# Robot Localization and Map Building

Edited by
Hanafiah Yussof

*In-Tech*
*intechweb.org*

Published by In-Teh

**In-Teh**
Olajnica 19/2, 32000 Vukovar, Croatia


www.intechweb.org
Additional copies can be obtained from:
publication@intechweb.org

First published March 2010
Printed in India

Technical Editor: Sonja Mujacic
Cover designed by Dino Smrekar

Robot Localization and Map Building,
Edited by Hanafiah Yussof

p. cm.
ISBN 978-953-7619-83-1

# Contents

7

# Evaluation of aligning methods for landmark-based maps in visual SLAM

Mónica Ballesta, Óscar Reinoso, Arturo Gil,
Luis Payá and Miguel Juliá
*Miguel Hernandez University of Elche*
*Spain*

## 1. Introduction

Building maps is one of the most fundamental tasks for an autonomous robot. This robot should be able to construct a map of the environment and, at the same time, localize itself in it. This problem, known as Simultaneous Localization and Mapping (SLAM), has received great interest in the last years (Leonard & Durrant-Whyte, 1991).
In our particular case, the robots build their maps using the FastSLAM algorithm (Montemerlo et al., 2002). 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. This algorithm uses a particle set that represents the uncertainty of the robot's pose (localization problem) meanwhile each particle has its own associated map (several individual estimates of the landmarks conditioned to the robot's path). 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 proved to be robust to false data association and it is able to represent models of non-linear movements in a reliable way (Howard, 2006).
In relation to the sensors used to build the maps, many authors use range sensors such as sonar (Kwak et al., 2008; Wijk & Christensen, 2000) or laser (Thrun, 2004; Triebel & Burgard, 2005). Nevertheless, there is an increasing interest on using cameras as sensors. This approach is denoted as visual SLAM (Valls Miro et al., 2006). These devices obtain a higher amount of information from the environment and they are less expensive than laser as well. Furthermore, 3D information can be obtained when using stereo cameras. These are the sensors used in this work.
Most visual SLAM approaches are landmark-based. These landmarks consist of a set of distinctive points which are referred to a global reference system. 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.
The map building problem can be solved by a single robot (Moutalier & Chatila, 1989), but it will be more efficiently managed if there is a team of robots, which collaborates in the construction of the map (Howard, 2006). In this case, the space can be divided so that the distances traversed are shorter and thus the odometry errors will be smaller. In this work,

we focus on this approach. In this context, two main proposals can be found in the literature. On the one hand, there are some solutions in which the estimate of the maps and trajectories is performed jointly (Fenwick et al., 2002; Gil et al., 2007; Thrun & Liu, 2004). In this case, there is a unique map, which is simultaneous built from the observations of the robots. In this way, the robots have a global notion of the unexplored areas so that the cooperative exploration can be improved. Moreover, in a feature-based SLAM, a landmark can be updated by different robots in such a way that the robots do not need to revisit a previously explored area in order to close the loop and reduce its uncertainty. However, the maintenance of this global map can be computationally expensive and the initial position of the robots should be known, which may not be possible in practice.
On the other hand, some approaches consider the case in which each robot builds a local map independently (Stewart et al., 2003; Zhou & Roumeliotis, 2006). Then, at some point the robots may decide to fuse their maps into a global one. In (Stewart et al., 2003), there is some point where the robots arrange to meet in. At that point, the robots can compute their relative positions and fuse their maps. One of the main advantages of using independent local maps, as explained in (Williams, 2001), is that the data association problem is improved. First, new observations should be only matched with a reduced number of landmarks in the map. Moreover, when these landmarks are fused into a global map, a more robust data association can be performed between the local maps. However, one of the drawbacks of this approach is dealing with the uncertainty of the local maps built by different robots when merging them.
The map fusion problem can be divided into two subproblems: the map alignment and the fusion of the data. The first stage consists in computing the transformation between the local maps, which have different reference systems. Next, after expressing all the landmarks in the same reference system, the data can be fused into a global map. In this work, we focus on the alignment problem in a multirobot visual SLAM context.

## 2. Map Building

The experiments have been carried out with Pioneer-P3AT robots, provided with a laser sensor and STH-MDCS2 stereo head from Videre Design. The stereo cameras have been previously calibrated and obtain 3D information from the environment. The maps thus built, are made of visual landmarks. These visual landmarks consist of the 3D position of the distinctive points extracted by the Harris Corner detector (Harris & Stephens, 1998). These points have an associated covariance matrix representing the uncertainty in the estimate of the landmarks. Furthermore these points are characterized by the U-SURF descriptor (Bay et al., 2006). The selection of the Harris Corner detector combined with the U-SURF descriptor is the result of a previous work, in which the aim was to find a suitable feature extractor for visual SLAM (Ballesta et al., 2007; Martinez Mozos et al., 2007; Gil et al., 2009).
The robots start at different positions and perform different trajectories in a 2D plane, sharing a common space in a typical office building. The maps are built with the FastSLAM algorithm using exclusively visual information. Laser readings are used as ground truth. The number of particles selected for the FastSLAM algorithm is M=200.
The alignment experiments have been initially carried out using two maps from two different robots (Section 5.1 and 5.2). Then, four different maps were used for the multi-robot alignment experiments (Section 5.2.1). The trajectories of the robots can be seen in

Figure 1. The laser measurements have been used as ground truth in order to estimate the accuracy of the results obtained.

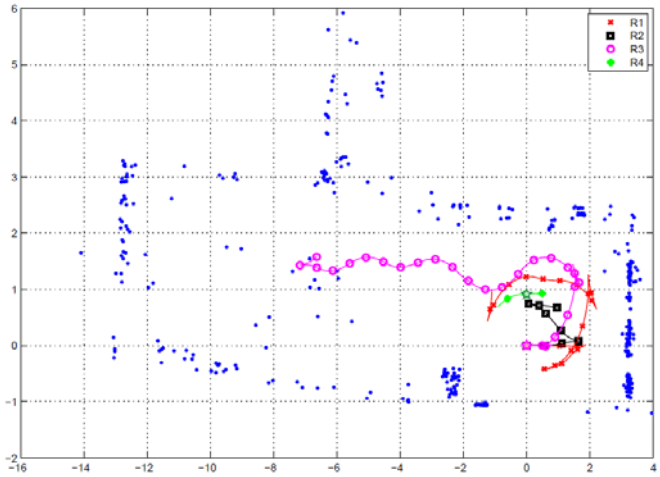



Fig. 1. Trajectories performed by four Pioneer P3AT robots and a 2D view of the global map.

## 3. Map alignment

The main objective of this work is to study the alignment stage in a multi-robot visual SLAM context. At the beginning, the robots start performing their navigation tasks independently, and build local maps. Given two of these feature maps, computing the alignment means computing the transformation, if existent, between those maps. In this way the landmarks belonging to different maps are expressed into the same reference system. Initially, before the alignment is performed, the local map of each robot is referred to its local reference system which is located at the starting point of the robot.

In order to compute the transformation between local maps, some approaches try to compute the relative poses of the robots. In this sense, the easiest case can be seen in (Thrun, 2001), where the relative pose of the robots is suppose to be known. A more challenging approach is presented in (Konolige et al., 2003; Zhou & Roumeliotis, 2006). In these cases, the robots, being in communication range, agree to meet at some point. If the meeting succeed, then the robots share information and compute their relative poses. Other approaches present feature-based techniques in order to align maps (Se et al., 2005; Thrun & Liu, 2004). The basis of these techniques is to find matches between the landmarks of the local maps and then to obtain the transformation between them. This paper focuses on the last approach.

In our case, although the maps are 3D, the alignment is performed in 2D. This is due to the fact that the robots' movements are performed in a 2D plane. The result of the alignment is a translation in x and y ($t_x$ and $t_y$) and a rotation $\theta$. This can be expressed as a transformation matrix T:

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

Given two maps m and m', T transforms the reference system of m' into the reference system of m.

In order to select an adequate method to align this kind of maps, we have performed a comparative evaluation of a set of aligning methods (Section 4). All these methods try to establish correspondences between the local maps by means of the descriptor similarity. Furthermore, we have divided this study into to stages: first with simulated data (Section 5.1) and then with real data captured by the robots (Section 5.2). These experiments are performed between pairs of maps. However, we have additionally considered the multi-robot case, in which the number of robots is higher than 2. In this case, the alignment should be consistent, not only between pair of maps but also globally. This is explained in detail in the next section (Section 3.1).

### 3.1 Multi-robot 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 (Se et al., 2005).

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:

$$T_1 \cdot T_2 \cdot \ldots \cdot T_n = I \quad (2)$$

where I is a 3X3 identity matrix. Each $T_i$ is the transformation matrix between $map_i$ and $map_{i+1}$ and corresponds to the matrix in Equation 1. The particular case of $T_n$ refers to the transformation matrix between $map_n$ and $map_1$. The equation 2 leads to three expressions that should be minimized:

E1. $\sin(\theta_1+\ldots+\theta_n)$

E2. $t_{x1} + t_{x2}\cos(\theta_1) + t_{y2}\sin(\theta_1) + t_{x3}\cos(\theta_1+\theta_2) + t_{y3}\sin(\theta_1+\theta_2) + \ldots + t_{xn}\cos(\theta_1+\ldots+\theta_{n-1}) + t_{yn}\sin(\theta_1+\ldots+\theta_{n-1})$

E2. $t_{y1} + t_{x2}\sin(\theta_1) + t_{y2}\cos(\theta_1) - t_{x3}\sin(\theta_1+\theta_2) + t_{y3}\cos(\theta_1+\theta_2) + \ldots - t_{xn}\sin(\theta_1+\ldots+\theta_{n-1}) + t_{yn}\cos(\theta_1+\ldots+\theta_{n-1})$

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

$$L_{Aj\{m\{k\}\}} - L_{i\{m(k)\}} \quad (3)$$

where m(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 3 is $2m(1)+2m(2)+\ldots+2m(n)$. For instance, if we have m(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:
Eδ. x2cos($\theta_1$)+y2sin($\theta_1$)+tx1-x1 with $\delta \in \{4,X+4\}$
Eλ. y2cos($\theta_1$)-x2sin($\theta_1$)+ty1-y1 with $\lambda \in \{X+5,3X+5\}$
where X=m(1)+m(2)+…+m(n)
So far, we have a non-linear system of S = 3 + 2m(1) +…+ 2m(n) constraints that we should minimize. In order to obtain the aligning parameters that minimize the previous S constraints, we used the **fsolve** MATLAB function. This iterative algorithm uses a subspace trust-region method which is based on the interior-reflective Newton method described in (Coleman, 1994; Coleman, 1996). The input for this algorithm is an initial estimate of the aligning parameters. This is obtained by the RANSAC algorithm of Sec. 4.1 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 of the results obtained with fsolve function to find a final solution.

## 4. Aligning methods

### 4.1 RANSAC (Random Sample Consensus)

This algorithm has been already aplied to the map alignment problem in (Se et al., 2005). The algorithm is performed as follows.
(a) In the first step, a list of possible corresponedences is obtained. The matching between landmarks of both maps in 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 $m'$.
(b) In a second step, two pair of correspondences ($[(x_i,y_i,z_i),(x_i',y_i',z_i')]$) are selected at random from the previous list. These pairs should satisfy the following geometric constraint (Se et al., 2005):

$$|(A^2 + B^2)-(C^2+D^2)| < th_1 \tag{4}$$

where $A = (x_i'-x_j')$, $B = (yi'-yj')$, $C = (xi-xj)$ and $D = (yi-yj)$ . We have set the threshold $th_1 = 0.8$ m. The two pairs of correspondences are used to compute the alignment parameters ($t_x$, $t_y$, $\theta$) with the following equations:

$$t_x = x_i - x_i'cos\theta - y_i' sin\theta \tag{5}$$

$$t_y = y_i - y_i'cos\theta + x_i' sin\theta \tag{6}$$

$$\theta = arctan((BC-AD)/(AC+BD)) \tag{7}$$

(c) The third step consist in looking for correspondences that support the solution obtained ($t_x$, $t_y$, $\theta$). Concretely, we transform the landmarks of the second map using the alignment obtained, so that it is referred to the same references 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 $th_2$=0.4m. As a result, we will have a set of matches that support the solution of the alignment.
(d) 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 the initial correspondences, $th_1$=0.8 for the geometric constraint of Equation 4 and $th_2$=0.4m for selecting supports. Furthermore, a parameter min =20 establishes the minimum number of supports in order to validae 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.

### 4.2 SVD (Singular Value Decomposition)

One of the applications of the Singular Value Decomposition (SVD) is the registration of 3D point sets (Arun et al., 1987; Rieger, 1987). The registration consists in obtaining a common reference frame by estimating the transformations between the datasets. 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 4.1). In addition, the geometric constraint of Equation 4 is evaluated. Given this list of possible correspondences, our aim is to minimize the following expression:

$$||Tm'-m|| \tag{8}$$

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 1). T is computed as shown in Algorithm 1 of this section.

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

### 4.3 ICP (Iterated Closest Point)

The Iterative Closest Point (ICP) technique was introduced in (Besl & McKay, 1992; Zhang, 1992) and applied to the task of point registration. 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 5, 6 and 7.
(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: $||T_{x+1}\, m'-m||$, which is

analogous to the expression (5). For this reason, we have solved this step with the SVD algorithm (Algorithm 1).
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 4.1). The same threshold values are used.

### 4.4 ImpICP (Improved Iterated Closest Point)

The improved ICP (ImpICP) method is a modification of the ICP algorithm presented in Section 4.3, which has been implemented ad hoc. This new version is motivated by the importance of obtaining a precise initial estimation of the transformation parameter $T_0$. 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 4.1), leading to three initial estimates. For each initial estimate, the algorithm runs as in Section 4.3. Finally, the solution selected is the transformation that is supported by a highest number of correspondences.

## 5. Experiments

The aim of this work is to find a suitable aligning method so that two or more maps can be expressed in the same reference system. This method should be appropriate for the kind of maps that our robots build, that is to say, landmark-based maps. With this idea, we have selected a set of algorithms that satisfy this requirement (See Section 4).
In order to perform these experiments, we have organised the work in two stages: first, a comparison of the aligning methods selected, using simulated data. In this case, we vary the amount of noise of the input data and observe the results by aligning pairs of maps. Secondly, we repeated the same experiments using real data captured by the robots. Furthermore, we include an experiment showing the performance of the multi-alignment case explained in Section 3.1, in which the number of maps we want to align is higher than 2.

### 5.1 Simulated Data

In order to perform the comparison between the aligning methods, we have built two 3D feature maps as can be seen in Figure 2. The coordinates of the landmarks have been simulated so that the alignment is evaluated with independence of the uncertainty in the estimate of the landmarks. Then, these points are described by U-SURF, extracted from real images which are typical scenarios of our laboratory. $map_1$ from Figure 2 has 250 points (stars), whereas map2 (circles) has a common area with $map_1$, whose size is variable, and a non-overlapping part which has 88 points. During the experimental performance, we test with different sizes of the overlapping area between these two maps (pentagons), so that we can observe the performance of the aligning methods vs. different levels of coincidence between the maps.

$map_2$ is 0.35 rads rotated and $t_x$ = 5m, $t_y$ = 10 m translated from map1. The size of the maps is 30X30 metres. Coincident points between the maps have initially the same descriptors. However, a Gaussian noise has been added to map2 so that the data are closer to reality. As a consequence, map2 has noise with σL in the localization of the points (coordinates' estimate) and noise with σD in the descriptors. The magnitude of σL and σD has been chosen experimentally.

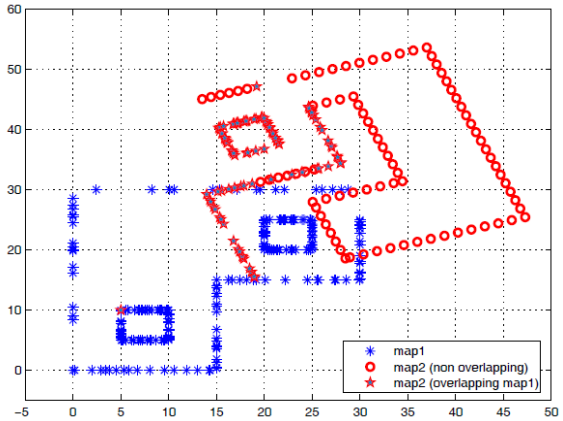



Fig. 2. 2D view of map1 and map2 (simulated data).

Figures 3, 4, 5 and 6 represent the results obtained with a noise of σL and σD equal to 0.20, whereas in Figures 7, 8, 9 and 10 these values are 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 overlap at all. For each value, the experiment is repeated 10 times. Then the Mean Quadratic Error is shown in the Y-axis (blue line). This value is computed comparing the alignment obtained and a ground truth. The blue points represent 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. The mean value of supports is represented with a red line. In Figures 3 and 7, the green line represents the minimum value of supports required to validate the solution. Finally, all the figures present with bars the number of failures obtained in the 10 repetitions. Each failure represents the case in which the method does not converge to any solution or the solution does not satisfy the requirements (RANSAC method). In these cases, we consider that no alignment has been found.

Figures 3 and 7 show the results obtained with the RANSAC algorithm of Sec. 4.1. In figure 3, no solution is obtained until the number of overlapping points is higher than 60 points. In all of those cases, the Mean Quadratic Error always below 2 m. Regarding the number of supports (red line), we observe an ascendant tendency due to the increasing number of overlapping points. In the case of 160 overlapping points, the number of supports is 80. If the Gaussian noise is higher, these results get worse, as in Figure 7 where the number of supports obtained is significantly lower. Furthermore, the first solution appears when the number of overlapping points is 120.

Figures 4 and 8 present the results of the SVD algorithm of Section 4.2. In those cases, 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 behaviour of the ICP algorithm of Section 4.3 is presented in Figs. 5 and 9. In Figure 5 the error value obtained is 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 very slightly. This last part of the curve shows that the error values are around 2, which is a quite good result. Nevertheless, the yellow bars show, in some cases, a small number of failures. Fig. 9 shows worse results.

Finally, in Figures 6 and 10, the results of the improved version of ICP are shown. In this case, the results obtained are similar to that of ICP in terms of mean support values. However, it is noticeable that the stability of the algorithm is higher. If we pay attention to the yellow bars in Figure 6, it is shown that the algorithm always obtains a solution when the number of overlapping points is equal or higher than 100. In Figure 10, the number of failures is lower than in Figure 9.

In general, the best results are obtained by the ImpICP and RANSAC algorithms. RANSAC obtains lower error values, whereas ImpICP is more stable in terms of having less number of failures.

In addition to the experiments performed to evaluate the accuracy and suitability of the aligning methods, we have also measured the computational time of each algorithm (see Figure 11). The curves show an ascendant tendency. This is due to the fact that the size of map2 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.

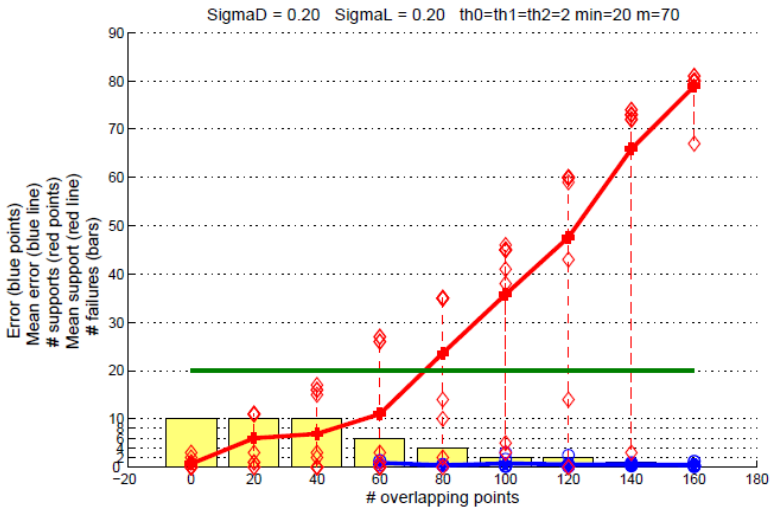



Fig. 3. RANSAC algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.20$.

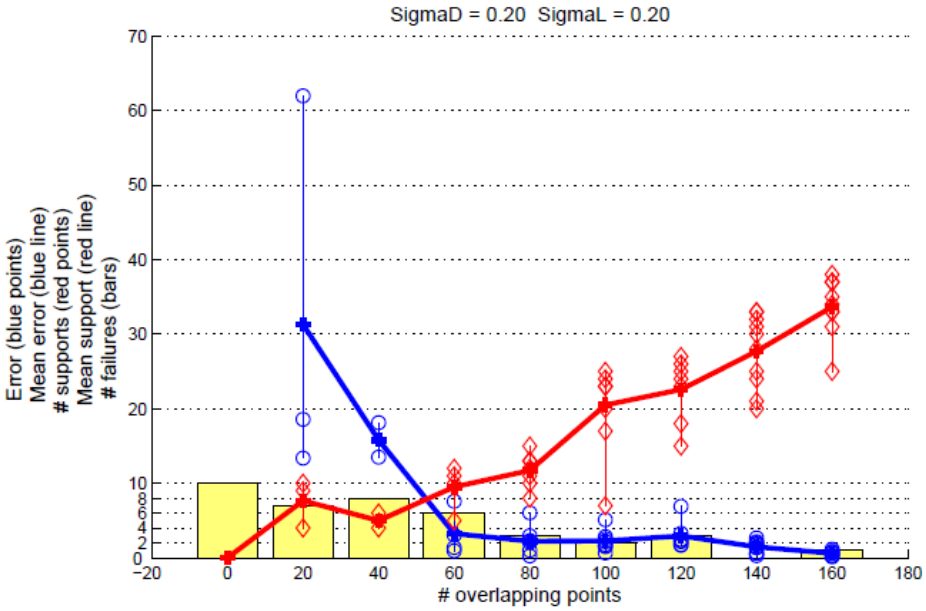



Fig. 4. ICP algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.20$.

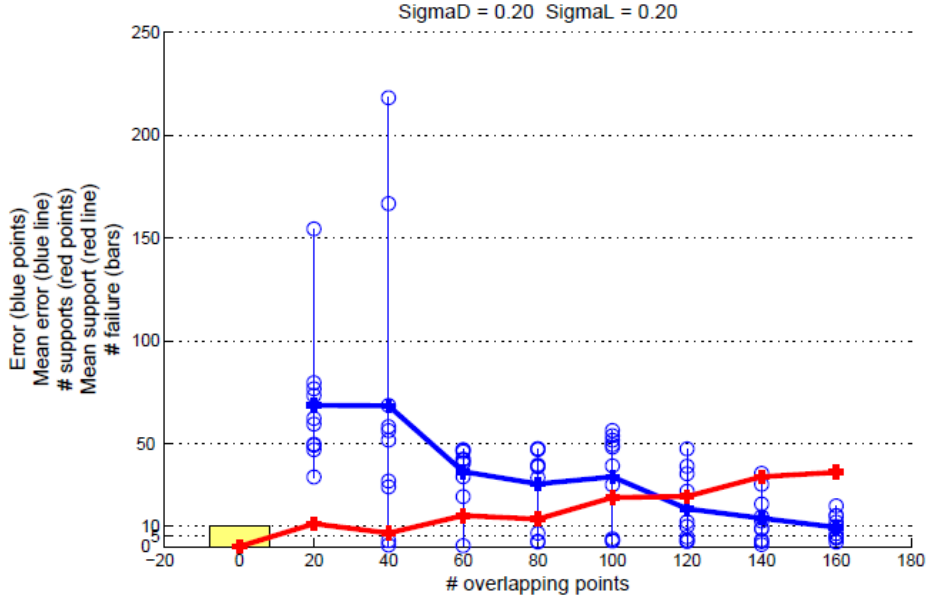



Fig. 5. SVD algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.20$.

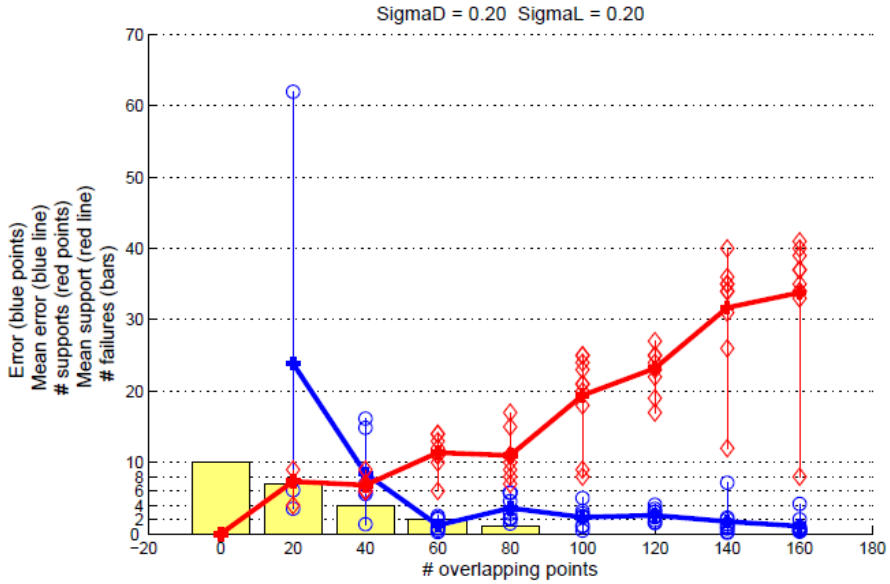



Fig. 6. ImpICP algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.20$.

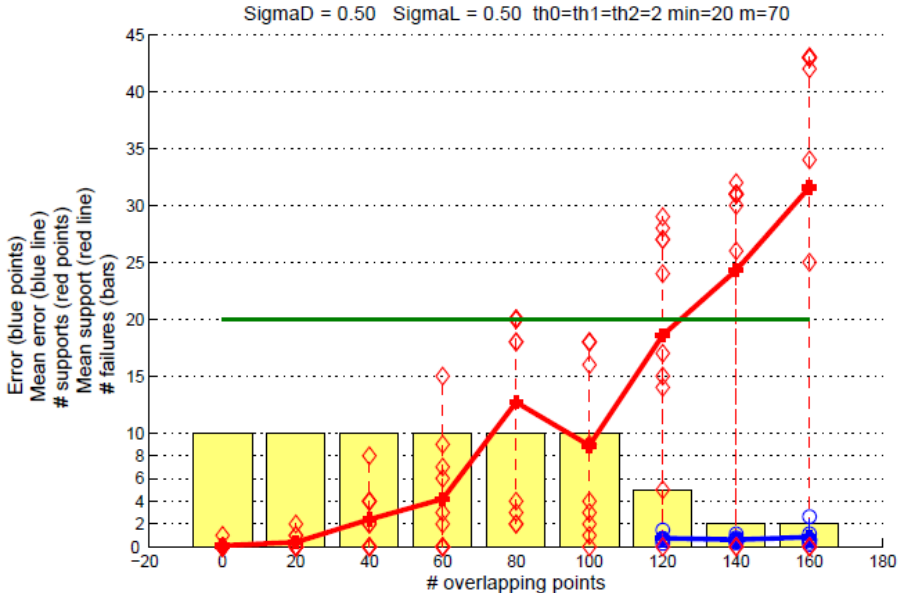



Fig. 7. RANSAC algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.50$.

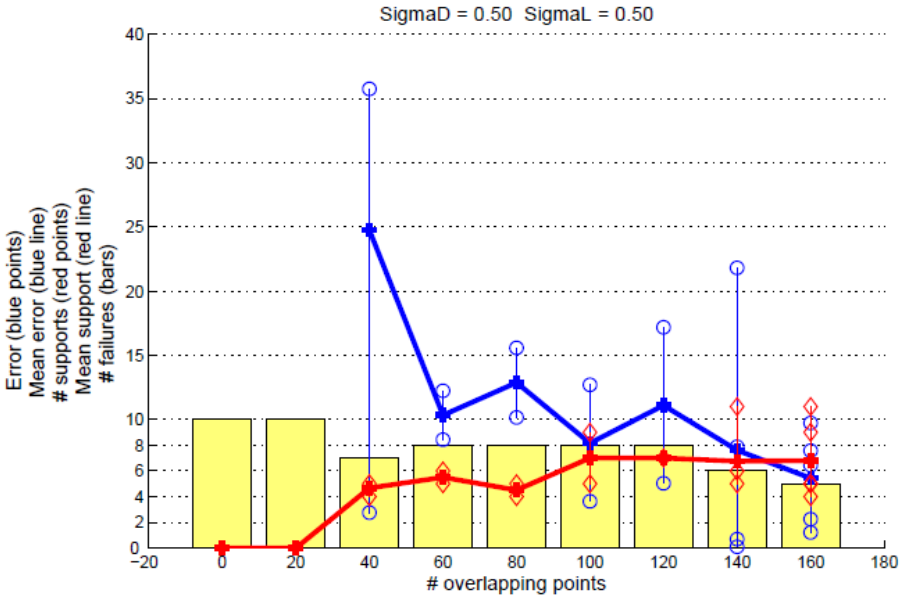



Fig. 8. ICP algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.50$.

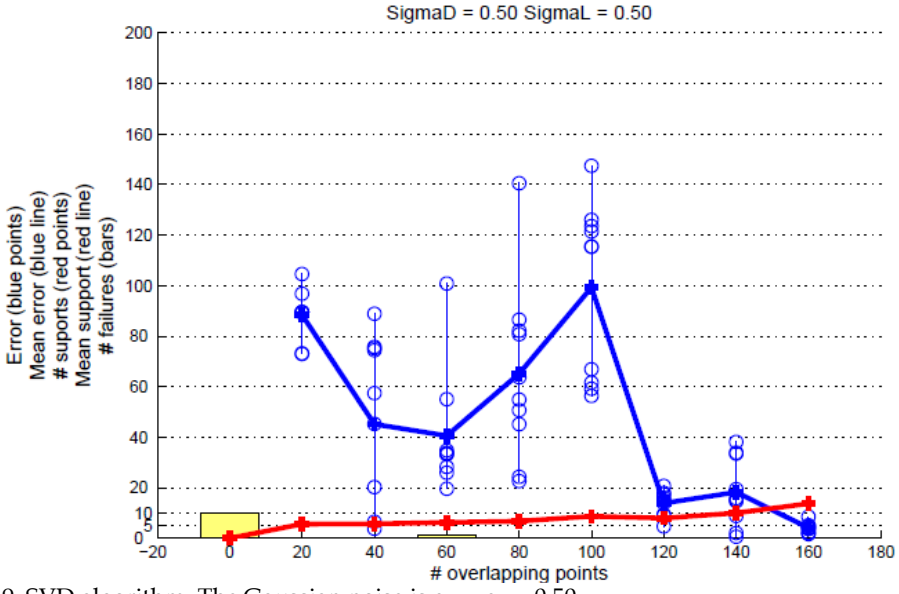



Fig. 9. SVD algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.50$.

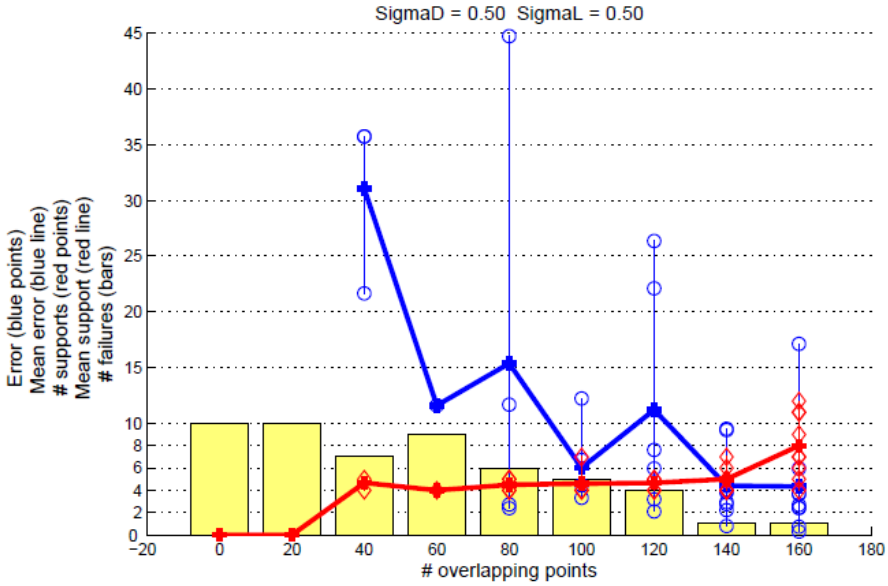



Fig. 10. ImpICP algorithm. The Gaussian noise is $\sigma_D = \sigma_L = 0.50$.

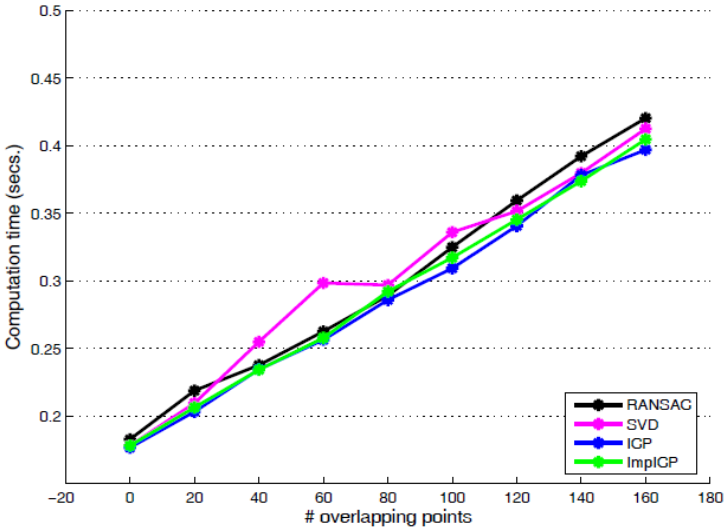



Fig. 11. Computational time vs. number of overlapping points.

### 5.2 Real Data

After performing the comparative analysis with the simulated data, the next step is to evaluate the same aligning methods using real data captured by the robots, i.e., landmarks consisting of Harris points detected from the environment and described by U-SURF.

We evaluate the performance of the aligning methods at different steps of the mapping process, i.e., at different iterations of the FastSLAM algorithm. At the beginning, the maps built by each robot have a reduced number of landmarks and therefore there are fewer possibilities of finding correspondences between these local maps. However, this situation changes as long as the maps are bigger. In this situation the probability of finding correspondences is higher and it is expected to obtain the alignment successfully.
In these experiments we have used the most probable map of each robot in order to compute the transformation between their reference systems. We obtain the most probable map of each robot at different iterations of the FastSLAM algorithm and try to align these maps. The most probable map is the map of the most probable particle of the filter in each particular moment.
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 number of iterations. In this case, this number is k=1410 and the sizes of the maps at that point are $map_1$=263 landmarks and $map_2$=346 landmarks. These maps have a dimension of 35X15 meters approximately.
In Figure 15(a), we can observe a 2D view of 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 it size is of 187 landmarks. In Figure 15(b), we can see the two local maps already aligned. In this case, the most probable maps of iteration k=810 have been used.
In order to compare the aligning methods with real data, we compute the alignment parameters for each method at different iterations of the FastSLAM algorithm. The evaluating measure is the Euclidean distance between the alignment parameters tx,ty and $\theta$ the real relative position between the robots, denoted as ground truth. This measure was obtained estimating the relative position of the robots being at their starting positions.
Figure 12, illustrates the comparison of the aligning methods we 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 increases, the size of the maps constructed will be higher and therefore it will be more probable to find a solution closer to the ground truth. For this reason, it is expected to obtain small error values as the k-iteration increases. In Figure 12 we can observe that the worst results are obtained with SVD. For instance, SVD has an error of 4m with k-iteration =1409, i.e., at the end of the FasSLAM 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 1 m. 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, Figures 13 and 14 focus on the results obtained by the RANSAC algorithm. Figure 13 shows the number of supports obtained in each case, which increases with the k-iteration values. On the other hand, Figure 14 shows the decomposition of the error in its three components (three alignment parameteres): error in tx, ty and $\theta$.

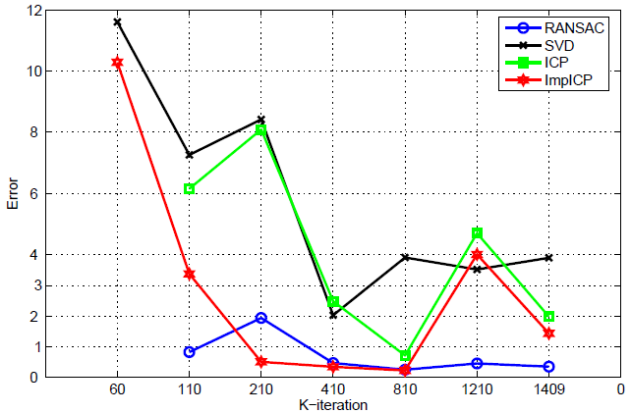



Fig. 12. Evaluation of the aligning methods.

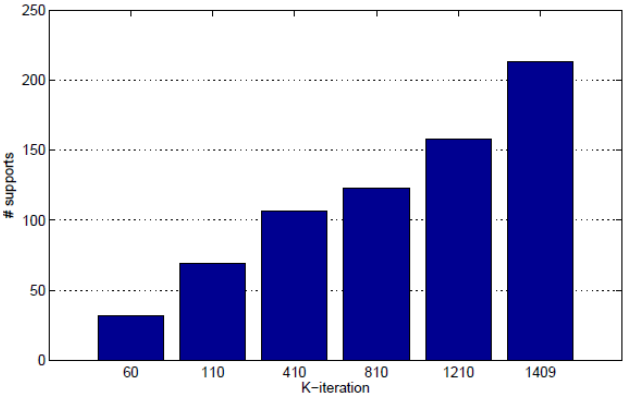



Fig. 13. Results obtained with the RANSAC algorithm. Number of supports.

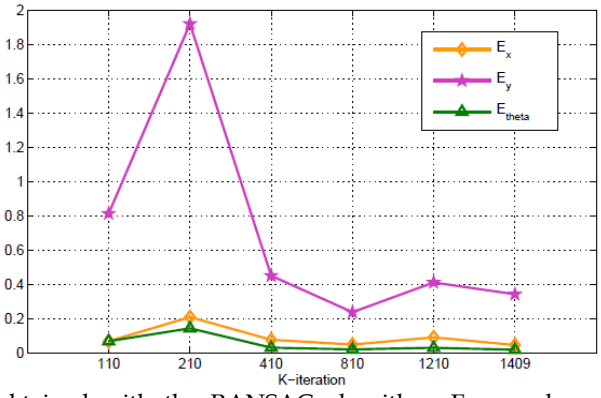
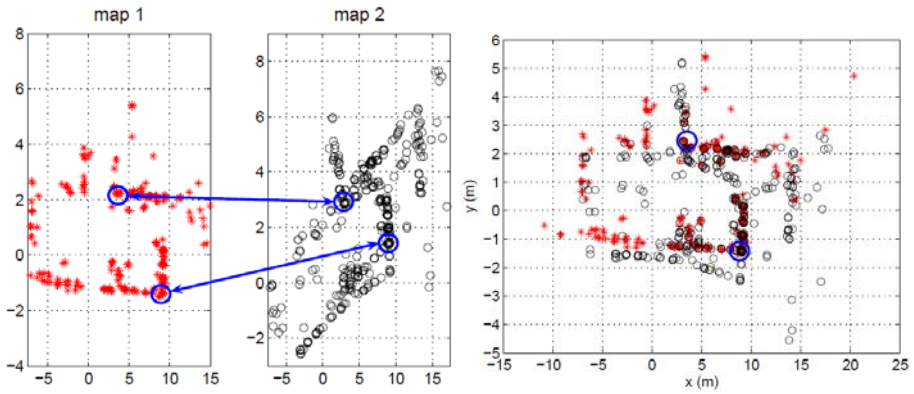



Fig. 14. Results obtained with the RANSAC algorithm. Error values for each aligning parameter.




(a) (b)

Fig. 15. Map alignment with real data. (a) Local maps before de alignment. Example of detected correspondences. (b) Map after the alignment.

#### 5.2.1 Multi-alignment results.

Table 1 presents an example of the results obtained with the **fsolve** function (Section 3.1). This table shows the aligning results obtained in the group of four robots, where $T^i_j$ represents the alignment between robot i and robot j. On the top part of the table, we can observe the aligning results between each pair of robots. These alignment parameters ($t_x$,$t_y$ and $\theta$) have been computed by means of the RANSAC algorithm described in Section 4.1. These solutions are valid between pairs of maps but may not be consistent globally. Then, on the bottom of the table, the alignment parameters ($t'_x$,$t'_y$ and $\theta'$) have been obtained with the fsolve function. In this case, the constraints imposed (see expressions E1 to E$\lambda$ of Section 3.1) optimize the solution so that it is globally consistent.

| | $T^1_2$ | $T^2_3$ | $T^3_4$ | $T^4_1$ |
|---|---|---|---|---|
| tx | -0.0676 | 0.1174 | -0.0386 | 0.0547 |
| ty | -0.0636 | 0.0423 | 0.8602 | -0.8713 |
| θ | -0.0144 | -0.0063 | 0.0286 | -0.0248 |
| tx' | -0.0388 | 0.0677 | -0.0408 | 0.0774 |
| ty' | 0.0363 | -0.1209 | 0.9521 | -0.9220 |
| θ | 0.0079 | -0.0375 | 0.0534 | -0.0436 |

Table 1. Alignment parameters.

## 7. Conclusion

This work has been focussed on the alignment problem of visual landmark-based maps built by the robots. The scenario presented is that of a team of robots that start their navigation tasks from different positions and independently, i.e., without knowledge of other robots' positions or observations. These robots share a common area of a typical office building. The maps built with the FastSLAM algorithm are initially referred to the reference system of each robot, located at their starting positions. In this situation, we consider the possibility of merging these maps into a global map. However, in order to do that, the landmarks of these

maps should be expressed in the same reference system. This is the motivation for the study of the alignment problem. To do that we have perform a comparison of several algorithms in order to select the most suitable for this kind of maps. The experiments have been carried out using simulated data as well as real data captured by the robots. The maps built by the robots are 3D maps. Nevertheless the alignment is performed in 2D, since the movement of the robots is performed in a 2D plane.
The next step is the study the second stage: map merging, i.e., fusing all the data into a global map. In this case, the uncertainty in the estimate of the landmarks should be considered. The map fusion problem has to be conceived as integrated into the FastSLAM problem, in which each robot pose is represented by a set of particles and each of them have a different estimate of the map. With this idea, observations coming from other robots should be treated different in terms of uncertainty. Furthermore, some questions are still open such as: when do we fuse the maps, how do we use the global map after the fusion is performed, etc. These ideas will be consider as future work.

## 8. 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 the Generalitat Valenciana under grant BFPI/2007/096.

## 9. References

Arun, K.S. ; Huang, T. S. & Blostein, S.D. (1987). Least square fitting of two 3d sets. *IEEE Transactions on Patterns Analysis and Machine Intelligence.* Vol. PAMI-9, No. 5, pp. 698-700, ISSN:0162-8828.

Ballesta, M. ; Gil, A. ; Martinez Mozos, O. & Reinoso, O. (2007). Local descriptors for visual SLAM. *Workshop on Robotics and Mathematics (ROBOMAT07)*, Portugal.

Bay, H. ; Tuytelaars, T. & Van Gool, L. (2006). SURF: Speeded-up robust features. *Proceedings of the 9th European Conference on Computer Vision,* 2006, pp. 404-417.

Besl, P.J. & Mckay, N. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Patterns Analysis and Machine Intelligence.* Vol. PAMI-14, No. 2, pp. 239-256.

Coleman, T.F. (1994). On the convergence of reflective newton methods for largerscale nonlinear minimization subject to bounds. *Mathematical Programming*. Vol. 67, No. 2, pp. 189-224.

Coleman, T.F. (1996). An interior, trust region approach for nonlinear minimization subject ot bounds. *SIAM Journal on Optimization*.

Fenwick, J.W. ; Newman, P.N. & Leornard, J. (2002). Cooperative concurrent mapping and localization. *Proceedings of the IEEE International Conference on Intelligent Robotics and Automation,* pp. 1810-1817, ISBN: 0-7803-7272-7, Washington, DC, USA, May 2002.

Gil, A. ; Reinoso, O.; Payá, L. & Ballesta, M. (2007). Influencia de los parámetros de un filtro de partículas en la solución al problema de SLAM. *IEEE Latin America*, Vol.6, No.1, pp.18-27, ISSN: 1548-0992.

Gil, A. ; Martinez Mozos, O. ; Ballesta, M. & Reinoso, O. (2009). A comparative evaluation of interest point detectors and local descriptors for visual slam. Ed. Springer Berlin / Heidelberg, ISSN: 0932-8092, pp. 1432-1769.

Howard, A. (2006). Multi-robot Simultanous Localization and Mapping using particle filters. *International Journal of Robotics Research,* Vol. 5, No. 12, pp. 1243-1256.

Kwak, N. ; Kim, G-W. ; Ji, S-H. & Bee, B-H. (2008). A mobile robot exploration strategy with low cost sonar and tungsten-halonen structural light. *Journal of Intelligent and Robotic Systems,* 5(1) : 89-111.

Konolige, K. ; Fox, D. ; Limketkai, B. ; Ko, J. & Stewart, B. (2003). Map merging for distributed robot navigation. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems.*

Leonard, J.J. & Durrant-Whyte, H.F. (1991). Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation,* Vol. 7, No. 3, pp. 376-382, ISSN 1042-296X.

Martínez Mozos, O. ; Gil, A. ; Ballesta, M. & Reinoso, O. (2007). Interest point detectors for visual SLAM. *Proceedings of the XII Conference of the Spanish Association ofr Artificial Intelligence (CAEPIA),* Salamanca, Spain.

Montemerlo, M.; Thrun, S.; Koller, D. & Wegbreit, B. (2002). FastSLAM: A factored solution to simultaneous localization and mapping. *Proceedings of the National Conference on Artificial Intelligence (AAAI).* Edmonton, Canada, pp. 593-598.

Moutalier, P. & Chatila, R. (1989). An experimental system for incremental environment modeling by an autonomous mobile robot. *1st International Symposium on Experimental Robotics*. Montreal, Vol. 139/1990, ISSN 0170-8643.

Rieger, J. (1987). On the classification of views of piecewise smooth objects. *Image and Vision Computing.* Vol. 5, No. 2 (May 1987), pp. 91-97.

Se, S. ; Lowe, D. & Little, J.J. (2005). Vision-based global localization and mapping for mobile robots. *IEEE Transactions on robotics.* Vol. 21, No. 3, pp. 364-375, ISSN 1552-3098 .

Stewart, B. ; Ko, J. ; Fox, D. & Konolige, K. (2003). A hierarchical bayesian approach to mobile robot map structure estimation. *Proceedings of the Conference on Uncertainty in AI (UAI).*

Thrun, S. (2001). A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research.* 20(5), pp. 335-363.

Thrun, S. & Liu,Y. (2004). Simultaneous localization and Mapping with sparse extended information filters. *International Journal of Robotics Research.* Vol. 23, No.7-8, pp. 693-716 (2004), ISSN 0278-3649.

Triebel, R. & Burgard, W. (2005). Improving simultaneous mapping and localization. *Proceedings of the National Conference on Artificial intelligence (AAAI).*

Valls Miro, J. ; Zhou, W. & Dissanayake, G. (2006). Towards vision based navigation in large indoor environments. *IEEE/RSJ Int. Conference on Intelligent Robots & Systems.*

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

Wijk, O. & Christensen, H.I. (2000). Localization and navigation of a mobile robot using natural point landmark extracted from sonar data. *Robotics and Autonomous Systems*. 1(31), pp. 31-42.

Zhang, Z. (1992). On local matching of free-form curves. *Proceedings of BMVC.* Pp. 347-356.

Zhou, X .S. & Roumeliotis, S.I. (2006). Multi-robot slam with unknown initial correspondence: The robot rendezvous case. *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems,* Beijing, China, pp. 1785-1792, 2006.