Aligning methods for visual landmark-based maps

MÓNICA BALLESTA, ÓSCAR REINOSO, ARTURO GIL, MIGUEL JULIÁ, LUIS PAYÁ
Systems Engineering Department
Miguel Hernández University
Avda. de la Universidad, s/n 03202 Elche (Alicante)
SPAIN
{m.ballesta,o.reinoso,arturo.gil,mjulia,lpaya}@umh.es http://isa.umh.es/arvc/

Abstract: **When having a multi-robot system in which each robot contructs its own local map, it can be necessary to perform the fusion of these local maps into a global one. The Map Fusion problem involves the consecution of two different tasks: Map Alignment and Map Merging. The Map Alignment consists in computing the transformation, if existent, between the local maps. In this way, all the observations will be referenced to a common global frame. In the Map Merging stage, a global map is constructed from the local maps by integrating their information. This paper is focussed on the first step: Map Alignment. Particularly, a collection of aligning algorithms is evaluated in order to select the method that obtains the best results in terms of accuracy and stability. The experiments are performed in a multi-robot system, in which each robot constructs its own local map independently. These maps are visual landmark-based and the mapping algorithm used is FastSLAM.**

***Key-Words*: multi-robot system, map alignment, visual SLAM.**

1 INTRODUCTION

A real autonomous robot must have the ability to explore an environment and build a map of it. As a consequence the problem of *Simultaneous Localization and Mapping* (SLAM) has received great attention. Regarding to the sensors used, many approaches use range sensor such as SONAR [1] or LASER [2]. However, there is an increasing interest in using cameras as sensors in SLAM. This is due to the fact that cameras obtain a higher amount of information and are less expensive than lasers. Besides, $3D$ information can be provided when stereo cameras are used.

Most visual SLAM approaches are landmark-based and the maps represent the localization of a set of distinctive points from the environment with respect to a global reference frame [3, 4]. These distinctive points are landmarks accompanied generally by a visual descriptor, which encodes the visual appearance of these landmarks.

The process of SLAM can be performed by a single robot, but it will be more efficient if a team of robots cooperate in the solution of this task. This approach is denoted as multi-robot SLAM. In a multi-robot system, the robots explore simultaneously the environment and perform observations of it in such a way that a suitable map can be built collaboratively. Many solutions to the multi-robot SLAM problem have emerged [5, 6] so far. These solutions can be classified into two different groups. On the one hand, there are solutions in which the estimate of the trajectories and map building is performed jointly [2, 7, 8] so that a unique map is built. In this case, robots have a global notion of the environment and thus the exploration can be performed efficiently. However, the computational cost is higher and the initial pose of robots should be known, which is something that may not be possible in practice. On the other hand, there are solutions in which each robot builds its own local map independently [9]. The robots maintain their local maps until the fusion of the maps is required. According to [10], one of the main advantages of using independent local maps is that the data association problem is made easier. In this case, new observations should only be matched with a limited number of landmarks in the local maps. Moreover, the fusion of the local maps into a global one increases the robustness of the association process. Besides, it can be considered the case in which the initial poses of the robots are unknown.

Map fusion has received attention since the last few years. This is a challenging approach since many important issues must be faced. For instance, the moment in which the map fusion should be performed. Some authors propose a *rendezvous* strategy [6, 9, 11], in which the robots try to meet each other in a location and then merge their maps by means of the shared data. In [11], the meeting point is estimated with a particle filter approach and afterwards the robots arrange to meet in that location.

The fusion of local maps is performed in two main steps. The first one consists in computing the transformation, if existent, between the local maps. This is denoted as Map Alignment. Then, once the transformation between maps is known, the second step is to merge the maps (Map Merging). The problem is to decide how to integrate the information from each local map into a unique global map.

This paper is focussed on the first step, i.e., Map Alignment. In order to solve the transformation between local maps, some approaches try to compute the relative poses of the robots. As soon as the relative poses are known, the alignment of the maps is immediate. In that sense, the easiest case can be seen in [2], where the relative pose of the robots is considered known. A more challenging approach is presented in [6, 9]. As mentioned above, in these strategies a meeting point is arranged by the robots in order to share information of their maps and compute their relative poses. More difficult would be the approaches in which the robots determinate whether any alignment exists or not without the need of meeting, just by sharing the information of their maps. Some authors present feature-based techniques in order to align maps [12, 13, 14]. The basis of these techniques is to find matches between the landmarks of the local maps and then to obtain the transformation between them. Our paper focusses on the latter approach. Particularly, our aim is to analyze the performance of some methods that compute the transformation between a pair of $3D$ landmark-based maps. The experiments are performed with real data using the FastSLAM algorithm.

2 MAP BUILDING

The mobile robots used in these experiments are Pioneer P3-AT, provided with a LASER and a STH-MDCS2 stereo head from Videre Design. The mapping and navigation algorithm used is the FastSlam algorithm introduced in [15]. The robots construct progressively visual landmark-based maps of the environment by using exclusively the stereo camera and the odometry information. The visual landmarks, extracted from images of the environment, consist of Harris points detected by the Harris Corner detector [16] and described by U-SURF [17]. The combination of this detector and descriptor was selected as the most suitable feature extractor under our requirements in a previous work [18, 19]. In this case, two robots explore an indoor environment in the first floor of our building. The appearance of this environment is that of a typical office building in which the most common elements are doors, posters in the walls, windows, etc. The robots initially start from different positions and then continue navigating and building their maps independently, i.e., each robot has no knowledge about the position of the other robot and its observations. The local maps built by the robots consist of the $3D$ coordinates of the Harris points detected and their correspondent descriptor U-SURF. Each map is referred to the local reference frame of each robot, which is located in its initial starting position.

3 ALIGNING METHODS

In this section, several methods for map alignment are presented. These methods are suitable for aligning maps which are landmark-based. Particularly, in this case these maps consist of the $3D$ coordinates of significant points extracted from the environment (Harris points) and their correspondent descriptor (U-SURF). All these methods try to establish correspondences between the detected points in both maps by means of their descriptor similarity. Then, different techniques are used in order to compute the alignment of these maps from these correspondences.

3.1 RANSAC

This technique has been already applied to map alignment in [12]. The steps of this algorithm are described below.

1. First, a list of possible correspondences is obtained. Two points are considered as correspondences if the Euclidean distance between them is the minimum and it is below the threshold th_0. The coordinates $m = (x_i, y_i, z_i)$ are the landmarks of one of the maps, and $m' = (x'_i, y'_i, z'_i)$ their correspondences in the other map.

2. In a second step, two pairs of correspondences are selected at random from the previous list. These pairs should satisfy the following geometric constraint:
$$A^2 + C^2 \approx B^2 + D^2 \qquad (3.1)$$

where $A = (x'_i - x'_j)$, $B = (y'_i - y'_j)$, $C = (x_i - x_j)$ and $D = (y_i - y_j)$. The geometric constraint is satisfied if $|(A^2 + C^2) - (B^2 + D^2)| < th_1$. The two pairs of correspondences are used to compute the alignment parameters (t_x,t_y, θ) with the following equations:

$$t_x = x_i - x'_i \cos\theta - y'_i \sin\theta \tag{3.2}$$

$$t_y = y_i - y'_i \cos\theta + x'_i \sin\theta \tag{3.3}$$

$$\theta = \arctan \frac{BC - AD}{AC + BD} \tag{3.4}$$

3. The third step consists in looking for possible correspondences that support the computed transformation (t_x,t_y, θ), setting the threshold th_2. Finally, the second and third step are repeated M times. The final solution will be the one with the highest number of supports.

In our experiments, we have selected these values for the thresholds mentioned above: $th_0 = 2m$, $th_1 = 2m$ and $th_2 = 2m$. Furthermore, a parameter $min = 20$ establishes the minimum number of supports in order to validate a solution and $M = 70$ is the number of times that steps 2 and 3 are repeated. These are considered as internal parameters of the algorithm and their values have been experimentally selected .

3.2 SVD

One of the applications of the Singular Value Decomposition (SVD) is the registration of $3D$ point sets [20, 21]. This concept means obtaining a common reference frame by estimating the transformations between the data sets. In this paper the SVD has been applied for the computation of the alignment between two maps.

Given a list of possible correspondences, our aim is to minimize the following expression:

$$\|m'B - m\| \tag{3.5}$$

where m' and m are sets of correspondences between both maps. Next, B is the transformation matrix between both coordinate systems:

$$B = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & 0 & 1 \end{pmatrix} \tag{3.6}$$

B is computed as shown in Algorithm 1 of this section. In order to construct this list of correspondences (m and m'), the first step of the RANSAC algorithm (3.1) is performed. Then, the geometric constraint of Equation 3.1 is also evaluated. The internal parameters are equal to those specified in Section 3.1.

Data: m and m'
Result: Computation of matrix B
$[u, d, v] = svd(m')$;
$z = u' \cdot m$;
$sv = diag(d)$;
$z_1 = z(1 : n)$; *// n is the number of eigenvalues (not equal to 0) in sv.*
$w = z_1./sv$;
$B = v * w$;

Algorithm 1: Computation of the transformation matrix with SVD.

3.3 ICP

The Iterated Closed Point (ICP) technique was introduced in [22, 23] and applied to the task of point registration. The ICP algorithm iterates two steps:

1. Compute correspondences (m, m'). Given an initial estimate B_0, a set of correspondences (m, m') is computed so that it supports the initial parameters of B_0. B_0 is the transformation matrix between both maps indicated in expression 3.6.

2. Update transformation B. The previous set of correspondences is used to update the transformation B. The new B_{x+1} will minimize the expression: $\|m - m' \cdot B_{x+1}\|$, which is analogous to the expression 3.5. For this reason, we have solved this step with the SVD algorithm (Algorithm 1 in Section 3.2).

The algorithm stops when the set of correspondences does not change in the first step, and therefore B_{x+1} is equal to B in the second step.

This technique needs a quite good initial estimation of the transformation parameters so that it converges properly. For that reason, in order to obtain an appropriate initial estimate we perform the two first steps in RANSAC algorithm (3.1).

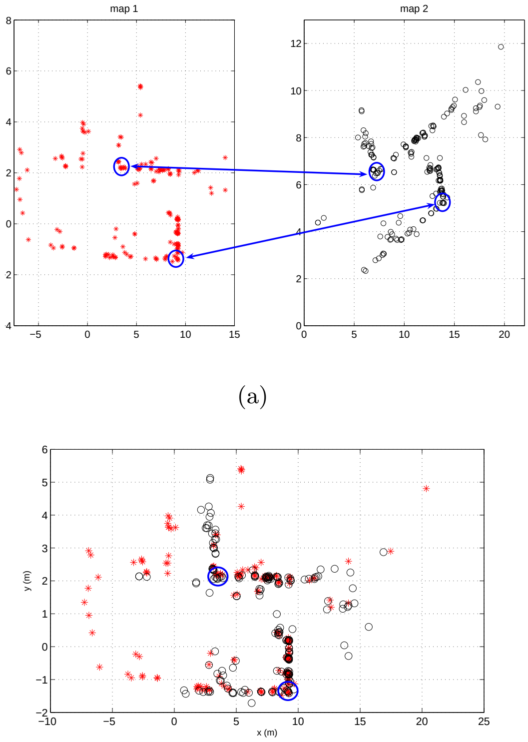
3.4 ImpICP

The improved ICP (ImpICP) method is a modification of the previous algorithm of Section 3.3, which has been performed *ad hoc*. In the previous subsection the importance of having a good initial estimate was explained. Besides, our method to compute this initial estimate was described. However, the accuracy of the results obtained is highly dependent on the goodness of the initial estimate. For that reason, in this new version of the ICP algorithm, we have increased the probability of obtaining a desirable result. Particularly, we obtain three different initial estimates instead of only one. This is performed by selecting three different pairs of correspondences each case in the second step of the RANSAC algorithm (Sec. 3.1), leading to three initial estimates. For each initial estimate, the algorithm runs as in Section 3.3. Finally, the solution selected is the transformation that is supported by the highest number of correspondences.

4 EXPERIMENTS

The purpose of these experiments is to compare the performance of the methods described in Section 3 as a tool for aligning visual landmark-based maps. The initial situation is that the robots begin the construction of their local maps independently and do not know their relative positions. In practice, we want to evaluate the behavior of the aligning methods at different steps of the mapping process. At the beginning, the maps built by each robot have sparse landmarks resulting in a extremely reduced number of correspondent landmarks between both maps. As a consequence, the alignment of these maps will surely fail. However, this situation improves as the size of both maps increases in such a way that there are more coincident landmarks between both maps. In this second situation, the map alignment is expected to be performed successfully.

In order to carry out our experiments, the most probable map of each robot is used to compute the transformation between both maps. This process is repeated in several iterations of the FastSlam algorithm. The most probable map is the map of the most probable particle of the filter in each particular moment. The aligning methods described above compute the alignment parameters t_x, t_y and θ. These parameters allow us to transform one of the maps into the reference frame of the other one, thus performing the map alignment stage. The accuracy of the aligning methods is evaluated by means of the error in the estimation of the alignment parameters. In our case, this error is computed as the Euclidean distance between the alignment parameters t_x, t_y and θ and the real relative position between both robots. This real relative position is what we call $Ground\ Truth$ and was obtained by calibrating the relative position between the robots when being both in their initial positions. These initial positions are the origin of the local reference frame of each robot. The error of the alignment parameters regarding to the $Ground\ Truth$ is measured in meters.



(a)

(b)

Figure 1: Map alignment ($2D$ view). Fig. 1(a) shows two local maps before the alignment. Fig. 1(b) shows the same maps after the alignment.

The FastSlam algorithm is performed in several iterations corresponding to the total number of movements performed by the robot. In the experiments k is an index that denotes the order of the iteration. In our case, the total number of iterations is $k = 1410$ and the final sizes of the maps are $map_1 = 263$ landmarks and $map_2 = 346$ landmarks. These maps have a dimension of 35×15 meters approximately. In Fig. 1(a) we can observe the local maps constructed by each robot and referred to its local frame. In this figure, map_1 is represented by stars and has 181 landmarks. On the other hand, map_2 is represented by circles and its size is of 187 landmarks. Then, Fig. 1(b) shows the result after aligning both maps. This example corresponds to the iteration $k = 810$.

Fig. 2 illustrates the comparison of the aligning methods we want to evaluate. For each method, the error values (y axis) $vs.$ the k-iteration of the algorithm (x axis) are represented. Logically, as the number of iterations

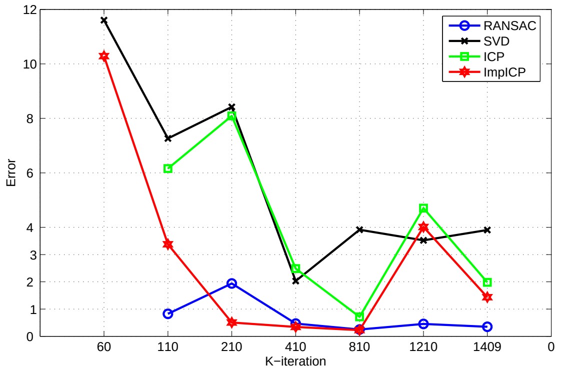


Figure 2: Evaluation of aligning methods.

increases, the size of the maps constructed will be higher and therefore it will be more probable to find a solution close to the *Ground Truth*. For this reason, it is expected to obtain small error values as the k-iteration increases. In Fig. 2 we can observe that the worst results are obtained with SVD. For instance, SVD has a error of $4m$ with $k - iteration = 1409$, i.e., at the end of the FastSlam algorithm. Next, ICP obtains similar results. However, it achieves better results in some cases. For example, with $k-iteration = 810$ the error is lower than $1m$. Then, the ImpICP algorithm outperforms these previous methods, since it achieves really small error values. Nevertheless, RANSAC is the method that obtains better results. Despite the fact that it gives no solution with $k-iteration = 60$ (probably because the maps are still too sparse in this iteration), the algorithm obtains the smallest error values. In fact, from $k - iteration = 410$ on the error is no higher than $0.5m$.

Finally, Fig. 3 focusses on the RANSAC algorithm results. Fig. 3(a) shows the number of correspondences that support the estimate of the alignment parameters obtained, i.e., the number of supports. The number of supports increases with the $k - interation$ values as can be observed in Fig. 3(a). On the other hand, Fig. 3(b) shows the decomposition of the error in its three components, i.e., the error in the estimate of each one of the alignment parameters t_x, t_y and θ. Fig. 3(b) leads to the deduction that the estimate of the t_y parameter is the most critical.

5 CONCLUSION

The main purpose of this paper was to evaluate and select a method for aligning visual landmark-based maps. In order to perform these experiments we have used real data collected by the robots in our building. The mapping process has been carried out with the FastSLAM algorithm. As a result, the RANSAC algorithm has proved to be the most suitable tool in order to align this kind of maps. The results presented by this algorithm have shown small error values and a stable behavior along different number of landmarks in the maps.

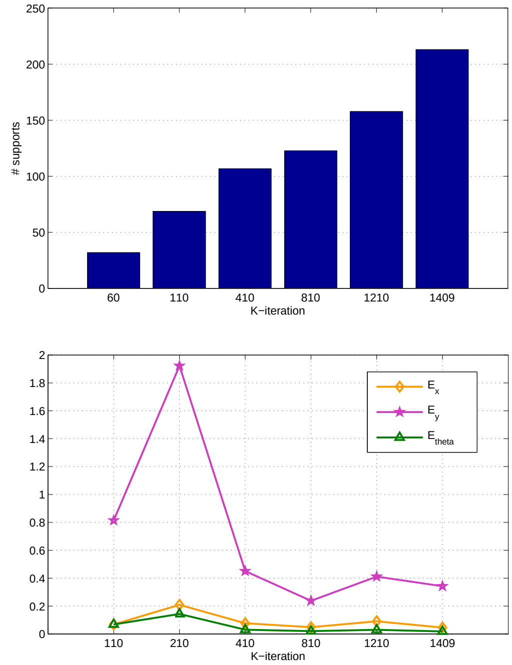


Figure 3: Results obtained with RANSAC algorithm. Fig. 3(a) shows the number of supports obtained. Fig. 3(b) shows error in each component of the alignment parameters.

As future work, our aim is to study the next stage in map fusion, which is *Map merging*. This is a quite challenging problem since maps built by different robots should be merged into a single one.

ACKNOWLEDGMENT

This work has been supported by the Spanish Government under project 'Sistemas de Percepción Visual Móvil y Cooperativo como Soporte para la Realización de Tareas con Redes de Robots' CICYT DPI2007-61107 and by the Generalitat Valenciana under grant BFPI/2007/096.

References

[1] O. Wijk and H. I. Christensen, "Localization and navigation of a mobile robot using natural point landmarkd extracted from sonar data," in *Robotics and Autonomous Systems, 1(31), pp. 3142,* 2000.

[2] S. Thrun, “A probabilistic online mapping algorithm for teams of mobile robots,” in *Int. Journal of Robotics Research, 20(5), pp. 335363*, 2001.

[3] J. Little, S. Se, and D. Lowe, “Global localization using distinctive visual features,” in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2002.

[4] A. Gil, O. Reinoso, W. Burgard, C. Stachniss, and O. Martínez Mozos, “Improving data association in rao-blackwellized visual SLAM,” in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2006.

[5] A. Howard, “Multi-robot simultaneous localization and mapping using particle filters,” in *The International Journal of Robotics Research, Vol. 25, No. 12, 1243-1256*, 2006.

[6] K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart, “Map merging for distributed robot navigation,” in *Proc. of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[7] J. W. Fenwick, P. N. Newman, and J. J. Leornard, “Cooperative concurrent mapping and localization,” in *Proc. of the 2002 IEEE International Conference on Intelligent Robotics and Automation, pp.1810-1817*, 2002.

[8] A. Gil, O. Reinoso, L. Payá, and M. Ballesta, “Influencia de los parmetros de un filtro de partculas en la solucin al problema de SLAM,” in *IEEE Latin America*, 2007.

[9] X. S. Zhou and S. I. Roumeliotis, “Multi-robot slam with unknown initial corrspondence: The robot rendezvous case,” in *Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, pp. 1785-1792*, 2006.

[10] S. Williams, “Phd dissertation: Efficient solutions to autonomous mapping and navigation problems.” Australian Center for Field Robotics, University of Sidney, 2001.

[11] D. Fox, “Distributed multi-robot exploration and mapping,” in *Proc. of the 2nd Canadian conference on Computer and Robot Vision*, 2005.

[12] S. Se, D. Lowe, and J. Little, “Vision-based global localization and mapping for mobile robots,” in *IEEE Transactions on robotics, vol.21, no.3*, 2005.

[13] S. Thrun and Y. Liu, “Simultaneous localization and mapping with sparse extended information filters,” in *The International Journal of Robotics Research.; 23: 693-716*, 2004.

[14] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai, “A practical, decision-theoretic approach to multi-robot mapping and exploration,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 3232–3238.*, 2003.

[15] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “Fastslam: A factored solution to simultaneous localization and mapping,” in *Proc. of the National Conference on Artificial Intelligence (AAAI), pp. 593598. Edmonton, Canada*, 2002.

[16] C. G. Harris and M. Stephens, “A combined corner and edge detector,” in *Alvey Vision Conference*, 1998.

[17] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded-up robust features,” in *Proc. of the 9th European Conference on Computer Vision*, 2006.

[18] O. Martínez Mozos, A. Gil, M. Ballesta, and O. Reinoso, “Interest point detectors for visual slam,” in *Proc. of the XII Conference of the Spanish Association for Artificial Intelligence (CAEPIA), Salamanca, Spain*, 2007.

[19] M. Ballesta, A. Gil, O. Martínez Mozos, and O. Reinoso, “Local descriptors for visual SLAM,” in *Workshop on Robotics and Mathematics (ROBOMAT07), Portugal*, 2007.

[20] K. Arun, T. Huang, and S. Blostein, “Least square fitting of two 3d sets,” in *IEEE Transactions on Patterns Analysis and Machine Intelligence. vol. PAMI-9 no. 5, pp.698-700*, 1987.

[21] J. Rieger, “On the classification of views of piecewise smooth objects,” in *Image and Vision Computing, vol. 5, no. 2, pp. 91-97*, 1987.

[22] P. Besl and N. McKay, “A method for registration of 3-d shapes,” in *IEEE Transactions on Patterns Analysis and Machine Intelligence. vol. PAMI-14 no. 2, pp. 239-256*, 1992.

[23] Z. Zhang, “On local matching of free-form curves,” in *Proc. of BMVC, pp. 347-356*, 1992.