

Multi-robot map alignment in visual SLAM

MONICA BALLESTA, ARTURO GIL, OSCAR REINOSO, MIGUEL JULIÁ, LUIS M. JIMÉNEZ

Miguel Hernández University

Department of Industrial Systems Engineering

Avda. Universidad sn, 03230 Elche

SPAIN

m.ballesta||arturo.gil||o.reinoso||mjulia||luis.jimenez@umh.es

Abstract: This paper focusses on the study of the Map Alignment problem in a multirobot SLAM context. Given a team of robots, we consider the situation in which each robot is building its own local map independently. These maps are landmark-based and three-dimensional. The local maps built by the different robots will have different reference systems. At some point, it may be interesting to express all maps in the same reference system. In that way, the maps can be fused into a unique global one. This is known as the map fusion problem and can be tackled by dividing the problem into two subproblems: map alignment and map merging. In this paper, we concentrate on the map alignment problem which consist on computing the transformation between the local maps so that we have one reference system. We therefore evaluate a set of aligning methods. These methods establish correspondences between each pair of maps and compute an initial estimate of the alignment. Finally, we apply the least squares minimization to obtain a more accurate solution. We also concentrate on the case in which we want to align more than two local maps. In this case, the alignment must be globally consistent. For the experiments, real data are used. Each robot extracts distinctive 3D points from the environment with a stereo camera. Then, the robots use these observations as landmarks to build their maps while simultaneously localize themselves. The SLAM problem is solved using the FastSLAM algorithm. The movements of the robots occur only in 2D plane, so although the maps are 3D landmark-based, the alignment takes place only in 2D.

Key-Words: Multirobot visual SLAM, Map alignment, visual landmarks.

1 Introduction

The capability of building a map of the environment while localizing itself in this map is essential for a robot to be autonomous. This problem is known as *Simultaneous Localization and Mapping* (SLAM) and has received a great attention in the last years [20, 9]. The basis of the SLAM problem is that the robot builds its map of the environment and simultaneously, localizes itself in this map. The existent SLAM algorithms share this idea but differ in the methods used to solve this problem. In this paper the SLAM algorithm is based on FastSLAM [24]. The main idea of the FastSLAM algorithm is to separate the two fundamental aspects of the SLAM problem: the estimate of the robot's pose and the estimate of the map. In this sense, the SLAM problem is divided into a localization problem and several individual estimates of the landmarks conditioned on the robot's path. This algorithm uses a particle set which represents the uncertainty of the robot's pose whereas each particle has its own associated map. The solution to the SLAM problem is performed by means of a sampling and particle generation process, in which the particles whose cur-

rent observations do not fit with their associated map are eliminated. The FastSLAM algorithm has proved to be robust to false data association and it is able to represent models of non-linear movements in a reliable way [17].

The problem of SLAM can be solved by a single robot, but this task will be performed more efficiently if there is a team of robots which construct cooperatively a map of the environment. This approach is denoted as multi-robot SLAM [17, 18]. In some approaches, the estimate of the trajectories and map building is performed jointly [12, 15]. In this case, the total area to be travelled is divided among the robots, therefore there is a global notion of the total space and the exploration can be more efficiently performed. However, the initial relative position of the robots should be known, which is something that may not be possible in practice. We concentrate on other solutions in which each robot builds its own map independently [34]. In these approaches, a set of local maps is maintained until the fusion of these maps is required. When using this approach, new observations should only be compared to a limited number of landmarks in the local map. This leads to smaller

errors due to the data association. Additionally, the construction of the local maps can be performed even if the relative positions of the robots are unknown. In our work, we focus on the last approach, i.e., robots build local maps independently.

Regarding the sensors used to extract information from the environment, many approaches use range sensors such as sonar [32, 19] or laser [29, 30]. However, there is an increasing interest in using cameras as sensors. These devices are less expensive than laser and obtain a higher amount of information from the environment [28]. Moreover, the 3D coordinates of the points from the scene can be obtained when using stereo cameras. This approach is denoted as visual SLAM [31, 10]. In most SLAM approaches using vision, the maps built are feature-based. The visual landmarks are extracted from the environment and define the 3D position of distinctive points that have a characteristic visual appearance. These points have a global reference system. In this case, each map is referred to the initial position of the robot.

The initial situation we consider is that there exists a team of robots which begin their navigation tasks independently and start at different positions. In this case, the map building is done independently, since the robots have initially no knowledge about other robots' positions or observations. This is one of the major advantages of this approach, since the initial positions of the robots do not need to be estimated.

At a specified moment, the fusion of the local maps into a global one may be required. We therefore study the map fusion problem [13]. We divide this problem into an alignment problem and a map merging problem. The map alignment computes the transformation between local maps which have different reference systems. Then the aligned data should be merged in order to obtain a global map. In this paper, we focus on the first stage. Particularly, we have performed a comparative evaluation of a set of aligning methods which are suitable for landmark-based maps. The selected aligning method should be robust to sparse correspondent landmarks between the maps and offer an accurate solution to the map alignment problem.

It is also remarkable that even though the maps have 3D landmarks, the result of the alignment is a 2D transformation. This is due to the fact that the motion of the robots is performed in a 2D plane. Besides, the solution given by this aligning methods is only an estimate of the alignment. Some post-processing is also recommendable in order to obtain a reliable solution. We therefore use the least squares minimization to obtain the real estimate of the alignment.

This paper is structured as follows. Section 2 presents some related work. Section 3 explains some

experimental details and the equipment used in this paper. Section 4 focusses on the alignment stage and Section 5 presents an evaluation of the aligning methods. Then, Section 7, focusses on the multi-robot case, in which the alignment is applied to more than two local maps. Finally, in Section 8, the main conclusions of this paper are stated.

2 Related work

As mentioned before, the map fusion is tackled in two different stages: map alignment and map merging. In the first case, the objective is to find the transformation between the different reference systems of the local maps. In this situation, most approaches try to find the relative position of the robots. In this sense, the easiest case can be seen in [29], where the relative position of the robots is supposed to be known. Nevertheless, more difficult approaches are [18] and [34]. In these cases, the robots try to establish a meeting point in order to measure their relative positions. In many approaches the transformation between maps is performed with the matching of landmarks [27]. In [11] and [6] 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 determine whether any alignment exists or not without the need of meeting, just by sharing the information of their maps. In this paper, we focus on finding these relative positions without the need of meeting at some point; on the contrary, the robots would share information of their maps in order to find the alignment between them. Our work is based on some previous approaches in which maps are aligned by means of landmark-based techniques. Particularly, we focus on the alignment of visual maps consisting of 3D landmarks.

3 Map building

We use a team of Pioneer-3AT robots, provided with a laser sensor and a STH-MDCS2 stereo head from Videre Design as can be seen in Figure 1.

As described in the previous section, robots build landmark-based maps which consist of the 3D position of distinctive points. 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



Figure 1: Pioneer-P3AT robot provided with laser and stereo vision.

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. In this sense, we have made an evaluation of several detectors and descriptors in order to select the most suitable one for our environment [22, 2, 14]. As a result of this prior study, the Harris Corner detector was selected to extract 3D points [16] from environment. This points are also characterized by the U-SURF descriptor [4], which describes a local neighbourhood around the point. In Figure 2, we can observe an example of the distinctive points detected in the environment. In these case we present an indoor scenario which corresponds to our Lab. Nevertheless, we have also test this detector and descriptor in outdoor environments [3].

The robots start at different positions and begin their navigation tasks independently. That is to say, they have no knowledge about other robot's positions or observations. Each local map is referred to a different reference system, which is located at the initial position of the robot. The robots build the local maps using exclusively information of the stereo camera and odometry. These maps are built by means of the Fast-SLAM algorithm, using a particle filter. Each particle provides a different estimate of the visual map. Concretely, in our work, the experiments have been carried out using 100 particles.

4 Map Alignment

The map alignment problem tackles the computation of the transformation between local maps so that they



Figure 2: Harris points detected in the scene. These points are described with the U-SURF descriptor and integrated as visual landmarks in the map.

can be expressed in the same reference system.

As described in Section 2, most approaches try to find the relative position of the robots [18, 34]. In these cases, the robots try to establish a meeting point in order to measure their relative positions.

In this paper, we focuss on finding these relative positions without the need of meeting at some point; on the contrary, the robots would share information of their maps in order to find the alignment between them. Our work is based on some previous approaches in which maps are aligned by means of landmark-based techniques [27]. Particular, we focuss on the alignment of visual maps consisting of 3D landmarks. We consider that the origin of the reference system is located in the starting point of the robot.

In the following, we present a set of aligning methods that are evaluated in order to find the most suitable one using this kind of maps (Section 5). In Section 6, the results of the comparative evaluation can be observed. These experiments are carried out using two local maps built by two different robots. Finally, in Section 7 we propose a method to solve the alignment problem when having more than two local maps.

5 Aligning Methods

The aligning methods presented in this section compute the transformation between two local maps. This transformation consists on three alignment parameters: translation in x and y (t_x and t_y) and rotation (θ). In order to do this, these methods establish first a list of correspondent landmarks between the maps. Although the landmarks are 3D (x,y,z) , the alignment is performed in 2D, because the robots move in a 2D plane (x,y) . Nevertheless, the third component of the landmarks (z) is also compared when establishing correspondences.

It is noticeable that these methods obtain only a first estimate of the aligning parameters. The set of correspondences and this estimate are used as the input of a *least squares minimization* that eliminates outliers and obtains the final solution [25].

In the following, we describe briefly the aligning methods that we compare.

5.1 RANSAC (Random Sample Consensus)

This technique has been already applied to map alignment in [27]. In the following, the steps of this iterative algorithm are described:

- In the first step, a list of possible correspondences is obtained. The matching between landmarks of both maps is done based on the Euclidean distance between their associated descriptors d_i . This distance should be the minimum and below a threshold $th_0 = 0.7$. As a result of this first step, we obtain a list of matches consisting of the landmarks of one of the maps and their correspondences in the other map, i.e., m and m' .
- In a second step, two pairs of correspondences ($[(x_i, y_i, z_i), (x'_i, y'_i, z'_i)]$ and $[(x_j, y_j, z_j), (x'_j, y'_j, z'_j)]$) are selected at random from the previous list. These pairs should satisfy the following geometric constraint[27]:

$$|(A^2 + B^2) - (C^2 + D^2)| < th_1 \quad (1)$$

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 the threshold $th_1 = 0.8 m$. The two pairs of correspondences are used to compute the alignment parameters (t_x, t_y, θ) with the following equations:

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

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

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

- The third step consists in looking for possible correspondences that support the solution obtained (t_x, t_y, θ) . Concretely, we transform the landmarks of the second map using the alignment obtained, so that it is referred to the same reference system as the first map. Then, for each landmark of the transformed map, we find the closest landmark of the first map in terms of the Euclidean distance between their positions. The pairing is done if this distance is the minimum and is below the threshold $th_2 = 0.4 m$. As a result, we will have a set of matches that support the solution of the alignment.
- Finally, steps c) and d) are repeated $M = 70$ times. The final solution will be that one with the highest number of supports.

In this algorithm, we have defined three different thresholds: $th_0 = 0.7$ for the selection of initial correspondences, $th_1 = 0.8 m$ for the geometric constraint of Equation (1) and $th_2 = 0.4 m$ 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 c) and d) are repeated. These are considered as internal parameters of the algorithm and their values have been experimentally selected.

5.2 SVD (Singular Value Decomposition)

One of the applications of the Singular Value Decomposition (SVD) is the registration of 3D point sets [1, 26, 23]. In this work the SVD has been applied for the computation of the alignment between two maps. We first compute a list of correspondences. In order to construct this list (m, m') , we impose two different constraints. The first one is tested by performing the first step of the RANSAC algorithm (Section 5.1). In addition, the geometric constraint of Equation 1 is evaluated. Given this list of possible correspondences, our aim is to minimize the following expression:

$$\|Tm - m'\| \quad (5)$$

where m are the landmarks of one of the maps and m' their correspondences in the other map. On the other hand, T is the transformation matrix between both coordinate systems (Equation 6). T is computed as shown in Algorithm 1 of this section.

$$T = \begin{pmatrix} \cos \theta & \sin \theta & 0 & t_x \\ -\sin \theta & \cos \theta & 0 & t_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6)$$

Algorithm 1 Computation of T , given m and m'

- 1: $[u, d, v] = \text{svd}(m')$
- 2: $z = u^T m$
- 3: $sv = \text{diag}(d)$
- 4: $z_1 = z(1 : n) \{n \text{ is the number of eigenvalues (not equal to 0) in } sv.\}$
- 5: $w = z_1 ./ sv$
- 6: $T = (v * w)^T$

5.3 ICP (Iterative Closest Point)

The Iterative Closest Point (ICP) technique was introduced in [5, 33] and applied to the task of point registration [21]. The ICP algorithm consists of two steps that are iterated.

- a) Compute correspondences (m, m') . Given an initial estimate T_0 , a set of correspondences (m, m') is computed, so that it supports the initial parameters T_0 . T_0 is the transformation matrix between both maps and is computed with Equations (2),(3) and (4).
- b) Update transformation T . The previous set of correspondences is used to update the transformation T . The new T_{x+1} will minimize the expression: $\|T_{x+1} \cdot m' - m\|$, which is analogous to the expression (5). For this reason, we have solved this step with the SVD algorithm (Algorithm 1 in Section 5.2).

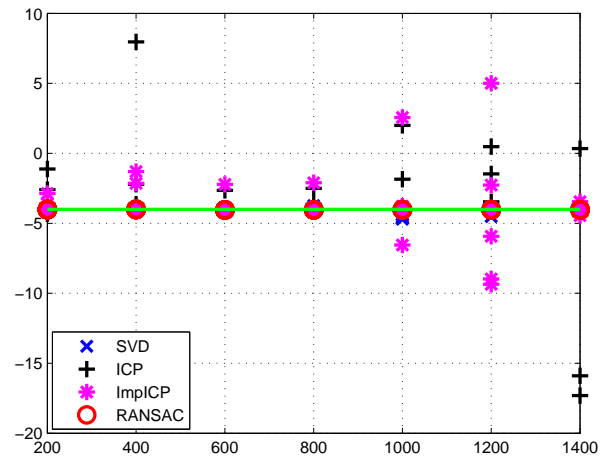
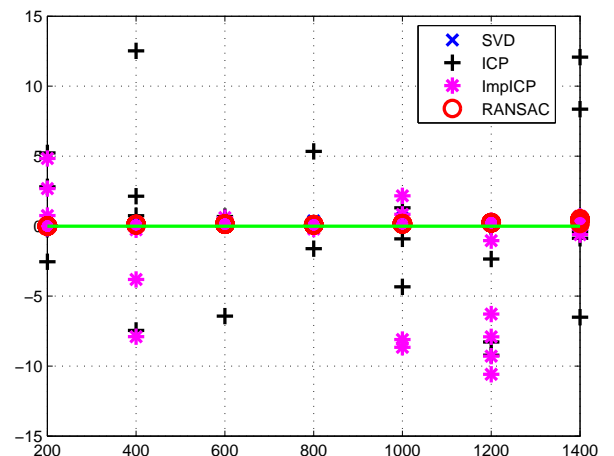
The algorithm stops when the set of correspondences does not change in the first step, and therefore T_{x+1} is equal to T in the second step.

This technique needs an accurate 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 (Section 5.1). The same threshold values are used.

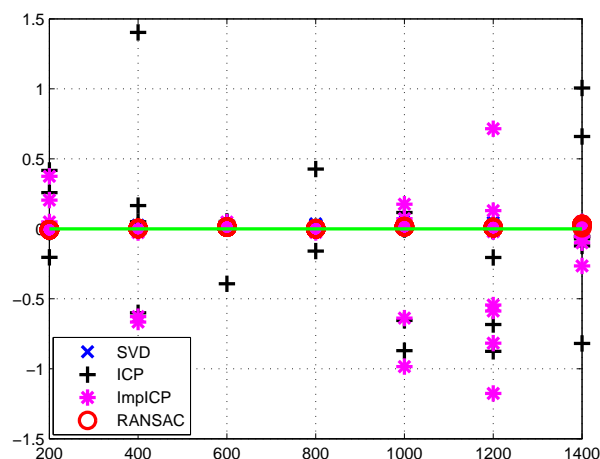
5.4 ImpICP (Improved Iterative Closest Point)

The improved ICP (ImpICP) method is a modification of the ICP algorithm presented in Section 5.3, which has been implemented *ad hoc*. This new version is motivated by the importance of obtaining a precise initial estimation of the transformation parameters T_0 . The accuracy of the results obtained is highly dependent on the goodness of this 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 (Section 5.1), leading to three initial estimates. For each initial estimate, the algorithm runs as in Section 5.3. Finally, the solution selected is the transformation that is supported by a highest number of correspondences.

6 Evaluation of the aligning methodsFigure 3: Results component t_x Figure 4: Results component t_y

In this section, we present a comparative evaluation of the aligning methods presented in Section 5. We consider the situation in which two robots begin the mapping task independently, starting at different positions. Each robot build its map with the FastSLAM algorithm using a set of $M = 100$ particles.

Figure 5: Results component θ

Particularly, 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 an extremely reduced number of correspondences 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. The alignment is done in several iterations of the FastSLAM algorithm. The most probable map is the map associated to the most probable particle of the filter at each particular moment. The aligning methods described above compute an estimate of alignment parameters t_x , t_y and θ . This first estimate is later refined with a least squares minimization [25], which eliminates outliers from the set of correspondences found by the aligning method. The result is a more refined estimate of the aligning parameters.

Figures 3, 4 and 5 show an example of the comparative evaluation performed. The x-axis represents the different k iterations of the FastSLAM algorithm at which the alignment is performed. In this case, we show values for $k = [200, 400, 600, 800, 1000, 1200, 1400]$. For each one of these k iterations, the alignment is performed by each method of Section 5, using the most probable map in each case. We have repeated the experiments 10 times, so that we obtain a set of values for each iteration and each aligning method. For reasons of

clarity, we have presented the results of the alignment parameters separately. Figures 3, 4 and 5 represent the results obtained for t_x , t_y and θ respectively. In those figures, the ground truth is represented with a green line. The values of the aligning parameters are the result of the least squares minimization and are expressed in meters.

In figure 3, for example, we can observe that ImplCP obtains values of t_x that vary from -10 to 5 meters, which is extremely unaccurate since the real value of t_x is -4. Then, in figure 4, the ICP algorithm obtains a variation of more than 10 meters around the ground truth. In general, ICP and ImplCP show the worst results in these experiments. SVD turns out to be much more accurate than the previous ones. The values obtained are very close to the ground truth in most cases. Only in Figure 3, at iterations $k = 1000$ and $k = 1200$, we observe that the estimate obtained is less accurate.

As we can deduce from the figures, RANSAC is, with no doubt, the method that obtains the most accurate results, having also the lowest variance.

6.1 Computational time

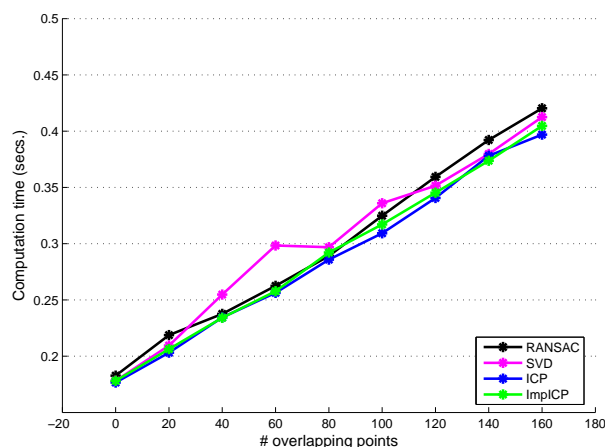


Figure 6: Computational time vs. number of overlapping points between the local maps.

We have evaluated these aligning methods not only qualitatively but also in terms of their computational efficiency. In Figure 6, a comparison of the computational time of the aligning methods presented in 5. In this figure, we present the time that it takes to obtain the aligning parameters (seconds) vs. different number of correspondent points between the local maps. Logically, the time is higher as the common part between the maps is bigger. It can be observed that the computational time of the different aligning

methods is very similar, so it can be deduced that this is not a determinant factor in order to select one of these methods as the most suitable to align visual landmark-based maps.

6.2 Results

Figure 7 shows an example of the alignment stage with two local landmark-based maps: map_1 and map_2 . Although the maps are 3D, due to clarity reasons, a 2D view of the maps is presented. These maps have been independently built by two robots and, at some point, the alignment transformation is computed. In this case, we show the result of aligning two maps with the solution given by RANSAC (Section 5.1). This method compares map_1 and map_2 , establishes correspondences between them by the similarity of the descriptors and computes an initial estimate of the aligning parameters. This initial estimate is improved afterwards by a least squares minimization. Fig. 7(c) shows the result of the alignment of map_1 and map_2 . In this case, the landmarks of both maps are expressed in the same reference system. map_1 is represented by circles and map_2 with diamonds. However, we still have two maps. In order to obtain a unique map, these maps should be fused. This stage is known as the map fusion problem.

7 Multirobot alignment

This section tackles the problem in which there are n robots ($n > 2$) whose maps should be aligned. In this case, the alignment should be consistent not only between pairs of maps but also globally. In order to deal with this situation, some constraints should be established [27].

First, given n maps ($n > 2$) and having each pair of them an overlapping part, the following constraint should be satisfied in the ideal case:

$$T_1 \cdot T_2 \cdot \dots \cdot T_n = I \quad (7)$$

where I is a 3×3 identity matrix. Each T_i is the transformation matrix between map_i and map_{i+1} and corresponds to the matrix in Eq.6. The particular case of T_n refers to the transformation matrix between map_n and map_1 . The constraint (7) leads to three expressions that should be minimized:

$$\begin{aligned} E1. & \sin(\theta_1 + \dots + \theta_n) \\ E2. & t_{x1} + t_{x2}\cos(\theta_1) + t_{y2}\sin(\theta_1) + t_{x3}\cos(\theta_1 + \theta_2) + \\ & t_{y3}\sin(\theta_1 + \theta_2) + \dots + t_{xn}\cos(\theta_1 + \dots + \theta_{n-1}) + \\ & t_{yn}\sin(\theta_1 + \dots + \theta_{n-1}) \end{aligned}$$

$$\begin{aligned} E3. & t_{y1} + t_{x2}\sin(\theta_1) + t_{y2}\cos(\theta_1) - t_{x3}\sin(\theta_1 + \theta_2) + \\ & t_{y3}\cos(\theta_1 + \theta_2) + \dots - t_{xn}\sin(\theta_1 + \dots + \theta_{n-1}) + \\ & t_{yn}\cos(\theta_1 + \dots + \theta_{n-1}) \end{aligned}$$

Additionally, given a set of corresponding landmarks between map_i and map_{i+1} , and having been aligned the landmarks of map_{i+1} (m_j) into map_1 's coordinate system with the transformation matrix T_i (see Equation 6), the following expression should be minimized:

$$m'_j\{m(k)\} - m_i\{m(k)\} \quad (8)$$

where $c(k)$ is the total number of correspondences between the k -pair of maps ($k \in \{1, n\}$). The number of equations that emerge from Equation 8 is $2c(1) + 2c(2) + \dots + 2c(n)$. For instance, if we have $c(1)$ common landmarks between map_1 and map_2 and the transformation matrix between them is T_1 , then for each common landmark we should minimize the following set of expressions:

$$E\delta. x_2\cos(\theta_1) + y_2\sin(\theta_1) + t_{x1} - x_1 \text{ with } \delta \in \{4, X + 4\}$$

$$E\lambda. y_2\cos(\theta_1) - x_2\sin(\theta_1) + t_{y1} - y_1 \text{ with } \lambda \in \{X + 5, 3X + 5\}$$

where $X = m(1) + m(2) + \dots + m(n)$.

So far, we have a non-linear system of $S = 3 + 2c(1) + 2c(2) + \dots + 2c(n)$ constraints that we should minimize. In order to obtain the aligning parameters that minimize the previous S constraints, we use the MATLAB function **fsolve**. This iterative algorithm uses a subspace trust-region method which is based on the interior-reflective Newton method described in [7, 8]. The input for this algorithm is a initial estimate of the aligning parameters. This is obtained by the RANSAC algorithm of Sec.5.1 between each pair of maps, i.e., $map_1 - map_2$, $map_2 - map_3$, $map_3 - map_4$ and $map_4 - map_1$. This will be the starting point for the **fsolve** function to find a final solution.

In Table 1 an example of the results obtained with **fsolve** function is presented. This table shows the aligning results obtained in a group of four robots, where A_j^i represents the alignment between robot i and robot j . At the top of the table, we can observe the aligning results between each pair of robots. These alignment parameters (t_x, t_y and θ) have been computed with the RANSAC algorithm of Section 5.1. These solutions are valid between pairs of maps but may not be consistent globally. Then, at the bottom of the table, the alignment parameters (t'_x, t'_y and θ') have been obtained with the **fsolve** function. In this case, the constraints imposed (see constraints $E1$ to $E\lambda$ of this section) optimizes the solution so that it is globally consistent.

Table 1: Alignment parameters

| | t_x | t_y | θ |
|---------------------|---------|---------|-----------|
| $A_{\frac{1}{2}}^1$ | -0.0676 | -0.0636 | -0.0144 |
| $A_{\frac{2}{3}}^2$ | 0.1174 | 0.0423 | -0.0063 |
| $A_{\frac{3}{4}}^3$ | -0.0386 | 0.8602 | 0.0286 |
| $A_{\frac{4}{1}}^4$ | 0.0547 | -0.8713 | -0.0248 |
| | t'_x | t'_y | θ' |
| $A_{\frac{1}{2}}^1$ | -0.0388 | 0.0363 | 0.0079 |
| $A_{\frac{2}{3}}^2$ | 0.0677 | -0.1209 | -0.0375 |
| $A_{\frac{3}{4}}^3$ | -0.0408 | 0.9521 | 0.0534 |
| $A_{\frac{4}{1}}^4$ | 0.0774 | -0.9220 | -0.0436 |

8 Conclusion

In this paper we present an approach that tackles the multirobot SLAM problem. Particularly, we focus on the situation in which a team of robots build their own local maps of the environment.

The task of building a map of the environment by a team of robots can be tackled in two different ways. On the one hand, the robots can build a map jointly. The result is a map which is common to the team of robots. On the other hand, each robot can build its own map independently. In this case, the result is a set of local maps which can be fused into a global one.

The main advantage of this approach is that the robots do not need to know other robots' positions, so that the map building can be performed independently. We concentrate on the map alignment stage, in which the transformation between different reference systems is computed.

In our experiments, the robots construct their maps by extracting Harris points from the environment. These points are characterized by a U-SURF descriptor in order to deal with the data association problem. Moreover, the FastSLAM algorithm is used to build the map. The maps thus built consist of the 3D coordinates of the landmarks, their corresponding uncertainty and an associated descriptor.

We have evaluated a set of methods in order to find the most suitable for aligning this kind of maps. These methods establish correspondent landmarks based on the descriptor similarity. Then, these estimates are refined by means of a least squares minimization. The result is a more accurate solution. The results obtained show that RANSAC has the best performance.

Additionally, we have studied the multirobot case,

in which the alignment should be consistent not only for pairs of maps but also for a bigger set.

In the future, our aim is to concentrate on the map merging stage, which is necessary to obtain a unique global map. In this case, we have to consider that the landmarks are integrated by different robots and therefore the source of uncertainty is different. Furthermore, as future work, we want to study the alignment and merging problem as being integrated in the mapping process. So far, the robots build independently their local maps, and then, at some point, they may decide to fusion these local maps. But it would be also interesting if the robots share observations during their mapping task. That is to say, each robot will have their own observations and observations from other robots of the team. The robot should be able to refer other robots' observations into its reference system (alignment) and integrate them as landmarks in the map.

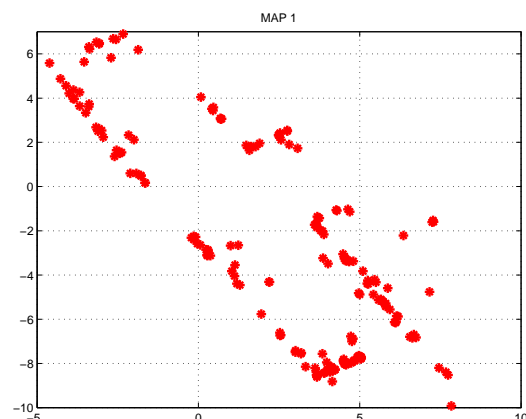
Acknowledgements: The research has been supported by the Spanish Ministry of Science and Innovation under projects DPI2004-07433- C02-01 and CI-CYT DPI2007-61197 and by the Generalitat Valenciana under grant BFPI/2007/096.

References:

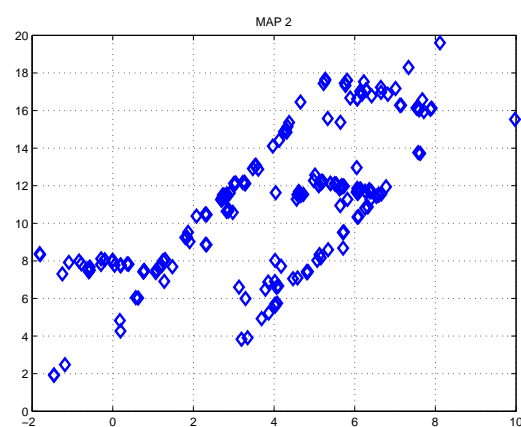
- [1] K.S. Arun, T.S. Huang, and S.D. Blostein. Least square fitting of two 3d sets. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. PAMI-9 no. 5, pp.698-700, 1987.
- [2] M. Ballesta, A. Gil, O. Martínez Mozos, and O. Reinoso. Local descriptors for visual SLAM. In *Workshop on Robotics and Mathematics (ROBOMAT07), Portugal, 2007*.
- [3] M. Ballesta, A. Gil, O. Reinoso, and D. Ubeda. Analisis de detectores y descriptores de características visuales en slam en entornos interiores y exteriores. In *To appear in RIAI (Revista Iberoamericana de Automatica e Informatica Industrial)*, 2010.
- [4] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded-up robust features. In *Proc. of the 9th European Conference on Computer Vision*, 2006.
- [5] P.J. Besl and N. McKay. A method for registration of 3-d shapes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. PAMI-14 no. 2, pp. 239-256, 1992.

- [6] S. Carpin, A. Birk, and V. Jucikas. On map merging. In *Robotics and Autonomous Systems, Elsevier Science*, volume 53, pages 1–14, 2005.
- [7] T.F. Coleman and Y. Li. On the convergence of reflective newton methods for large-scale non-linear minimization subject to bounds. In *Mathematical Programming*, volume 67, pages 189–224, 1994.
- [8] T.F. Coleman and Y. Li. An interior, trust region approach for nonlinear minimization subject to bounds. In *SIAM Journal on Optimization*, 1996.
- [9] A. Cumani, S. Denasi, A. Guiducci, and G. Quaglia. Integrating monocular vision and odometry for slam. In *WSEAS Transactions on Computers*, volume 3, pages 625–630, 2004.
- [10] A. Cumani and A. Guiducci. Fast stereo-based visual odometry for rover navigation. In *WSEAS Transactions on Circuits and Systems, ISSN:1109-2734*, volume 7, pages 648–657, 2008.
- [11] G. Dedeoglu and G. Sukhatme. Landmark-based matching algorithm for cooperative mapping by autonomous robots. In *DARS*, pages 251–260, 2000.
- [12] John W. Fenwick, Paul N. Newman, and John J. Leonard. Cooperative concurrent mapping and localization. In *Proc. of the 2002 IEEE International Conference on Intelligent Robotics and Automation*, pp.1810-1817, 2002.
- [13] D. Fox. Distributed multi-robot exploration and mapping. In *Proc. of the 2nd Canadian conference on Computer and Robot Vision*, 2005.
- [14] A. Gil, O. Martinez Mozos, M. Ballesta, and O. Reinoso. A comparative evaluation of interest point detectors and local descriptors for visual slam. *Machine Vision and Applications*, pages 1432–1769, 2009.
- [15] A. Gil, O. Reinoso, L. Payá, and M. Ballesta. Influencia de los parámetros de un filtro de partículas en la solución al problema de SLAM. In *IEEE Latin America*, 2007.
- [16] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1998.
- [17] 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.
- [18] 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.
- [19] N. Kwak, G-W. Kim, S-H. Ji, and B-H. Lee. A mobile robot exploration strategy with low cost sonar and tungsten-halogen structural light. *Journal of Intelligent and Robotic Systems*, 51(1):89–111, 2008.
- [20] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(4), 1991.
- [21] D. Maier, J. Hessler, and R. Manner. Fast and accurate closest point search on triangulated surfaces and its application to head motion estimation. In *In 3rd WSEAS International Conference on SIGNAL, SPEECH and IMAGE PROCESSING*, 2003.
- [22] 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.
- [23] N.E. Mastorakis. The singular value decomposition (svd) in tensors (multidimensional arrays) as an optimization problem. solution via genetic algorithms and method of nelder-mead. In *In 6th WSEAS International Conference on Systems Theory and Scientific Computation*, pages 7–13, 2006.
- [24] 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. 593-598. Edmonton, Canada, 2002.
- [25] S.T. Pfister, K. L. Kreichbaum, S.I. Roumeliotis, and J.W. Burdick. Weighted range sensor matching algorithms for mobile robot displacement estimation. In *IEEE International Conference on Robotics and Automation, Washington D.C.*, pages 1667–74, 2002.
- [26] J. Rieger. On the classification of views of piecewise smooth objects. In *Image and Vision Computing*, vol. 5, no. 2, pp. 91-97, 1987.

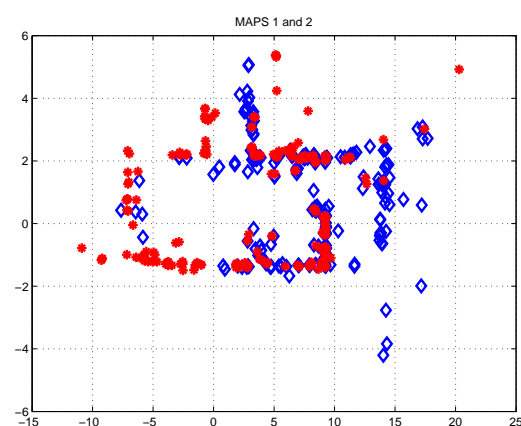
- [27] S. Se, D. Lowe, and J.J. Little. Vision-based global localization and mapping for mobile robots. In *IEEE Transactions on robotics*, vol.21, no.3, 2005.
- [28] S. Takahashi, I. R. Khan, M. Okuda, and M. Ikehara. 3d avatar modeling and transmission using cylinder mapping. In *WSEAS Transactions on Electronics*, ISSN: 1109-9445, volume 5, pages 57–64, 2008.
- [29] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. In *Int. Journal of Robotics Research*, volume 20, pages 335–363, 2001.
- [30] R. Triebel and W. Burgard. Improving simultaneous mapping and localization. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2005.
- [31] 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.
- [32] O. Wijk and H. I. Christensen. Localization and navigation of a mobile robot using natural point landmark extracted from sonar data. In *Robotics and Autonomous Systems*, volume 3, pages 31–42, 2000.
- [33] Z. Zhang. On local matching of free-form curves. In *Proc. of BMVC*, pp. 347-356, 1992.
- [34] Xun S. Zhou and Sergios I. Roumeliotis. Multi-robot slam with unknown initial correspondence: The robot rendezvous case. In *Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China*, pp. 1785-1792, 2006.



(a)



(b)



(c)

Figure 7: Map alignment (2D view). Fig. 7(a) shows the local map 1 before the alignment. Fig. 7(b) shows map 2. Fig. 7(c) show the same maps after the alignment.