 2002), 79-88.

[9] Biber, P., Straßer, W.: The Normal Distributions Transform: A New Approach to Laser Scan Matching, IEEE/RJS Int. Conf. on Intel. Robots and Systems, (2003).

[10] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors, Int. Journal of Computer Vision 65, 1/2, (2005), 43-72,

[11] Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors, IEEE Transactions on Pattern Analysis & Machine Intelligence, 10, 27, (2005), 1615-1630.

[12] Newman, P., Ho K.: SLAM- Loop Closing with Visually Salient Features, Int. Conference on Robotics and Automation, Barcelona (2005).

[13] Se, S., Lowe, D., Little, J.: "Vision-based Mobile robot localization and mapping using scale-invariant features, Proceedings of the IEEE Int. Conference on Robotics and Automation (ICRA), Seoul, Korea (May 2001), 2051-2058.

[14] Brown, M. Z., Burschka, D., and Hager, G. D.: Advances in Computational Stereo, IEEE Trans. on Pattern Analysis & Machine Intel. 25, 8 (Aug. 2003), 993-1008.

[15] Vincent, L., Soille, P.: Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations, IEEE Trans. on Pattern Analysis and Machine Intel., vol. 13, no. 6, (1991), pp. 583-598.

[16] Shum, H., Szeliski. R.: Panoramic image mosaics, Technical Report TR-97-23, Microsoft Research, 1997.

[17] Keller, Y., Averbuch, A.: Robust Multi-Sensor Image Registration Using Pixel Migration, IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), Washington D.C, USA. August 2002.

[18] Baker, S. and Matthews, I.: Lucas-Kanade 20 Years On: A Unifying Framework, Int. Journal of Computer Vision 56, 3 (Feb. 2004), 221-255.

[19] Blank, D., Kumar, D., Meeden, L., and Yanco, H.: Pyro: A python-based versatile programming environment for teaching robotics. Journal on Educational Resources in Computing (JERIC). 3, 4 (Dec. 2003), 1.

[20] Nister, D. and Stewenius, H.: Scalable Recognition with a Vocabulary Tree. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Washington D.C, USA, (June 2006), 2161-2168.

[21] Bay, H., Tuytelaars, T., "Van Gool", L.: SURF: Speeded Up Robust Features, Proceedings of the ninth European Conference on Computer Vision (ECCV), Graz, Austria, (May 2006), 404-417.

[22] Matas, J., Chum, O., Urban, M., and Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In The 13th British Machine Vision Conference (BMVC), Cardiff University, UK, (September 2002), 384-393.

[23] Hartley, R. I., Zisserman, A.: Multiple View Geometry in Computer Vision, Cambridge University Press, ISBN: 0521623049 (2000)

[24] Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos, Proceedings of Int. Conference of Computer Vision (ICCV), Nice, France, (October 2003), 1470-1477.

[25] Csurka, G. and Dance, C. and Fan, L. and Willamowski, J. and Bray, C.: Visual categorization with bags of keypoints, Workshop on Statistical Learning in Computer Vision, ECCV (Prague 2004), 1-22.

[26] David G. Lowe.: Distinctive Image Features from Scale-Invariant Keypoints. Int. Journal of Computer Vision, 60(2):91110, 2004.

[27] Nowak, E. and Jurie, F. and Triggs, B.: Sampling Strategies for Bag-of-Features Image Classification, Proceedings of the European Conference on Computer Vision (ECCV), Graz, Austria, (May 2006) 490-503.

[28] Cristianini, N. and Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, March 2000