ETFA 2008

13th IEEE International Conference on

Emerging Technologies and Factory Automation

15 - 18 September 2008, Hamburg, Germany

www.etfa2008.org

HELMUT SCHMIDT UNIVERSITÄT
Universität der Bundeswehr Hamburg

© 2008 IEEE

IEEE Catalog Number: 08TH8968C

ISBN: 1-4244-1506-3

Library of Congress: 2007904705

Help and Support Informations in file: support.txt on the CD!

Analysis of Map Alignment Techniques in visual SLAM systems

Mónica Ballesta, Óscar Reinoso, Arturo Gil, Miguel Juliá and Luis Payá
Miguel Hernandez University
Dept. of Industrial Systems Engineering (Edif. Torreblanca)
Avda. Universidad s/n 03202-Elche (Spain)
(m.ballesta, o.reinoso, arturo.gil, mjulia, lpaya)@umh.es

Abstract

In a multi-robot system, in which each of the robots constructs its own local map, it is necessary to perform the fusion of these maps into a global one. This task is normally performed in two different steps: by aligning the maps and then merging the data. This paper focusses on the first step: Map Alignment, which consists in obtaining the transformation between the local maps built independently. In this way, these local maps will have a common reference frame. In this paper, a collection of algorithms for solving the map alignment are analyzed under different conditions of noise in the data and intersection between local maps. This study is performed in a visual SLAM context, in which the robots construct landmark-based maps. The landmarks consist in $3D$ points captured from the environment and characterized by a visual descriptor.

1. Introduction

Building maps is one of the fundamental requirements for autonomous robots, since they can perform higher level tasks by using a map of the environment. Odometry measures can be suitable for short distances. However, the error is accumulated progressively, thus leading to high errors in odometry and in the estimate of the robot's pose. Also, other techniques such as GPS can be employed, but this approach is not suitable for indoor environments. It is evident that a real autonomous robot must have the ability to explore an environment and build a map of it. That concept is known as *Simultaneous Localization and Map Building* (SLAM), which has received great attention. The purpose of the SLAM problem is to build a map of the environment while, simultaneously, the robot is localized in it.

The solutions for the SLAM problem can be classified according to the following aspects:

- The nature of the map.
- The sensors employed to observe the environment.
- The SLAM algorithm used to solve the problem.

Regarding to the first aspect, two fundamental kinds of maps have been built so far. On the one hand, *occupancy maps* represent the environment in a 2D plane divided in cells [7]. Each cell represents the probability of the space to be occupied. However, the environment is usually three-dimensional, so *occupancy maps* might not be accurate enough. On the other hand, *landmark-based maps* represent the localization of a set of points from the environment with regard to a global reference frame. The main advantage of *landmark-based maps* is the compactness of their representation. By contrast, this kind of maps requires the existence of structures or objects that are distinguishable enough.

In relation to the second aspect, typical approaches use range sensors such as SONAR [31] or LASER in 2D [17, 27] and 3D [29]. Nevertheless, there is an increasing interest in using cameras as sensors. The main reason for this interest stems from the fact that cameras obtain a higher amount of information and are less expensive than lasers. Moreover, they can provide 3D information when stereo cameras are used. This approach is denoted as visual SLAM [30]. Most visual SLAM approaches are feature-based. In this case, the map is formed by a set of visual landmarks that are extracted from images of the environment. The landmarks define the $3D$ position of a set of distinctive points which have a characteristic visual appearance. Mainly, two steps must be distinguished in the selection of visual landmarks. The first step involves the detection of interest points in the environment. The detection should be as stable as possible, since the points of the environment are observed from different viewpoints. Then, at a second step the interest points are described by a feature vector which is computed using local image information. This descriptor is used in the data association problem, i.e., when the robot has to decide whether the current observation corresponds to one of the landmarks in the map or to a new one. Different detectors and descriptors have been used for mapping and localization using monocular or stereo vision, such as SIFT [18, 10, 30], the Harris corner detector [5, 13], Harris-Laplace [14] or SURF [22]. In a prior work, we performed a comparative study in order to find the most suitable combination detector-descriptor in the visual SLAM context [19, 2].

1-4244-1506-3/08/$25.00 ©2008 IEEE

Finally, according to the algorithm that solves the SLAM problem, we describe the main existent techniques. So far, most solutions are based on the following approaches:

- The *Extended Kalman Filter* (EFK).
- *Rao-Blackwellized* particle filters (denoted with the general term *FastSLAM*).

The *Extended Kalman Filter* was initially introduced as solution to the SLAM problem in [25], but the first real applications of this algorithm appeared in [17, 21]. The EKF suggests, as solution, the estimate of an augmented state vector including the robot's pose estimate and the position of landmarks in the map. It has been shown in [25] that as long as the robot is moving in an unknown environment while obtaining relative observation from the landmarks, the estimates of the localization of each landmarks are all correlated among them. This is due to the common error in the estimate of the robot's pose. The EKF assumes that the observation model and the movement model of the robot can be modeled as gaussian processes. In general, the EKF works well when having a robust data association and there is a sparse set of landmarks, which are dispersed in the environment.

A more recent and also successful approach is the FastSLAM algorithm, which was introduced in [20]. The most characteristic aspect of this algorithm is the use of a particle set which represents the uncertainty of the robot's pose whereas each particle has its own associated map. Logically, the SLAM problem is a sum of two fundamental aspects: the estimate of the robot's pose and the estimate of the map. Although these two aspects are intrinsically related, they can be considered separately. For instance, if we knew the robot's path, then the estimation of the map would be trivial. Analogously, if the map of the environment is known, it would be easy to implement an algorithm in order to find the robot's pose. This is the main idea of the FastSLAM algorithm, in which the SLAM problem is divided in a localization problem and in several individual estimations of the landmarks. The solution to the SLAM problem is performed by means of a sampling and particle generation process, in which the particles whose current observations do not fit with their associated map, are eliminated. The FastSLAM algorithm has shown to be robust to false data association and it is able to represent models of non-linear movements in a reliable way.

The process of SLAM can be performed by a single robot, but it will be more efficient if there is a team of robots that cooperate in the solution of this task. This approach is denoted as multi-robot SLAM. This is a more challenging approach, since many key aspects should be solved. For instance, it is necessary to define an efficient exploration strategy as well as the way of estimating the map. In the last case, there are two possible solutions. The first one is to maintain a unique global map. However, it can be too expensive computationally. On the contrary, a second possibility would be that each of the robots maintains its own local map, until the fusion of the maps is required.

Map Fusion has received attention since relatively few years. When a map fusion problem is faced, many questions arise. The first one is the moment in which the maps should be merged. Some authors propose a *rendezvous* strategy [16, 9, 34], in which the robots try to meet each other in a location and then merge their maps by means of the shared data. In [9], the meeting point is estimated with a particle filter approach. Another aspect is to determine the transformation, if existent, between the local maps. This is denoted as *Map Alignment*. In many approaches the transformation between maps is performed with the matching of landmarks [24, 15]. In [6] and [4] the overlap between maps is supposed to exist. With relation to the previous strategies based on the *rendezvous* case, it is clear that the alignment between maps is possible and immediate if the robots succeed in finding each other. This is due to the fact that in this moment, the robots are sharing the same space in the map. More difficult would be the approach in which the robots determinate whether any alignment exists or not without the need of meeting, just by sharing the information of their maps. The final stage of the Map Fusion is merging the local maps. The problem is to decide how to integrate the information from each one of the local maps into a unique global map.

In this paper, we focuss on the *Map Alignment* stage, which allow us to obtain a global reference frame to all the landmarks from the independent local maps. In particular, our aim is to analyze the performance of some methods applied to compute the transformation between two $3D$ landmark-based maps belonging to two different robots. Visual landmarks have been used in many SLAM performances [10, 24]. The methods evaluated here are suitable for the nature or our maps (landmark-based). We have selected the RANSAC algorithm, the Singular Value Decomposition (SVD), the Iterated Closest Point (ICP) and an improved version of the last method that we have denoted as ImpICP. Only the RANSAC algorithm has been already applied to *Map Alignment* so far [24]. The rest of methods have never been applied to this purpose. Particularly, the goodness of these methods is evaluated regarding the number of overlapping points between both maps.

The paper is structured as follows. Section 2 presents a comparison between the common map *vs.* individual local maps strategies. Next, in Section 3, the methods selected to be analyzed in this paper are presented. Then, Section 4 describes the experiments performed and the results obtained. Finally, in Section 5 the conclusions of the study are presented.

2. Multi-robot Visual SLAM

Map building can be managed more efficiently if a team of robots performs this task instead of a single robot. In the last case, the robot has to cover the whole space around it in order to build the map. By contrast, if the same robot collaborates with other robots, the space will be divided and the distances traversed by each robot will be reduced. Hence the map will be finished in less time and the odometry errors will be smaller. For that reason, it is evident that a team of collaborative robots can perform the same task in a more efficient way.

In a multi-robot system, the robots explore simultaneously the environment and perform observations of the environment so that a suitable map can be built collaboratively. So far, many solutions to the multi-robot SLAM problem have emerged [12, 16]. These solutions can be classified into two different categories:

- Solutions in which the estimate of the trajectories and map building is performed jointly.
- Solutions in which each robot builds its own local map.

In the first case, there is an unique map which is built simultaneously with the observations of all the robots. Some examples of this approach can be seen in [8, 27]. In [8] an EKF estimates an state vector containing the poses of all robots together with a set of two-dimensional landmarks. A common map is built by means of the EKF equations. Thrun [27] also proposes a solution with a common map, in which a particle filter associated to each robot is maintained. The advantages of sharing a global map are the following. Firstly, robots have a global notion of the unexplored frontiers and thus the cooperative exploration can be improved. Besides, in a feature-based SLAM, a landmark can be updated by different robots. For this reason, robots do not need to revisit an area of the map in order to close the loop and reduce their uncertainty, since they can be localized with regard to landmarks which were correctly estimated by other robots. However, the computational cost of maintaining a global map can be considerable and the initial pose of the robots should be known, which is something that may not be possible in practice.

In the second case, the robots initially build their own individual maps referenced to a local frame. Then, the robots can merge their maps into a common one. Many approaches belong to this second group, such as [26] and [34]. In [26] the robots agree to meet in some point of the map in order to obtain their relatives positions. Once the transformation between robots is known, the maps can be merged. This approach considers the case in which the initial poses of the robots are unknown. Regarding to approaches that use vision in a multiple-map approach, there are only sparse examples. One of them is the case of [24], where a vision-based map building approach is performed. In this case, 3D maps are built using scale-invariant visual landmarks. Furthermore, a submap alignment algorithm for global localization is presented, which can be applied to multiple 3D submaps alignment. In [11] the robots use visual appearance in order to detect intersections between their local maps. Each of the multiple robots collects a sequence of images while exploring. Then, a similarity matrix is constructed with the similarity measures between images captured by each robot.

One of the main advantages of using independent local maps, as explained in [32], is that the data association problem is improved. Firstly, new observations should only be matched with a limited number of landmarks in the local submaps. Next, when the local submaps are fused into a global map a more robust association can be performed between the local submaps. Besides, the computational cost of maintaining those maps is lower. Nevertheless, one of the drawbacks of this approach is dealing with the uncertainty of the local maps built by different robots when merging them.

This paper focusses on the study of this last approach, i.e., a team of robots, in which each robot builds its own map. These submaps should then be aligned and merged. This aim of this paper is to study the alignment stage under different noise and intersection conditions.

3. Map alignment

In a multi-robot system, in which each robot constructs its own local submap, it is necessary to perform a later task, which consists in the fusion of those local submaps into a global one.

The fusion of local maps is performed in two main steps. The first one consists in computing the alignment, if existent, between the local maps. Then, once the transformation between maps is known, the second step is to merge the maps.

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 [27], where the relative pose of the robots is considered known. A more challenging approach is presented in [16] and in [34]. In these cases the robots, being in communication range, agree to meet in some point. If the meeting succeed, then the robots share information in order to compute their relative poses. Otherwise, they continue to build their local maps. In [9], a particle filter is used to estimate a possible localization of the other robots, thus arranging a meeting in that location. Other approaches present feature-based techniques in order to align maps [24, 28]. 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 last approach. As a consequence, the following feature-based techniques have been analyzed.

3.1. RANSAC

This technique has been already applied to map alignment in [24]. The algorithm is performed as follows.

1. The first step is to compute a list of possible correspondences. Each pair of correspondences has the

minimum Euclidean distance in their descriptors and it is below a 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:

$$A^2 + C^2 \approx B^2 + D^2 \qquad (1)$$

where $A = (x'_i - x'_j)$, $B = (y'_i - y'_j)$, $C = (x_i - x_j)$ and $D = (y_i - y_j)$. We have set a threshold th_1 so that $|(A^2+C^2)-(B^2+D^2)| < th_1$. The two pairs of correspondences are used to compute the alignment parameters (t_x, t_y, θ) with the following equations:

$$t_x = x_i - x'_i \cos\theta - y'_i \sin\theta \qquad (2)$$

$$t_y = y_i - y'_i \cos\theta + x'_i \sin\theta \qquad (3)$$

$$\theta = \arctan\frac{BC - AD}{AC + BD} \qquad (4)$$

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 that one with the highest number of supports.

In this algorithm, we have defined three different thresholds: $th_0 = 2$ for the selection of initial correspondences, $th_1 = 2$ for the geometric constrain of Equation (1) and $th_2 = 2$ for selecting supports. 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 [1, 23]. The *registration* consists in obtaining a common reference frame by estimating the transformations between the datasets. In our 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:

$$||m'B - m|| \qquad (5)$$

where m' are the landmarks of one of the maps and m their correspondences in the other map. On the other hand, B is the transformation matrix between both coordinate systems. B is computed as shown in Algorithm 1 of this section. In order to construct this list of correspondences (m and m'), we impose two different constraints. The first one is tested by performing the first step of the RANSAC algorithm (3.1). Then, the geometric constraint of Equation 1 is also evaluated.

```
Data: $m$ and $m'$
Result: Computation of matrix $B$
$[u, d, v] = svd(m')$;
$z = u'm$;
$sv = diag(d)$; .
$z_1 = z(1:n)$; //$n$ 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 [3, 33] 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 B_0. B_0 is the transformation matrix between both maps.

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 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 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 a highest number of correspondences (supports).

4. Experimental study

The experiments have been performed using two $3D$ feature-based maps. We have simulated the coordinates of the landmarks, as can be appreciated in Fig. 1. In this way the alignment is evaluated independently of the uncertainty in the point detection. Then, we use real U-SURF descriptors computed from real images of our laboratory. This descriptor proved to perform properly under our requirements [2]. Fig. 1 shows a $2D$ view of the two maps (map_1 and map_2). map_1 is represented by 250 points (stars), whereas map_2 is represented by circles and its size is variable as it will be explained next. The circles are those points of map_2 which are not contained in map_1 (non overlapping points). There is a fixed number of this kind of points in map_2 (#non overlapping points = 88). Then, the pentagrams are points which belong to both map_1 and map_2 (overlapping points). The number of these points varies along the experimental performance. Furthermore, map_2 is rotated and translated from map_1 in the XY plane ($\theta = 0.35\ rads$, $t_x = 5\ m$ and $t_y = 10\ m$). Initially, points which are common to map_1 and map_2 have the same descriptor in both maps. However, a Gaussian noise has been added to map_2 so that the data are closer to reality. Therefore, map_2 has noise with σ_L in the localization of the points (estimate of coordinates) and noise with σ_D in the descriptors. The magnitude of σ_L and σ_D has been chosen as suitable values after performing some previous experiments. The number of overlapping points between map_1 and map_2 is varied progressively, thus the size of map_2 is different in each case. map_1 maintains always its original size. In this way, the goodness of the methods from Section 3 can be analyzed with respect to the coincidence rate between both maps. Given these two maps, the transformation between map_1 and map_2 is computed with each one of the methods described in Section 3. In each case, the results are evaluated according to the error obtained. This is the *Mean Quadratic Error* computed with the map_2 and a *ground truth*. map_2 is transformed again with the solution obtained and the *ground truth* is the map_2 before being initially transformed regarding map_1 .

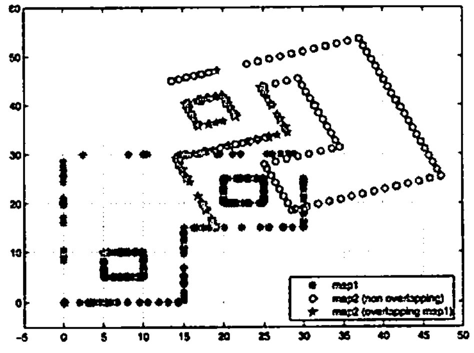

Figure 1. 2D view of map_1 and map_2 before the alignment.

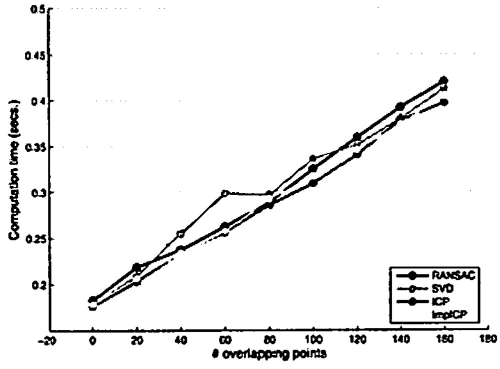

Figure 2. Time of computation vs no. of overlapping points.

In Figs. 3, 4, 5 and 6, the results are presented with σ_L and σ_D equal to 0.20, whereas Figs. 7, 8, 9 and 10 show the results with σ_L and σ_D equal to 0.50. In the X-axis the number of points that both maps have in common is represented. This value varies from 0 to 160. The first value shows the case in which the maps do not have any point in common. For each value, the experiment is repeated 10 times. Then, the *Mean Quadratic Error* is shown in the Y-axis (blue line). The blue points represents the individual values of the *Error* in each one of the 10 repetitions. In a similar way, the number of supports is also included in the graphics (red points). The number of supports is the number of correspondences that satisfy the transformation obtained as solution. The mean value of the supports is represented by a red line. In Figs. 3 and 7 an horizontal green line represents the minimum number of supports that must have a solution to be considered as acceptable (parameter m in Sec. 3.1). If the number of supports is bellow m, no solution will be considered and therefore no value of *Error* will be represented. Finally, all figures show the number of failures obtained in the 10 repetitions (see bars). Each failure represents the case in which the method does not converge to any solution or the solution does not satisfy the constraints (in the case of the RANSAC method), so that no alignment between the maps has been found.

Figs. 3 and 7 show the results obtained with the RANSAC algorithm of Sec. 3.1. In Fig. 3 the first solutions appear when the number of overlapping points is equal or higher than 60 points. In all of those cases, the *Mean Quadratic Error* is always below 2. Regard-

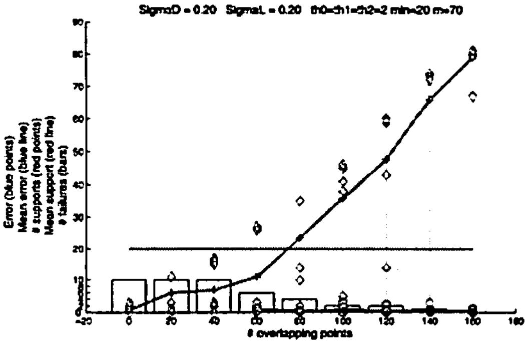

Figure 3. RANSAC algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.20$.

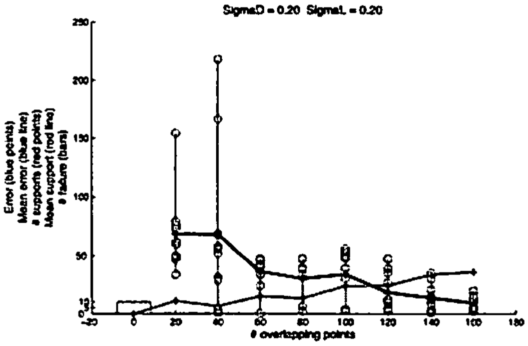

Figure 4. SVD algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.20$.

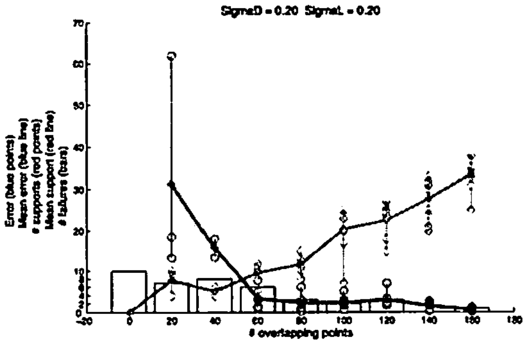

Figure 5. ICP algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.20$.

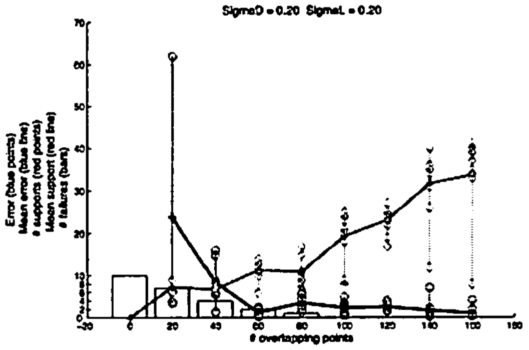

Figure 6. ImpICP algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.20$.

ing to the number of supports (red line), a logically ascendent tendency is appreciable. The maximum value of *Mean support* achieved is 80 from the total of 160 coincident points in that case. If the Gaussian noise grows, as occurs in Fig. 7, the results get worse. It can be seen that the number of supports decreases in all the experiments. Besides, it is noticeable that no solution is obtained until the number of overlapping points is 120. Before that, the number of failures is always 10 out of 10 repetitions (bars).

Figs. 4 and 8 present the results of the SVD algorithm of Sec. 3.2. In those cases, the error values are too high. For instance, in Fig. 4 the error value having 100 overlapping points is close to 30. At least, the error has a descendent tendency as the number of overlapping points increases. However, in Fig. 8 the error values are much more unstable. Regarding the number of supports, the tendency is quite constant in both graphics.

The behavior of the ICP algorithm of Sec. 3.3 is reflected in Figs. 5 and 9. In Fig. 5 the error value obtained are quite acceptable. It is noticeable that the error curve decreases sharply from the case of 20 to 60 overlapping points. Then, the curve continues descending but very slightly. This last part of the curve shows that the error values are around 2, which is a quite good result. However, the yellow bars show, in some cases, a small number of failures. Fig. 9 shows worse results.

Finally, in Figs. 6 and 10 the results of the improved version of the ICP algorithm (Sec. 3.4) are shown. In these cases, the results obtained are very similar to the previous ICP algorithm in terms of error and mean support values. However, it is noticeable that the stability of the algorithm is higher. Paying attention to the yellow bars in Fig. 6 it is shown that the algorithm always obtains a solution when the number of overlapping points is equal or higher than 100. In Fig. 10 the number of failures is also reduced in comparison with Fig. 9.

In general, the best results are obtained by the ImpICP and the RANSAC algorithm. It is true that the error values of RANSAC are slightly lower. However, the ImpICP algorithm is more stable in terms of having less number of failures. Comparing Figs. 3 and 6, it can be seen that RANSAC presents failures until the case in which the number of overlapping points is 140. By contrast, the

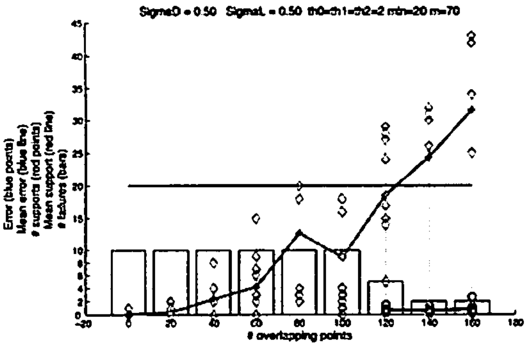

Figure 7. RANSAC algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.50$.

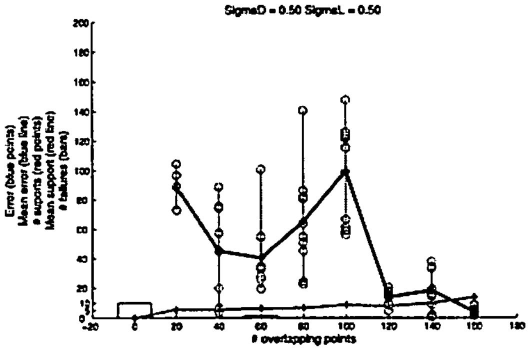

Figure 8. SVD algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.50$.

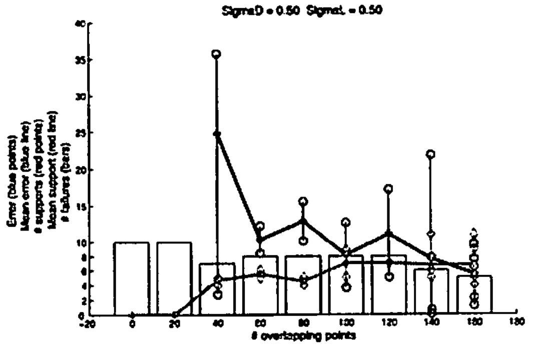

Figure 9. ICP algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.50$.

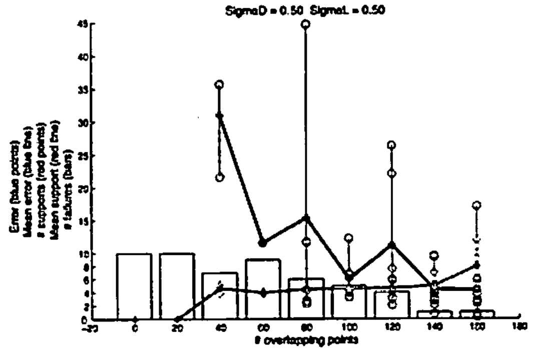

Figure 10. ImplCP algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.50$.

ImplCP algorithm do not obtain any failure when having more or equal than 100 overlapping points. In Figs. 7 and 10, it is shown that RANSAC obtains a much lower value of error than ImplCP. Nevertheless, it only achieves results until 120 overlapping points, whereas ImplCP obtains solutions with 40 overlapping points.

In addition to the experiments performed, the computation time of each algorithm has also been measured, as Fig. 2 shows. The curves shown have an ascendent tendency. This is due to the size of map_2, which is higher as the number of overlapping points increases. It is remarkable that the values of the computation time are very similar in all of the methods. For that reason, this criterion can not be used to select one of the methods.

5. Conclusions

Several algorithms for performing map alignment have been analyzed in this paper. The experiments have been performed with two 3D landmark-based maps constructed by two different robots in a multi-robot visual SLAM system. The methods have been evaluated under noisy conditions and different intersections between both maps. The ImplCP has shown to perform the best in our experiments, resulting in low error values and an acceptable stability, in terms of a small number of failures.

As future work, our aim is to analyze these methods using real input data captured in a SLAM process. We also like to study the next stage of the fusion of local maps, i.e., map merging.

6. Acknowledgements

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] 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.

[2] M. Ballesta, A. Gil, O. Martínez Mozos, and O. Reinoso. Local descriptors for visual slam. In *Workshop on Roboticas and Mathematics (ROBOMAT07), Coimbra, Portugal*, 2007.

[3] 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.

[4] S. Carpin, A. Birk, and V. Jucikas. On map merging. In *Robotics and Autonomous Systems, 53 (1), pp. 1-14, Elsevier Science*, 2005.

[5] A. J. Davison and D. W. Murray. Simultaneous localisation and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[6] G. Dedeoglu and G. Sukhatme. Landmark-based matching algorithm for cooperative mapping by autonomous robots. In *DARS, pp. 251-260*, 2000.

[7] A. Elfes. Using occupancy grids for mobile robot perception and navigation. In *IEEE Computer, pp. 46-57*, 1989.

[8] 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.

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

[10] A. Gil, O. Reinoso, W. Burgard, C. Stachniss, and O. Martínez Mozos. Improving data association in raoblackwellized visual SLAM. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2006.

[11] K. Ho and P. Newman. Multiple map intersection detection using visual appearance. In *3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems, Singapore*, 2005.

[12] 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.

[13] E. Hygounenc, I.-K. Jung, P. Soueres, and S. Lacroix. The autonomous blimp project of laas-cnrs: Achievements in flight control and terrain mapping. *International Journal of Robotics Research*, 23(4–5), 2004.

[14] P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman. A framework for vision based bearing only 3D SLAM. In *IEEE Int. Conf. on Robotics & Automation*, 2006.

[15] 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.

[16] 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.

[17] J. Leonard and H. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(4), 1991.

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

[19] 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.

[20] 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.

[21] P. Moutalier and R. Chatila. An experimental system for incremental environment modeling by an autonomous mobile robot. In *1st International Symposium on Experimental Robotics, Montreal*, 1989.

[22] A. C. Murillo, J. J. Guerrero, and C. Sagüés. Surf features for efficient robot localization with omnidirectional images. In *IEEE Int. Conf. on Robotics & Automation*, 2007.

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

[24] 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.

[25] R. Smith and P. Cheeseman. Estimating uncertain spatial realtionships in robotics. In *I. Cox y G. Wilfong (Eds.), Autonomous Robot Vehicles, pp. 167193. Springer Verlag*, 1990.

[26] B. Stewart, J. Ko, D. Fox, and K. Konolige. A hierarchical bayesian approach to mobile robot map structure estimation. In *Proc. of the Conference on Uncertainty in AI (UAI)*, 2003.

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

[28] 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.

[29] R. Triebel and W. Burgar. Improving simultaneous mapping and localization. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2005.

[30] J. Valls Miro, W. Zhou, and G. Dissanayake. Towards vision based navigation in large indoor environments. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2006.

[31] 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.

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

[33] Z. Zhang. On local matching of free-form curves. In *Proc. of BMVC, pp. 347-356*, 1992.

[34] 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.