Obstacle Detection and Alignment Using an Stereo Camera Pair

David Aldavert, Arnau Ramisa, Ricardo Toledo and Ramon López de Mántaras

***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 descriptors of salient 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 bag of features approach to identify putative corresponding maps followed by a Gauss-Newton 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 algorithms performs better in structured environments and IDC in unstructured environments, Gutmann combines bot methods in [3]. The IDC is a variant of the ICP (Iterative Closest Points) 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 because for each primitive of a measurement set its corresponding primitive of the other measurement set must be found. Therefore, other methods tried 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 the correlation measure. This method is designed to work in very structured environments, so, when applied in unstructured environments the expected results should 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, a normal distribution function is assigned to each cell, 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 used range information which is not discriminative enough to directly find correct correspondences between primitives, so that, such methods iteratively searched the corresponding 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 low 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 positive correspondences rises rapidly and they cannot be filtered. 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* approach, signatures of obstacle maps are built using robust invariant features extracted from stereo images. Such map signatures are used as a fast measure to determine if two maps are likely to be related or not. Finally, from the obstacle map a dense probability distribution is built and it is used to iteratively determine robot motion using a Newton minimization algorithm. This approach shares the same idea underlying in the Normal Distributions Transform [9], but the probability distributions map has to be built taking into account that stereo points have an heteroscedastic (point dependent) and anisotropic error distribution. Besides, color image information is added to the probabilistic map in order

This work has been partially funded by TIN 2006-15308-C02-02 project grant of the Ministry of Education of Spain.

D. Aldavert and R. Toledo are with the Computer Vision Center (CVC), Dept. Ciències de la Computació, Universitat Autònoma de Barcelona (UAB), 08193, Bellaterra, Spain aldavert@cvc.uab.cat, ricard@cvc.uab.cat

A. Ramisa and R López de Mantaras are with the Artificial Intelligence Research Institute (IIIA-CSIC), Campus UAB, 08193, Bellaterra, Spain aramisa@csic.iiia.es, mantaras@iiia.csic.es

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

Our robot obtains environment information from a stereo head instead of using a typical range sensor such as a laser range finder. Stereo camera pairs give illumination, color and texture information in addition to the depth information. Such depth information cannot be directly obtained and it requires the use of a dense stereo algorithm [14]. However, obstacle maps obtained from a stereo head have more information than maps obtained from a typical depth range sensor. Using stereo data, image information can be directly added to the detected obstacles and the data is not restricted to a world plane. Besides, a signature that identifies the obstacle map is obtained by extracting robust invariant features from the images using a *bag of features* approach.

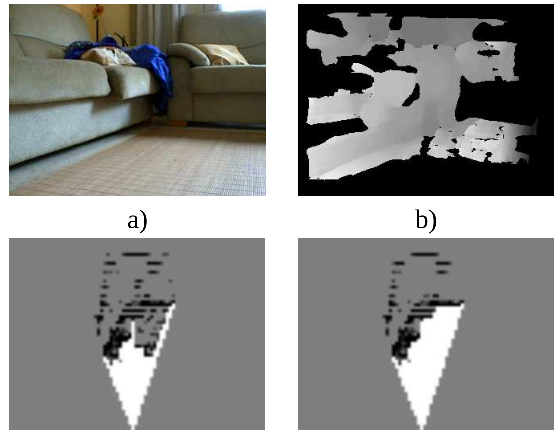
A. Obstacle maps

Obstacle maps are represented by a 2D histogram in the X-Z world plane where each cell represents the probability that an obstacle is present in a certain area. Obstacles are detected using an algorithm based on [15], which for a relatively small resolution can obtain a dense stereo map in real time. It is a correlation based algorithm that uses the SAD (Sum of Absolute Differences) function as similarity measure. In our approach, several expensive refinements of the method, such as the left-right consistency check, are removed in order to reduce the computational cost of the algorithm. To remove possible disparity map inconsistencies due to occlusions, the resulting disparity map is segmented using the watershed algorithm [16] and small disparity regions are removed from it.

Once the dense stereo map is obtained, image points are transformed from pixel coordinates to image plane coordinates, so 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 point pair in image plane coordinates:

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

where b is the baseline and f is the focal length of the camera. The resulting 3D world points that are within a height range, are reprojected to a 2D histogram in the X-Z world plane. Histogram cells that fail to have the minimum support to be consider an object and isolated cells are removed from the 2D histogram. 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



a) b) c) d)

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.

disparity maps have gaps in low textured regions, objects 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 pan & tilt unit servo motors are quite small (about $0.5°$ degrees) compared to the obstacle map resolution and to the stereo depth estimation error, therefore, the translation and rotation of the stereo head cameras at each scan step are 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.

B. Maps signature

In order to filter most of the unrelated scans a 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 technique proposed by Nistér and Stewénius in [22]. This technique builds a codebook of local descriptors using hierarchical *k*-means on a database of local descriptors from training images. 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 region detectors: Harris Affine, Hessian Affine, Harris Laplace, Hessian Laplace [10] or the SURF detector [**?**]. The signature consists of a normalized histogram of the labeled descriptors of a scan. This method was selected because of its simplicity and because it specifically addresses the issue of scalability to large databases, of great importance when the number of mapped locations increases.

Once a new scan is acquired, its signature histogram is constructed and compared to the ones stored in the database. Only the first k nearest neighbors are considered for local alignment. This measure notably reduces the search time as only the most relevant scans of the database are considered for a Newton alignment step described in Section II-A.

C. Color obstacle maps

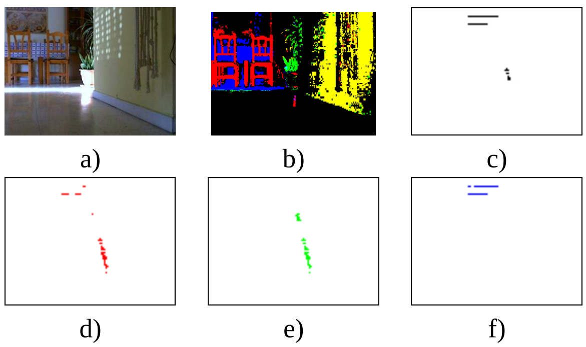


Fig. 2. Original image (a) is segmented obtaining (b) and the different layers of the local map are built: Locals map with greyish features (c), reddish features (d), greenish features (e) and bluish features (f).

Instead of using only depth information to build local environment maps, they are enhanced by adding color information to each 2D histogram cell. Basically, the 2D histogram is divided into 4 layers that contain greyish, reddish, greenish and bluish color information. Images are transformed to the *HSV* color space (Hue-Saturation-Value) and the *Hue* channel is used to determine to which color layer the pixel must be assigned while pixels with a small difference between the max and min *RGB* values or with a high Value are assigned to the grayish channel. In Fig. 2, a segmentation example is shown. Black pixels correspond to regions assigned to the greyish layer and color pixels correspond to the other three color layers. Pixels whose color is a combination of two primary colors, are assigned to their primary components, e.g. a yellow pixels is assigned to both red and green layers. Once the histogram is built, if a histogram cell of a color layer has not enough support, the cell value is set to zero and its contribution is added to greyish layer.

III. Map Alignment

In this section we present the method used to align different local maps. First of all, the local map signatures are used to determine if the local map is likely to be aligned. Then, a PDF distribution is made from obstacle maps and is used to align two local maps using a Newton approach.

A. Signature comparison

The signatures obtained in section II-B are used to sort all putative local maps using the Euclidean distance between histograms. As it is difficult to establish a global or relative threshold to determine if two signatures correspond to related local maps or not, the k nearest neighbours are selected and then the alignment method is used in order to filter possible incorrect matches. The amount of nearest neighbours has been determined experimentally from our data set. Using the different detector and descriptor combinations it is shown that selecting only the tenth similar signatures we ensure that the 95% of the relations presents in our database are selected. Therefore, the signature is useful to reduce the number of alignments in 1/5.

B. Iterative map alignment

In indoor or flat urban environments, robot measurements taken from two places are related by a rigid 2D transformation. To align two local stereo maps, the following method is proposed:

1) Build an object probability distribution from one of the two local stereo maps.
2) Use the object coordinates of the other local stereo map to initialise the set of points S that will be registered to the first local stereo map. Initialise the motion parameters to zero or by an estimation obtained with the robot odometry.
3) Apply the parameters of the transformation to the set of points S.
4) From the values of the probability transformation of the set of transformed points a score value is obtained.
5) Estimate a new parameter value by optimising the score using a Gauss-Newton algorithm.
6) While the convergence criterion is not meet, go to 3.

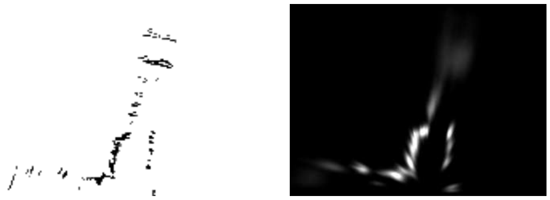
The alignment method used is similar to the methods used in computer vision for registration of image information obtained from different sensors [18] or aligning images related by an affine or projective transformation [19], [17]. The first step of the method consists in building the dense obstacle probability distribution. Although building the probability distribution is quite computationally expensive, it has to be built only once. The 2D histograms built in the previous section have several layers of information. However, for the sake of simplicity, the method will be explained first as if it had only a single layer. At each pixel where an object is found, we define Gaussian with mean μ_o equal to the location of the object and variance σ_o defined by the uncertainty of stereo points:

$$\sigma_o = RJ \left[\begin{array}{cc} \sigma_{l_x} & 0 \\ 0 & \sigma_{r_x} \end{array} \right] J^\top R^\top \ . \qquad (2)$$

where σ_{l_x} and σ_{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:

$$J = \left[\begin{array}{cc} \frac{-bx_r}{d^2} & \frac{bx_l}{d^2} \\ \frac{-bf}{d^2} & \frac{bf}{d^2} \end{array} \right] \ . \qquad (3)$$

where $d = x_l - x_r$ is the disparity between x_l and x_r expressed in image plane coordinates. The rotation matrix



a) b)

Fig. 3. a) Objects 2D histogram. b) Probability distribution build from the 2D histogram.

R is expressed as follows:

$$R = \left[\begin{array}{cc} cos\beta & -sin\beta \\ sin\beta & cos\beta \end{array} \right] \ . \tag{4}$$

where β is the pan unit rotation angle. The covariance matrix σ_o of each probability distribution cell can be calculated *a priori* because we know the stereo head orientation for each histogram cell. Figure 3 shows how a probability distribution (Fig. 3.b) is formed from a 2D histogram (Fig. 3.a).

Now, once the probability distribution is defined, a Gauss-Newton algorithm is used to find the transformation parameters that best align two probability distributions by minimising the following expression:

$$\sum \left(PD_2(x,p) - PD_1(x) \right)^2 \ . \tag{5}$$

where PD_1 and PD_2 represent the probability distributions of each local map. However, to minimise (5) the probability distribution PD_2 must be transformed using parameters p at each algorithm step. This operation is quite expensive. Therefore, instead of using all probability distribution values, only the set of points S_1 and S_2, which respectively correspond to the objects of PD_1 and PD_2, are used. As equation 5 is a non-linear equation, it is minimised using a Gauss-Newton approach similar to the Lucas-Kanade algorithm [19]. The Lucas-Kanade algorithm assumes that a current estimate of p is known an iteratively solves for increments of the parameters $\triangle p$. So that expression 5 is approached by:

$$\sum \left(PD_2(x,p + \triangle p) - PD_1(x) \right)^2 \ . \tag{6}$$

Then the parameters are updated using the following equation:

$$p = p + \triangle p \ . \tag{7}$$

These two are iterated until the estimate of p converges. The non-linear expression 6 is linearised by performing a first order Taylor expansion on $PD_2(x, p + \triangle p)$:

$$\sum \left(PD_2(x,p) + \bigtriangledown PD_2 \frac{\partial W}{\partial p} \triangle p - PD_1(x) \right)^2 \ . \tag{8}$$

where $\bigtriangledown PD_2$ is the gradient of PD_2 evaluated at $PD_2(x,p)$:

$$\bigtriangledown PD_2 = \left(\frac{\partial PD_2}{\partial x}, \frac{\partial PD_2}{\partial y} \right) \ . \tag{9}$$

and $\frac{\partial W}{\partial p}$ is the Jacobian of $\bigtriangledown PD_2$ with respect to parameters p. As robot motion in a flat environment 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 of $\bigtriangledown PD_2$ is:

$$\frac{\partial W}{\partial p} = \left[\begin{array}{ccc} -x_i sin\alpha - y_i cos\alpha & 1 & 0 \\ x_i cos\alpha - y_i sin\alpha & 0 & 1 \end{array} \right] \ . \tag{10}$$

Minimizing the expression 9 is a linear squares problem and has a closed form solution which can be derived as follows. The partial derivative of expression 9 with respect $\triangle p$ is:

$$2 \sum_x \left[\bigtriangledown Pd_2 \frac{\partial W}{\partial x} \right]^T \left[Pd_2(x,p) + \bigtriangledown Pd_2 \frac{\partial W}{\partial p} \triangle p - Pd_1(x) \right] \tag{11}$$

Setting equation 11 equal to zero and solving gives the closed form solution for the minimum of the expression 8 as:

$$\triangle p = H^{-1} \sum_x \left[\bigtriangledown Pd_2 \frac{\partial W}{\partial x} \right]^T \left[Pd_1(x) - Pd_2(x,p) \right] \tag{12}$$

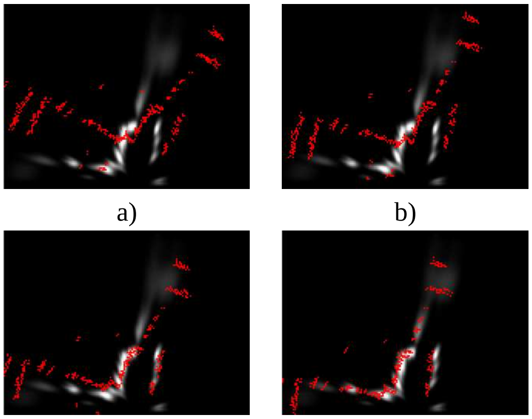
where H is a 3×3 Gaussian-approximation of the Hessian matrix:

$$H = \sum_x \left[\bigtriangledown Pd_2 \frac{\partial W}{\partial x} \right]^T \left[\bigtriangledown Pd_2 \frac{\partial W}{\partial x} \right] \tag{13}$$

Finally, the algorithm is iterated until a max number of iterations is reached or the update of the parameters $\| \triangle p < \varepsilon \|$. 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. Figure 4 depicts an example where the points set S_2 gradually converges to the means of the dense probability distribution Pd_1. To estimate motion parameters using all 2D color layers, the sums of Equations 13 and 12 must take into account the correspondence of Pd_1 and Pd_2 to the obstacle color.

IV. Results

In order to perform the experiments, data has been acquired by an autonomous mobile robot developed at our department. It is based on a Lynxmotion 4WD3 robot kit and has been designed to be as cheap as possible. The robot is controlled by an on-board VIA Mini-ITX EPIA PE computer with a VIA C3 1 GHz CPU and by an AVR ATmega128 microcontroller. The AVR controls all low-level sensors and actuators, such as sonars, infrared range sensors, a digital compass and an accelerometer. The Mini-ITX computer controls the AVR and processes the images obtained by the two Philips SPC900NC webcams, providing a high level programming interface. Indeed, we developed a driver in order to include a robotic software platform called Pyro [21] as the highest software abstraction level. To test the performance of our method, a database of 48 panoramas is



a) b) c) d)

Fig. 4. Local maps alignment after: a) first iteration, b) 10 iterations, c) 20 iterations and d) final solution after 42 iterations.

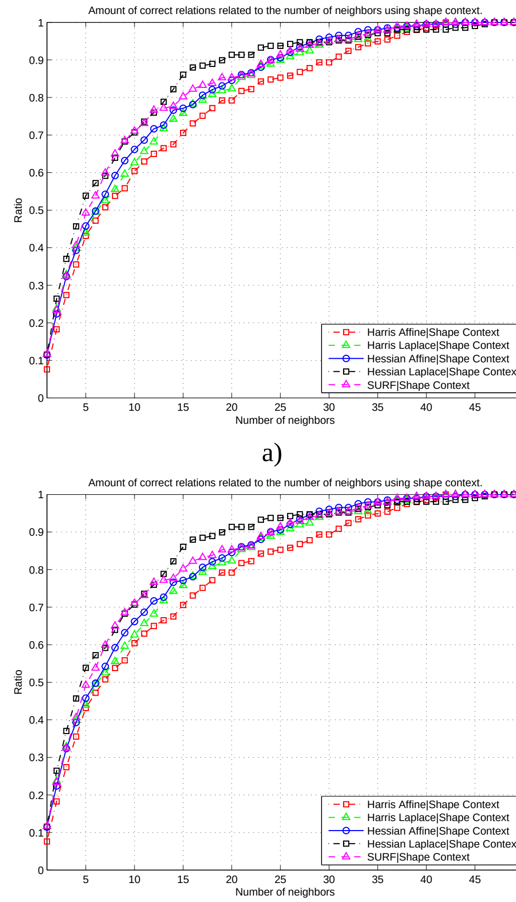
build in a typical indoor environment. The local maps are built from a pair of stereo images that have a resolution of 320×240 pixels and stereo measurements are stored in a 2D histogram that has 160 cell width per 120 cell height. Each cell side is 0.05 meters long, so that, the local map has a width of 8 meters and a height of 6 meters.

A. Map identification

As explained in chapter II-B we have used a *bag of features* approach to select the most similar scans of the set of local maps. The hierarchical dictionary that we have used have 625 words and a branch factor of 5. Figure 5.a) shows the average ratio of map relations (Y axis) that are included between the k nearest neighbors (X axis) using the shape context descriptor. Figure 5.b) shows the same results but using the SIFT descriptor. No significant differences are appreciated between both descriptors, therefore as the dimensionality of shape context is much lower than SIFT, it is a better choice as it needs less computational effort. Based on the obtained results we have chosen the Hessian-Laplace detector, which performs similar to Hessian-Affine and is significantly faster. Also we have set the k value to 15 which on average will include 87% of the correct relations.

B. Map alignment

We have selected 10 significant examples from the data set and we have applied an exhaustive set of rigid transformations to the data in order to test our alignment method. The set of transformations consists in rotations of the local map from $-90\circ$ to $90\circ$ with steps of $5\circ$ degrees and translations in steps of 10 centimeters from -1 meter to 1 meter. Additionally, different amounts of outliers where introduced to test the robustness of the method. In Fig. **??**.a) the ratio of correctly aligned maps applying rotational transformations combined with translational transformations limited to a maximum distance of 70 centimeters is shown. Figure 6.b) shows the ratio of correctly aligned maps applying the full set of translation transformations combined with rotational transformations up to 20 degrees.



a)

b)

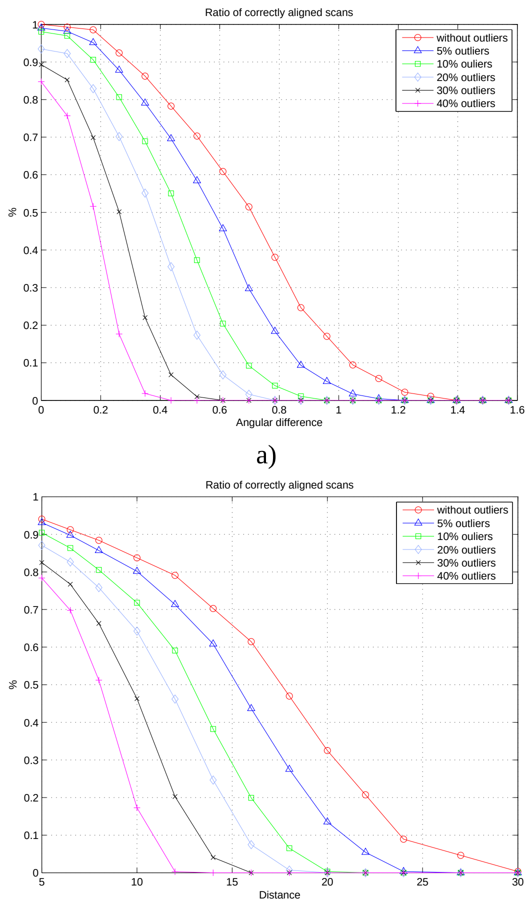
Fig. 5. Ratio of correct relations between in the k nearest neighbors using a) the shape context descriptor and b) the SIFT descriptor.

True positive	175
False positive	40
True negative	495
False negative	19

TABLE I

ALIGNMENT PERFORMANCE OBTAINED AFTER APPLYING THE NEWTON METHOD.

Finally, we have conducted a test to evaluate the performance of the whole system. The test consisted on finding the existing relations between the maps of our data set. We have applied the alignment method to the 15 maps selected in the *bag of features* step. Then, we set a threshold depending on the score obtained by equation 5 and the number of iterations spent by the Newton method. Taking into account the results of the previous exhaustive tests, we have selected the threshold that better separated the correctly aligned maps and failures. Table I shows the results of this test. As can be seen the method correctly filtered 92.52% of false relations between maps, while only losing 9.8% of correct relationships.



a)

b)

Fig. 6. Ratio of correctly aligned scans depending on angular (in a) and translational (in b) distance with different amounts of outliers.

V. Conclusion

In this paper, we have presented a method to build local maps from information acquired by a stereo head. The local map provides information about the 2D distribution of the 3D objects in the environment and it also stores color information. Next, we proposed a method that uses a Newton minimization algorithm to align these local maps. To avoid aligning maps with similar geometrical layout, we use a *bag of features* approach to also take into account image appearance information. Both methods avoid the expensive step of searching implicit feature correspondences. The obtained results show that *bag of features* effectively filters unrelated maps and, combined with the alignment method, nearly all relation of our data set are detected. This method performs well in environments with poor or repetitive textures, where methods based on feature matching tend to fail.

Future work includes improving the color segmentation algorithm to make it invariant to illumination changes. We also plan to test our method in larger data sets that include different types of environments. Finally, we want to implement a fast version of this schema to used it in a real time mapping experiment.

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

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

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

[18] 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.

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

[20] Gill, P. E., Murray, W. and Wright, M. H.: Practical Optimization, Academic Press, London and New York (1981).

[21] 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.

[22] 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.