

# Workshop of Physical Agents 2009

Joaquín López

Editor of the special issue on Waf'08

E-mail: Joaquin@uvigo.es

**Abstract**—The Workshop of Physical Agents intends to be a forum for information and experience exchange in different areas regarding the concept of agent on physical environments, especially applied to the control and coordination of autonomous systems: robots, mobile robots, industrial processes or complex systems. This special issue is devoted to the selected papers presented at the WAF'08 that took place from September 10th to 13th in the city of Vigo (SPAIN).

**Index Terms**—Waf, physical agents

## THE WORKSHOP OF PHYSICAL AGENTS, WAF'2008

**T**HIS issue of the Journal of Physical Agents (JoPhA) is devoted to the selected papers presented at the Workshop of Physical Agents (WAF) 2008 that took place from September 10th to 13th in the city of Vigo (SPAIN). For those that are not familiar with the workshop, allow me to explain a little bit about the WAF in general and the IX edition in particular.

The Workshop of Physical Agents intends to be a forum for information and experience exchange in different areas regarding the concept of agent on physical environments, especially applied to the control and coordination of autonomous systems: robots, mobile robots, industrial processes or complex systems. This meeting is a forum of discussion for International and Spanish Research Groups of associated areas related to the Autonomous Control Systems and the Artificial Intelligence applied to robotics. We had contributions describing scientific works in different areas that use agent-based technologies concepts, robotics, vision, (software agents) highlighting the similarities and synergies among physical and software agents. We understand that physical agent applications are mainly oriented to the control, coordination and planning of autonomous systems: industrial or mobile robots, industrial processes, domotics and complex systems.

The IX edition was once again a success. We started September 11th enjoying the plenary talk “Probabilistic Techniques for Robot Navigation” given by Professor Wolfram Burgard from the University of Freiburg, R.F.A). After that, during the rest of the day we had working sessions on Localization, Robots and Vision with some breaks to take a look at the posters and enjoy the different demos. Finally, on September 12th we enjoyed the plenary talk “Social Robots” given by Professor Miguel Angel Salichs from the University of Carlos III, Spain. After that, we had a working session on Applications, a working session on Agents and, after lunch we had more demos. Talks on the different working sessions can be seen from the UVigoTV web page (<http://www.tv.uvigo.es/>).

Joaquín López is with the department of Ingeniería de Sistemas y Automática, Universidad de Vigo. Campus de Lagoas-Marcosende 36200 Vigo, PO. SPAIN.

As a summary, we had sixteen papers presented on the different sessions and nine posters.

As in former editions of this workshop, the live demos and exhibitions brought significant media press such as TV and newspapers to the event. We had a chance to see an industrial robot brought by FANUC, challenging conference attendees to “work” faster than him. We could also interact with mobile social robots such as RATO from the University of Vigo and watch robots that play soccer from the University of Rey Juan Carlos and Alicante, surveillance robots, humanoids, etc. In this publication we include six papers about different topics of SLAM, agents and applications. Even though Localization and SLAM has extensively been studied and some solutions have been presented, it is still a challenging research topic and we included the first three papers of this issue to further analyze it.

The first one “MSISpIC: A Probabilistic Scan Matching Algorithm Using a Mechanical Scanned Imaging Sonar” compares two well known scan matching algorithms: the MbICP and the pIC and propose the MSISpIC, a probabilistic scan matching algorithm for the localization of an Autonomous Underwater Vehicle (AUV). The second one “SLAM and Map Merging” presents a multi-robot mapping and localization system. The approach uses a team of mobile robots which use scan-matching and fast-slam techniques for mapping. The third paper on this topic “Localization through omnivision for a tour-guide robot” presents an algorithm that solves the global localization based on a merit function that ranks different possible poses obtained from an acquired image. It uses an iterative process for the minimization of that function using a particle filter.

The next two papers are focused on applications. The first one “Pedestrian tracking using inertial sensors” proposes a personal navigation system based on the use of inertial sensors. The system is intended to be a compliment for the Global Positioning System in places where GPS signals are not available. The second one “Developing a Low-Cost Autonomous Indoor Blimp” describes the design of an autonomous blimp-based robot and its navigation system. The robot is based on a commercial kit and its dimensions were suitable for use in indoor environments.

The last paper of this issue “A Cognitive Approach to Multimodal Attention” presents a novel cognitive architecture which enables integrated and efficient filtering of multiple modality sensory information. The proposed attention mechanism is based on contexts that determine what data is relevant to the current situation.

And finally, I would like to thank all the people that collaborated with the journal and I would like to ask for

your input regarding different aspects to improve it further. welcome.  
Any thoughts, recommendations, or ideas will be more than

# MSISpIC: A Probabilistic Scan Matching Algorithm Using a Mechanical Scanned Imaging Sonar

Emili Hernández, Pere Ridao, David Ribas and Joan Batlle

**Abstract**—This paper compares two well known scan matching algorithms: the MbICP and the pIC. As a result of the study, it is proposed the MSISpIC, a probabilistic scan matching algorithm for the localization of an Autonomous Underwater Vehicle (AUV). The technique uses range scans gathered with a Mechanical Scanning Imaging Sonar (MSIS), and the robot displacement estimated through dead-reckoning with the help of a Doppler Velocity Log (DVL) and a Motion Reference Unit (MRU). The proposed method is an extension of the pIC algorithm. Its major contribution consists in: 1) using an EKF to estimate the local path traveled by the robot while grabbing the scan as well as its uncertainty and 2) proposing a method to group into a unique scan, with a convenient uncertainty model, all the data grabbed along the path described by the robot. The algorithm has been tested on an AUV guided along a 600m path within a marina environment with satisfactory results.

**Index Terms**—Autonomous robots.

## I. INTRODUCTION

**D**URING a long term mission with an autonomous robot it is necessary to keep the track of the vehicle's position. Scan matching is a technique that can be used to estimate the vehicle displacement using successive range scans. Many applications in robotics like mapping, localization or tracking use this technique to estimate the robot's relative displacement [1], [2], [3] (to mention some but a few). Scan Matching estimates the robot relative displacement between two configurations, by maximizing the overlap between the range scans normally gathered with a laser or a sonar sensor. Moreover, the scans can then be mapped to set up a local map of the surrounding environment of the robot to be used for reactive/deliberative obstacle avoidance like in [4].

The existing scan matching techniques can be divided in two groups depending on if they use high-level entities like lines or planes or otherwise they rely on the raw scan. On one hand, it is possible to assume the existence of polygonal structures in structured environments as is supposed in [5] [6] [7], or even in some underwater applications [8]. However, extracting simple polygons for representing the environment is not always possible, particularly in unstructured scenarios which are the most common in underwater robotics. On the other hand, there is a second type of algorithms that work with raw data from the scanner to solve the matching. Usually, these techniques are based on a two step iterative process which is repeated till convergence. The sensor displacements are computed by approximating the solution to the best overlap between two scans by looking for the closest point

for each single data of the scan. After that, a minimization process to estimate the solution is done. The process is repeated until convergence. The most popular technique is the ICP (Iterative Closest Point) algorithm [9] which has been modified in different ways [10]. Most of the versions of the ICP algorithm use the Euclidian distance to estimate the correspondences between scans. However, this distance does not take into account that small rotations of the sensor mean large displacements as the distance is increased. To overcome this limitation several approaches have been done. The IDC (Iterative Dual Correspondence) algorithm [11] computes two different minimizations, one for the translation and another for the rotation. The MbICP (Metric-based Iterative Closest Point) algorithm [12] establishes a new distance concept which captures the sensor displacement and rotation at the same time. However, none of these algorithms take into account the sensor nor the displacement uncertainties which are very important, specially when sonar sensors are used. The probabilistic Iterative Correspondence method (pIC), proposed in [13], explicitly deals with those uncertainties to decide which points in the reference scan are statistically compatible with a certain point of the new scan. Moreover, the minimization process is carried out over the Mahalanobis distance. Although in this case the algorithm was applied to laser scans, its probabilistic formulation is able to improve the robustness and accuracy. On the other hand, in [14] another probabilistic approach is proposed in order to deal with the noisy and sparse measurements obtained from conventional time-of-fly sonar sensors of a mobile robot. This method deals with the sparsity of readings by grouping sonar measurements along short trajectories. It uses probabilistic models of ultrasonic and odometry sensors as well as a method to propagate the error through them in order to estimate a group of scan positions together with their uncertainty.

This work presents a study where the MbICP and pIC algorithms are compared to deal with acoustic data. Then we propose the MSISpIC as an extension of the basic pIC algorithm to deal with data gathered by an AUV using a MSIS. In our approach, the robot moves while scanning the environment. Hence, an EKF using a constant velocity model with acceleration noise, updated with velocity and attitude measurements obtained from a DVL and a MRU respectively, is used to estimate the trajectory followed by the robot along the scan. This trajectory is used to remove the motion induced distortion of the acoustic image as well as to predict the uncertainty of the range scans prior to register them through the standard pIC algorithm.

The paper is structured as follows. In section II it is described the metric scan matching using the MbICP algorithm.

The authors are with the Dept. Computer Engineering, University of Girona, 17071 Girona, Spain.  
E-mail: {emilihb,pere,drivas,jbatlle}@eia.udg.edu

Section III shows the probabilistic scan matching algorithm. Next, in section IV the two algorithms are compared using synthetic noisy data. Section V details the process to grab complete scans from the MSIS to be used in our scan matching algorithm which is described in section VI. Section VII reports the experimental results using the MSISpIC before conclusions.

## II. METRIC SCAN MATCHING

In the conventional ICP algorithm correspondences between two scans are chosen based on the closest-point rule, normally using the Euclidean distance. As pointed out in [13], this distance does not take into account that the points in the new scan, which are far from the sensor, could be far from their correspondences in the previous scan due to a rotation. To overcome this limitation, the MbICP algorithm [12] proposes a new concept of distance using the translation and rotation simultaneously. Introducing this new distance into the ICP framework, translational and rotational movements are taken into account at once while looking for the correspondences as well as during the minimization process.

Given a certain displacement vector  $\hat{\mathbf{q}} = (\hat{x}, \hat{y}, \hat{\theta})$  where  $\hat{\theta} \in [-\pi, \pi]$ , a norm function can be defined as:

$$\|\hat{\mathbf{q}}\| = \sqrt{\hat{x}^2 + \hat{y}^2 + L^2\hat{\theta}^2}$$

where  $L$  is a positive real number.

Then, given two points,  $\mathbf{p}_1 = (p_{1_x}, p_{1_y})$  and  $\mathbf{p}_2 = (p_{2_x}, p_{2_y})$  in  $\mathbb{R}^2$ , a distance function can be defined as follows:

$$d(\mathbf{p}_1, \mathbf{p}_2) = \|\arg \min_{\hat{\mathbf{q}}} \|\mathbf{p}_2 - \hat{\mathbf{q}} \oplus \mathbf{p}_1\|$$

As a distance it accomplishes the following properties:

- 1) Positiveness:  $d(\mathbf{p}_1, \mathbf{p}_2) \geq 0$
- 2) Symmetry:  $d(\mathbf{p}_1, \mathbf{p}_2) = d(\mathbf{p}_2, \mathbf{p}_1)$
- 3) Triangular inequality:  $d(\mathbf{p}_1, \mathbf{p}_3) \leq d(\mathbf{p}_1, \mathbf{p}_2) + d(\mathbf{p}_2, \mathbf{p}_3)$
- 4)  $d(\mathbf{p}_1, \mathbf{p}_2) = 0 \Rightarrow \mathbf{p}_1 = \mathbf{p}_2$

This distance is used in the ICP framework becoming the basis of the MbICP algorithm (see Algorithm 1). The inputs are the reference scan  $S_{ref}$  with points  $\mathbf{r}_i$  ( $i = 1..n$ ), the new scan  $S_{new}$  with points  $\mathbf{n}_j$  ( $j = 1..m$ ) and the initial relative displacement estimation  $\hat{\mathbf{q}}$ . The following procedure is iteratively executed until convergence. First, the points of the new scan ( $\mathbf{n}_j$ ) are compounded with the robot displacement ( $\hat{\mathbf{q}}$ ) as follows:

$$\mathbf{c}_j = \hat{\mathbf{q}} \oplus \mathbf{n}_j = \begin{pmatrix} \hat{x} + n_{j_x} \cos \hat{\theta} - n_{j_y} \sin \hat{\theta} \\ \hat{y} + n_{j_x} \sin \hat{\theta} + n_{j_y} \cos \hat{\theta} \end{pmatrix} \quad (1)$$

Assuming that the rotation between both scans is small enough, the following simplification is applied:

$$\sin \hat{\theta} \approx \hat{\theta}; \quad \cos \hat{\theta} \approx 1$$

obtaining a linear approximation of the eq. 1:

$$\mathbf{c}_j = \hat{\mathbf{q}} \oplus' \mathbf{n}_j = \begin{pmatrix} \hat{x} + n_{j_x} - \hat{\theta}n_{j_y} \\ \hat{y} + \hat{\theta}n_{j_x} + n_{j_y} \end{pmatrix} \quad (2)$$

The result ( $\mathbf{c}_j$ ), are the points of the new scan referenced to the reference frame. Then, for each point ( $\mathbf{c}_j$ ) it is possible to compute its association point ( $\mathbf{a}_j$ ) using the Closest-Point Rule because as stated in eq. 2, the relation between  $\mathbf{n}_j$  and  $\mathbf{c}_j$  is a linear function of  $\theta$  and it can be proved (see [12] for details) that the square of their norm is a quadratic equation in  $\theta$ . Solving the equation to find the  $\theta$  that minimizes the norm allows to find the distance function used in the MbICP algorithm:

$$d(\mathbf{r}_i, \mathbf{c}_j) = \sqrt{\delta_x^2 + \delta_y^2 - \frac{(\delta_x r_{i_y} - \delta_y r_{i_x})^2}{r_{i_y}^2 + r_{i_x}^2 + L^2}} \quad (3)$$

where

$$\begin{aligned} \delta_x &= c_{j_x} - r_{i_x} \\ \delta_y &= c_{j_y} - r_{i_y} \end{aligned}$$

Once the correspondences  $\mathbf{a}_j$  have been found, it is possible to estimate the displacement  $\hat{\mathbf{q}}_{min}$  which minimizes the mean square error between  $\mathbf{a}_j$  and  $\mathbf{c}_j$  using the eq. 3 distance function. This is done using Least Squares minimization (eq. 4). If there is convergence, the function returns, otherwise another iteration is required.

$$q_{min} = \arg \min_{\mathbf{q}} \left\{ \sum_{j=1}^n \left( \delta_{j_x}^2 + \delta_{j_y}^2 - \frac{(\delta_{j_x} a_{j_y} - \delta_{j_y} a_{j_x})^2}{a_{j_y}^2 + a_{j_x}^2 + L^2} \right) \right\} \quad (4)$$

where

$$\begin{aligned} \delta_{j_x} &= c_{j_x} - c_{j_y}\theta + x - a_{j_x} \\ \delta_{j_y} &= c_{j_x}\theta + c_{j_y} + y - a_{j_y} \end{aligned}$$

---

### Algorithm 1 The MbICP algorithm

---

```

 $\hat{\mathbf{q}}_{MbICP} = MbICP(S_{ref}, S_{new}, \hat{\mathbf{q}}) \{$ 
   $k = 0$ 
   $\hat{\mathbf{q}}_k = \hat{\mathbf{q}}$ 
  do  $\{$ 
    for  $(j = 0; size(S_{new}); j + +)$ 
       $\mathbf{a}_j = \arg \min_{\mathbf{r}_i} \{d(\mathbf{r}_i, \hat{\mathbf{q}}_k \oplus \mathbf{n}_j)\}, \forall \mathbf{r}_i \in S_{ref} (eq. 3)$ 
       $\hat{\mathbf{q}}_{min} = \arg \min_{\mathbf{q}} \left\{ \sum_j ((\mathbf{a}_j - \mathbf{q} \oplus \mathbf{n}_j)^2) \right\}$ 
      if  $(Convergence())$ 
         $\hat{\mathbf{q}}_{MbICP} = \hat{\mathbf{q}}_{min}$ 
      else  $\{$ 
         $\hat{\mathbf{q}}_{k+1} = \hat{\mathbf{q}}_{min}$ 
         $k + +$ 
       $\}$ 
     $\}$ 
  while  $(!Convergence() \text{ and } k < maxIterations)$ 
 $\}$ 

```

---

## III. PROBABILISTIC SCAN MATCHING

Despite the increase of accuracy and robustness of the MbICP algorithm, it does not model the uncertainty of the sensor measurements. Because of that, if the scan data is very noisy, two statistically compatible points could appear far enough, in terms of the Euclidean distance. This situation

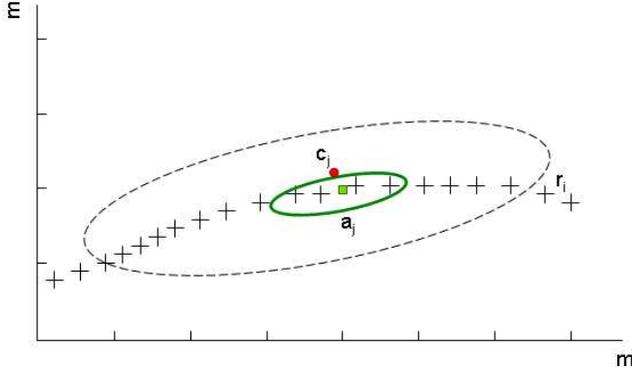


Fig. 1. pIC correspondence computation. The large ellipse contains all the statistically compatible points and the squared point represents the correspondence with its uncertainty (small ellipse).

might prevent a possible association or even generate a wrong one. The pIC algorithm proposed in [13] is a statistical extension of the ICP algorithm where the relative displacement  $\mathbf{q}$  as well as the observed points in both scans  $\mathbf{r}_i$  and  $\mathbf{n}_j$ , are modeled as random Gaussian variables (r.g.v.). Whereas geometric ICP algorithm uses the closest-point rule to find the correspondence for a point in the new scan, the pIC algorithm first computes the set of compatible points (in terms of the Mahalanobis distance) and then computes the virtual 'expected' compatible point to be used as the correspondence (Fig. 1).

For a better understanding the algorithm is reproduced here (Algorithm 2). The inputs are the reference scan  $S_{ref}$  with points  $\mathbf{r}_i$  ( $i = 1..n$ ), the new scan  $S_{new}$  with points  $\mathbf{n}_j$  ( $j = 1..m$ ) and the initial relative displacement estimation  $\hat{\mathbf{q}}$  with its covariance  $\mathbf{P}_q$ . The following procedure is iteratively executed until convergence. First, the points of the new scan ( $\mathbf{n}_j$ ) are compounded with the robot displacement ( $\mathbf{q}_k$ ). The result ( $\mathbf{c}_j$ ), are the points of the new scan referenced to the reference frame. Then, for each point ( $\mathbf{c}_j$ ), a set ( $A_j$ ) of all the compatible points in the reference scan ( $S_{ref}$ ) is established using a compatibility test over the Mahalanobis distance. Next step consists in computing the virtual association point ( $\mathbf{a}_j$ ) as the expectancy over the random variable defined by the set ( $A_j$ ). It is worth noting that if  $\mathbf{q} \equiv N(\hat{\mathbf{q}}, \mathbf{P}_q)$ ,  $\mathbf{n}_j \equiv N(\hat{\mathbf{n}}_j, \mathbf{P}_{n_j})$  and  $\mathbf{r}_i \equiv N(\hat{\mathbf{r}}_i, \mathbf{P}_{r_i})$ , it is possible to compute the probability  $p(\mathbf{r}_i = \mathbf{c}_j)$  for every element ( $\mathbf{r}_i$ ) of  $A_j$  of being a correct matching for  $\mathbf{c}_j$ . In order to do so, let us define a r.g.v. which describes the error of the  $\{\mathbf{r}_i, \mathbf{c}_j\}$  pairing:

$$\mathbf{e}_{ij} = \mathbf{r}_i - \mathbf{q}_k \oplus \mathbf{n}_j ; \mathbf{e}_{ij} \approx N(\hat{\mathbf{r}}_i - \hat{\mathbf{q}}_k \oplus \hat{\mathbf{n}}_j, \mathbf{P}_{e_{ij}})$$

$$\mathbf{P}_{e_{ij}} = \mathbf{P}_{r_i} + \mathbf{J}_q \mathbf{P}_q \mathbf{J}_q^T + \mathbf{J}_n \mathbf{P}_{n_j} \mathbf{J}_n^T$$

with

$$\mathbf{J}_q = \left. \frac{\partial \mathbf{r}_i - \mathbf{q} \oplus \mathbf{n}_j}{\partial \mathbf{q}} \right|_{\hat{\mathbf{q}}}, \mathbf{J}_n = \left. \frac{\partial \mathbf{r}_i - \mathbf{q} \oplus \mathbf{n}_j}{\partial \mathbf{n}_j} \right|_{\hat{\mathbf{n}}_j}$$

then  $p(\mathbf{r}_i = \mathbf{c}_j) = p(\mathbf{e}_{ij} = 0)$  where  $p(\mathbf{e}_{ij} = 0)$  can be computed as follows:

$$p(\mathbf{e}_{ij} = 0) = \frac{f_{e_{ij}}(\mathbf{r}_i - \mathbf{c}_j)}{\sum_{\mathbf{r}_i \in A_j} f_{e_{ij}}(\mathbf{r}_i - \mathbf{c}_j)} \quad (5)$$

where  $f_{e_{ij}}$  is the probability density function of  $\mathbf{e}_{ij}$  r.g.v. Once  $\hat{\mathbf{a}}_j$  has been computed, a similar procedure can be used to estimate its uncertainty  $\mathbf{P}_{a_j}$ , before computing the uncertainty  $\mathbf{P}_{e_j}$  of the matching error ( $\hat{\mathbf{a}}_j - \hat{\mathbf{c}}_j$ ). Now, it is possible to estimate the displacement  $\hat{\mathbf{q}}_{min}$  which minimizes the mean square error of the Mahalanobis Distance [15] between  $\hat{\mathbf{a}}_j$  and  $\hat{\mathbf{c}}_j$ . This is done using Least Squares minimization method. If there is convergence, the function returns, otherwise another iteration is required.

---

#### Algorithm 2 The pIC algorithm

---

```

 $\hat{\mathbf{q}}_{pIC} = pIC(S_{ref}, S_{new}, \hat{\mathbf{q}}, \mathbf{P}_q) \{$ 
   $k = 0$ 
   $\hat{\mathbf{q}}_k = \hat{\mathbf{q}}$ 
  do {
    for( $j = 0; size(S_{new}); j++$ ) {
       $\hat{\mathbf{c}}_j = \hat{\mathbf{q}}_k \oplus \hat{\mathbf{n}}_j$ 
       $A_j = \{\mathbf{r}_i \in S_{ref} / D_M^2(\mathbf{r}_i, \mathbf{c}_j) \leq \chi_{2,\alpha}^2\}$ 
       $\hat{\mathbf{a}}_j = \sum_j \hat{\mathbf{r}}_i p(\mathbf{r}_i = \mathbf{c}_j), \forall \mathbf{r}_i \in A_j$ 
       $\mathbf{P}_{a_j} = \sum_{\mathbf{r}_i \in A_j} [(\hat{\mathbf{r}}_i - \hat{\mathbf{a}}_j)(\hat{\mathbf{r}}_i - \hat{\mathbf{a}}_j)^T + \mathbf{P}_{r_i}] p(\mathbf{r}_i = \mathbf{c}_j)$ 
       $\mathbf{P}_{e_j} = \mathbf{P}_{a_j} + \mathbf{J}_q \mathbf{P}_q \mathbf{J}_q^T + \mathbf{J}_n \mathbf{P}_{n_j} \mathbf{J}_n^T$ 
    }
     $\hat{\mathbf{q}}_{min} = arg \min_{\mathbf{q}} \left\{ \sum_j ((\hat{\mathbf{a}}_j - \hat{\mathbf{c}}_j)^T \mathbf{P}_{e_j}^{-1} (\hat{\mathbf{a}}_j - \hat{\mathbf{c}}_j)) \right\}$ 
    if( $Convergence()$ )
       $\hat{\mathbf{q}}_{pIC} = \hat{\mathbf{q}}_{min}$ 
    else {
       $\hat{\mathbf{q}}_{k+1} = \hat{\mathbf{q}}_{min}$ 
       $k++$ 
    }
  }
  while(! $Convergence()$  and  $k < maxIterations$ )
}

```

---

#### IV. COMPARISON BETWEEN MBICP AND PIC ALGORITHMS

In order to examine in detail the effectiveness of the MbICP and the pIC algorithms, several tests with synthetic data were carried out. Simulated data was generated with Matlab using two different state vectors  $\mathbf{q}_1$  and  $\mathbf{q}_2$  as an input for a script to generate a static scan of a rectangular scenario. The initial guess for the displacement  $\mathbf{q}_0$  was computed using the real displacement of the robot with Gaussian additive noise.

Two virtual scans, depicted in Fig. 2 without noise, have been used as a reference dataset to test the two techniques compared in this paper. In both cases, it is assumed that the robot was stopped when the scan was gathered. Hence, all the noise comes from the scanner sensor, so the scans are not distorted except for the used noise level, which is more or less the same as found in commercial MSIS sensors. This case is quite similar to the conventional case in which laser scan

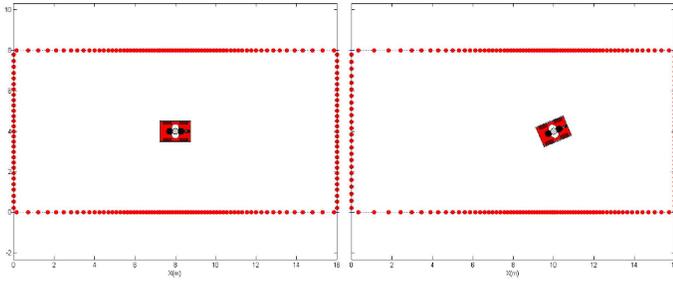


Fig. 2. Matlab virtual scans without noise: a) Scan k; b) Scan k+1.

Alg. \ Noise	$\sigma_r = 0.05$ $\sigma_\psi = 1.5$	$\sigma_r = 0.1$ $\sigma_\psi = 3$	$\sigma_r = 0.2$ $\sigma_\psi = 8$
MbICP	(1.83,0.68,25.06)	(2.0,25.78)	(1.94,-0.07,15.77)
pIC	(1.97,0.22.9)	(1.98,0.02,22.1)	(1.98,0.08,22.15)

TABLE I

DISPLACEMENT ESTIMATIONS. UNITS OF  $\sigma_r$  AND  $\sigma_\psi$  ARE METERS AND DEGREES RESPECTIVELY.

sensors are used. The displacement between the scans is  $\mathbf{q}_0 = (2m, 0m, 22.5^\circ)$ . At each execution, random noise is added to the range and bearing of each beam. In order to stabilize the results, each algorithm have been executed 50 times for each level of noise. Table I shows the means of the displacement estimations, Fig. 3 shows the mean of iterations needed to reach convergence for each noise step and Fig. 4 depicts the graphical solutions of a single representative execution.

According to Fig. 4, the MbICP algorithm is able to perform a good scan matching for scans with low noise. However, the results are not so good when the noise is increased. In Fig. 3, it can be appreciated that the number of iterations needed for the MbICP algorithm to reach convergence increases with respect to the applied uncertainty. This algorithm is computationally efficient because it only takes into account geometric information. However, it does not behave well in some situations where associations are not computed correctly, for example in

the corners (see Fig. 4 when  $\sigma_\psi = 3^\circ$  and  $\sigma_\psi = 8^\circ$ ). Hence, it is not a strong candidate to be used with uncertain sonar data.

In contrast, the pIC algorithm performs well in situations where the MbICP produces poor results. This technique does not present problems in associating data from corners because correspondences are virtual points created from statistically compatible points of the first scan. Thus, the algorithm is more robust and accurate (Table I). As shown in Fig. 3, the pIC algorithm requires fewer iterations to converge even when the noise is increased but the computational load of each iteration is higher. In addition, it also offers better results than the MbICP under the same conditions (Fig. 4). In conclusion, although the increased computational load due to the statistical nature of the algorithm, the pIC algorithm is a better candidate to be used with acoustic data. For this reason, the work that follows is based only on the pIC algorithm.

## V. SCAN GRABBING USING A MSIS

Scan matching techniques are conceived to accept as input parameters two range scans with a rough displacement estimation between them. Most of the algorithms use laser range finders which gather scans almost instantaneously. However, for the underwater environment, commercially available scan sensors are based on acoustics. Most of these sensors have a mechanical head that rotates at fixed angular steps. At each step, a beam is emitted and received a posteriori, measuring ranges to the obstacles found across its trajectory. Hence, getting a complete scan lasts few seconds while the vehicle is moving, generating deformed scans. Therefore, a correction taking into account the robot pose when the beam was grabbed is necessary.

### A. Beam segmentation and range detection

The MSIS returns a polar acoustic image composed of beams. Each beam has a particular bearing angle value and a set of intensity measurements. The angle corresponds to the orientation of the sensor head when the beam was emitted. The acoustic linear image corresponding to one beam is returned as an array of bins. A bin is an element of the array that represents an acoustic intensity detected at a certain distance with a value between 0 and 255 (Fig. 5). The beam is then segmented using a predefined threshold to compute the intensity peaks. Due to the noisy nature of the acoustic data, a minimum distance between peaks criteria is also applied. Hence, positions finally considered are those corresponding to high intensity values above the threshold with a minimum *distance* between each other. Fig. 6 illustrates this process.

### B. Relative vehicle localization

The pIC algorithm needs a complete scan to be registered with the previous one in order to estimate the robot's displacement. Since MSIS needs a considerable period of time to obtain a complete scan, if the robot does not remain static, the robot's motion induces a distortion in the acoustic image (Fig. 7). To deal with this problem it is necessary to know the

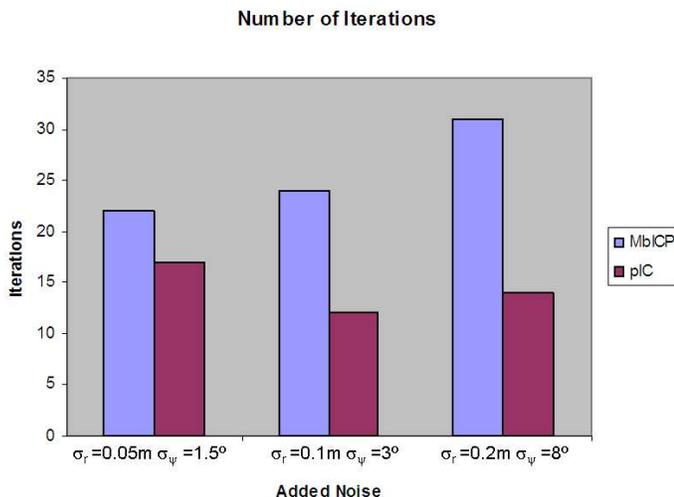


Fig. 3. Mean of iterations until convergence.

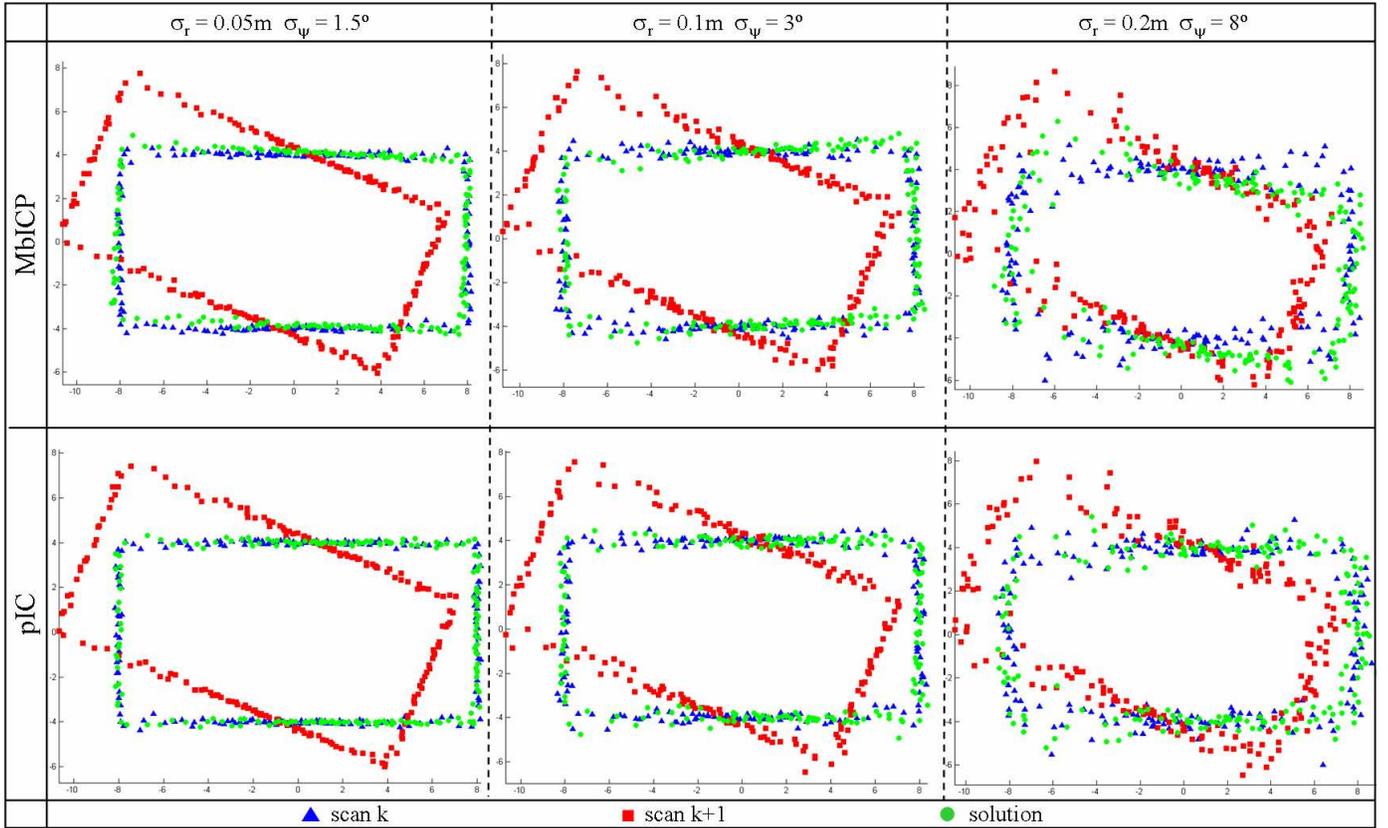


Fig. 4. MbICP and pIC results.

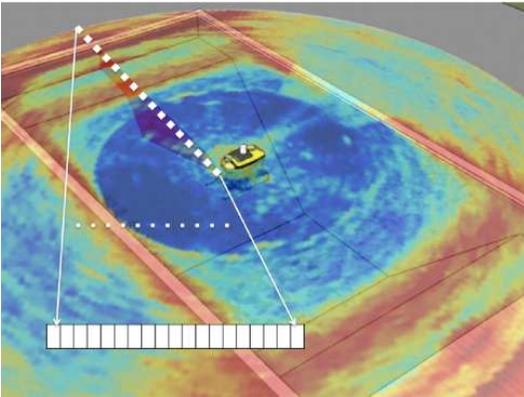


Fig. 5. Interpretation of a polar image gathered with an MSIS. The current beam is detailed.

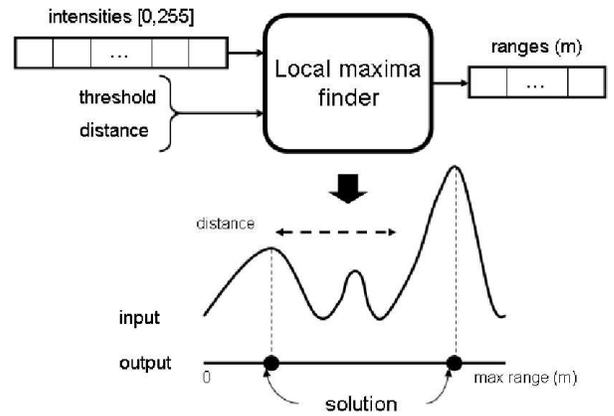


Fig. 6. Peaks detector for a MSIS beam.

robot's pose at the beam reception time. Hence, it is possible to define an initial coordinate system  $I$  to reference all the range measurements belonging to the same scan. In our case, this initial frame is fixed at the robot pose where the first beam of the current scan was read.

The localization system used in this work is a slight modification of the navigation system described in [16]. In this system, a Xsense MRU provides heading measurements and a SonTek Argonaut DVL unit which includes 2 inclinometers and a depth sensor is used to estimate the robot's pose during the scan (navigation problem). MSIS beams are read at 30

Hz while DVL and MTi readings arrive asynchronously at a frequency of 1.5 Hz and 10 Hz respectively. An EKF is used to estimate the robot's 6DOF pose whenever a sonar beam is read. DVL and MTi readings are used asynchronously to update the filter. To reduce the noise inherent to the DVL measurements, a simple 6DOF constant velocity kinematics model is used.

The information of the system at step  $k$  is stored in the state vector  $\mathbf{x}_k$  with estimated mean  $\hat{\mathbf{x}}_k$  and covariance  $\mathbf{P}_k$ :

$$\hat{\mathbf{x}}_k = [\hat{\eta}^B, \hat{v}^R]^T \quad \mathbf{P}_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \quad (6)$$

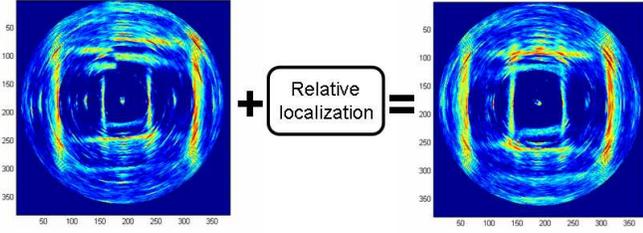


Fig. 7. Relative displacement correction.

with:

$$\eta^B = [x, y, z, \phi, \theta, \psi]^T; \quad \nu^R = [u, v, w, p, q, r]^T \quad (7)$$

where, as defined in [17],  $\eta^B$  is the position and attitude vector referenced to a base frame  $B$ , and  $\nu^R$  is the linear and angular velocity vector referenced to the robot's coordinate frame  $R$ . The coordinate frame  $B$  is chosen coincident with  $I$  but oriented to the north, hence the compass measurements can be integrated straight forward.

The vehicle's movement prediction is performed using the 6DOF kinematic model:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) = \begin{bmatrix} \eta_k^B \\ \nu_k^R \end{bmatrix} = \begin{bmatrix} \eta_{k-1}^B + J(\eta_{k-1}^B) \nu_{k-1}^R \\ \nu_{k-1}^R \end{bmatrix} \quad (8)$$

$$J(\eta) = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi & 0 & 0 & 0 \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - s\psi s\phi & 0 & 0 & 0 \\ -s\theta & c\theta s\phi & c\theta c\phi & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & s\phi t\theta & c\phi t\theta \\ 0 & 0 & 0 & 0 & c\phi & -s\phi \\ 0 & 0 & 0 & 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \quad (9)$$

Although in this model the velocity is considered to be constant, in order to allow for slight changes, a velocity perturbation modeled as the integral of a stationary white noise  $v_k$  is introduced. The covariance matrix  $\mathbf{Q}_k$  of this acceleration noise is diagonal and in the order of magnitude of the maximum acceleration increment that the robot may experience over a sample period.

$$\nu_k^R = \hat{\nu}_k^R + v_k T \quad (10)$$

$$E[v_k] = 0; \quad E[v_k v_j^T] = \delta_{kj} \mathbf{Q} \quad (11)$$

Hence,  $v_k$  is an acceleration noise which is integrated and is added in velocity (Equation 10), being nonlinearly propagated to the position. Finally, the model prediction and update is carried out as detailed below:

1) *Prediction*: The estimate of the state is obtained as:

$$\hat{\mathbf{x}}_k = f(\hat{\mathbf{x}}_{k-1}) \quad (12)$$

and its covariance matrix as:

$$\mathbf{P}_k = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \quad (13)$$

where  $\mathbf{F}_k$  and  $\mathbf{G}_k$  are the Jacobian matrices of partial derivatives of the non-linear model function  $f$  with respect to the state  $\mathbf{x}_{R,k}^B$  and the noise  $v_k$ , respectively.

2) *Update using DVL measurements*: The model prediction is updated by the standard Kalman filter equations each time a new DVL measurement arrives:

$$\mathbf{z}_{DVL,k} = [u_b, v_b, w_b, u_w, v_w, w_w, \phi_i, \theta_i, \psi_c, z_{depth}]^T \quad (14)$$

where subindex  $b$  stands for bottom tracking velocity,  $w$  for through water velocity,  $i$  for inclinometers and  $c$  represents the compass. The measurement model is:

$$\mathbf{z}_{DVL,k} = \mathbf{H}_{DVL,k} \mathbf{x}_{k|k-1} + w_k \quad (15)$$

$$\mathbf{H}_{DVL} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ 0 & 0 & 1 & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} \end{bmatrix} \quad (16)$$

where  $w_k$  (measurement noise) is a zero-mean white noise:

$$E[w_k] = 0; \quad E[w_k w_j^T] = \delta_{kj} \mathbf{R}_{DVL,k} \quad (17)$$

Since the DVL sensor provides a status measurement for bottom tracking and water velocity, depending on the quality of the measurements, different versions of the  $\mathbf{H}$  matrix are used to fuse one (removing row 2), the other (removing row 1), or both readings (using the full matrix).

3) *Update using MTi measurements*: Whenever a new attitude measurement is available from the MTi sensor, the model prediction is updated using the standard Kalman filter equations:

$$\mathbf{z}_{MTi,k} = [\phi, \theta, \psi]^T, \quad \mathbf{z}_{MTi,k} = \mathbf{H}_{MTi,k} \mathbf{x}_{k|k-1} + w_k \quad (18)$$

$$\mathbf{H}_{MTi} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 6} \end{bmatrix} \quad (19)$$

where  $w_k$  (measurement noise) is a zero-mean white noise:

$$E[w_k] = 0; \quad E[w_k w_j^T] = \delta_{kj} \mathbf{R}_{MTi,k} \quad (20)$$

### C. Scan forming

The navigation system presented above is able to estimate the robot's pose, but the uncertainty will grow without limit due to its dead-reckoning nature. Moreover, we are only interested in the robot's relative position (and uncertainty) with respect to the beginning of the scan ( $I$ ). Hence a slight modification to the filter is introduced making a reset in position (setting  $x, y, z$  to 0 in the vector state) whenever a new scan is started. Therefore, while the filter is working, the estimated position is always relative to the position where the first beam of the scan was gathered ( $I$ ). Note that it is important to keep the  $\psi$  value (it is not reset) because it represents an absolute angle with respect to the magnetic north and a reset would mean an unreal high rotation during the scan. The same thing happens with  $\phi$  and  $\theta$ . Since we are only interested in the uncertainty accumulated during the scan, the reset process also affects the  $x, y$ , and  $z$  terms of the covariance matrix  $\mathbf{P}$ . Now, the modified filter provides the robot's relative position where the beams were gathered including its uncertainty accumulated during the scan. Hence, it is possible to reference all the ranges computed from the beams to the initial frame  $I$ , removing the distortion induced by the robot's motion by using the following method.

Let

- $\rho \equiv N(\hat{\rho}, \mathbf{P}_\rho)$  be a r.g.v. corresponding to the polar measurement where  $\hat{\rho} = (\beta, r)$  is the observed measurement and  $\mathbf{P}_\rho$  its corresponding uncertainty.
- $\mathbf{x}_R^B \equiv N(\hat{\mathbf{x}}_R^B, \mathbf{P}_{BR})$  be a r.g.v. corresponding to the robot's uncertain position where the  $\rho$  beam was gathered. This value is estimated by the EKF and is represented in the northern referenced frame  $B$ .
- $\mathbf{x}_B^I \equiv N(\hat{\mathbf{x}}_B^I, \mathbf{P}_{IB})$  be a r.g.v. corresponding to the transformation needed to map  $B$  frame to  $I$  frame. In our particular case, this is a null translation followed by a rotation used to align  $B$  with  $I$ .
- $\mathbf{x}_S^R$  be a deterministic vector that describes the position and attitude of the sensor frame  $S$  with respect to the robot's frame  $R$ . Note that this is non-random rigid body transformation.

then, it is possible to compute the position (and uncertainty) of any observed point referenced to the initial frame  $I$  as follows:

- 1)  $\mathbf{p}^S = P2C(\rho) \Rightarrow \mathbf{p}^S = N(\underbrace{P2C(\hat{\rho})}_{\hat{\mathbf{p}}^S}, \underbrace{\mathbf{J}_S \mathbf{P}_\rho \mathbf{J}_S^T}_{\mathbf{P}_S})$   
 where  $P2C(\rho)$  turn polar into cartesian coordinates and  
 $\mathbf{J}_S = \left. \frac{\partial P2C(\rho)}{\partial \rho} \right|_{\hat{\rho}}$
- 2)  $\mathbf{p}^R = \mathbf{x}_S^R \oplus \mathbf{p}^S \Rightarrow$   
 $\mathbf{p}^R = N(\underbrace{\mathbf{x}_S^R \oplus \hat{\mathbf{p}}^S}_{\hat{\mathbf{p}}^R}, \underbrace{\mathbf{J}_{R\oplus} \mathbf{P}_S \mathbf{J}_{R\oplus}^T}_{\mathbf{P}_R})$   
 where  $\mathbf{J}_{R\oplus} = \left. \frac{\partial \mathbf{x}_S^R \oplus \mathbf{p}^S}{\partial \mathbf{p}^S} \right|_{\hat{\mathbf{p}}^S}$
- 3)  $\mathbf{p}^B = \mathbf{x}_R^B \oplus \mathbf{p}^R \Rightarrow$   
 $\mathbf{p}^B = N(\underbrace{\hat{\mathbf{x}}_R^B \oplus \hat{\mathbf{p}}^R}_{\hat{\mathbf{p}}^B}, \underbrace{\mathbf{J}_{B1\oplus} \mathbf{P}_{BR} \mathbf{J}_{B1\oplus}^T + \mathbf{J}_{B2\oplus} \mathbf{P}_R \mathbf{J}_{B2\oplus}^T}_{\mathbf{P}_B})$   
 where  $\mathbf{J}_{B1\oplus} = \left. \frac{\partial \mathbf{x}_R^B \oplus \mathbf{p}^R}{\partial \mathbf{x}_R^B} \right|_{\hat{\mathbf{x}}_R^B}$   
 and  $\mathbf{J}_{B2\oplus} = \left. \frac{\partial \mathbf{x}_R^B \oplus \mathbf{p}^R}{\partial \mathbf{p}^R} \right|_{\hat{\mathbf{p}}^R}$
- 4)  $\mathbf{p}^I = \mathbf{x}_B^I \oplus \mathbf{p}^B \Rightarrow$   
 $\mathbf{p}^I = N(\underbrace{\hat{\mathbf{x}}_B^I \oplus \hat{\mathbf{p}}^B}_{\hat{\mathbf{p}}^I}, \underbrace{\mathbf{J}_{I1\oplus} \mathbf{P}_{IB} \mathbf{J}_{I1\oplus}^T + \mathbf{J}_{I2\oplus} \mathbf{P}_B \mathbf{J}_{I2\oplus}^T}_{\mathbf{P}_I})$   
 where  $\mathbf{J}_{I1\oplus} = \left. \frac{\partial \mathbf{x}_B^I \oplus \mathbf{p}^B}{\partial \mathbf{x}_B^I} \right|_{\hat{\mathbf{x}}_B^I}$   
 and  $\mathbf{J}_{I2\oplus} = \left. \frac{\partial \mathbf{x}_B^I \oplus \mathbf{p}^B}{\partial \mathbf{p}^B} \right|_{\hat{\mathbf{p}}^B}$

First, the function  $P2C$  transforms the range and bearing data  $\rho = (\beta, r)^T$  from Polar space to Cartesian space. The result is the observed point  $\mathbf{p}^S$  referenced to the  $S$  frame. As stated,  $\mathbf{p}^S$  is a r.g.v. which mean ( $\hat{\mathbf{p}}^S$ ) and covariance ( $\mathbf{P}_S$ ) can be easily computed. Then, by means of a rigid body transformation, the point is referenced to the robot's frame  $R$ . Again, the new representation  $\mathbf{p}^R$  is a r.g.v. with mean  $\hat{\mathbf{p}}^R$  and covariance  $\mathbf{P}_R$ . Now, the robot's relative position  $\mathbf{x}_R^B$  computed with the EKF is compounded with the robot's referenced point  $\mathbf{p}^R$  to get the r.g.v.  $\mathbf{p}^B$  with mean  $\hat{\mathbf{p}}^B$  and covariance  $\mathbf{P}_B$ . Finally, the last compounding operation rotates the point to reference it to the initial frame  $I$ . As in

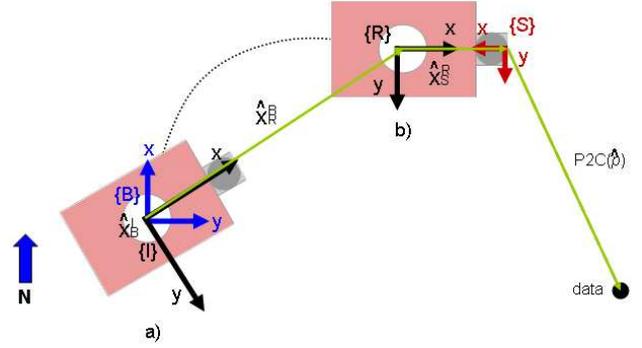


Fig. 8. Scan forming process. Point a represents the position of the robot at the first beam of the scan, point b) Represents it at the position of the beam k.

the previous cases,  $\mathbf{p}^I$  is a r.g.v. with a known mean ( $\hat{\mathbf{p}}^I$ ) and covariance  $\mathbf{P}_I$ . Fig. 8 illustrates this process while the scan grabbing process in algorithmic notation is described in Algorithm 3.

### Algorithm 3 Scan grabbing

---

```

[ $S_{new}, \hat{\mathbf{q}}_{new}, \mathbf{P}_{q_{new}}$ ] = ScanGrabbing() {
  ResetDeadReckoningXYZ()
  [ $\hat{\mathbf{x}}_B^I, \mathbf{P}_{IB}$ ] = GetDeadReckoning()
  for all beams {
    beam = GetBeam()
    beam = Segment(beam)
    [ $\hat{\rho}, \mathbf{P}_\rho$ ] = LocalMaximaFinder(beam)
    [ $\hat{\mathbf{x}}_R^B, \mathbf{P}_{BR}$ ] = GetDeadReckoning()
    //  $\hat{\rho}$  and  $\mathbf{P}_\rho$  from the local frame  $I$ 
     $\hat{\mathbf{n}} = \hat{\mathbf{x}}_B^I \oplus \hat{\mathbf{x}}_R^B \oplus \mathbf{x}_S^R \oplus P2C(\hat{\rho})$ 
     $\mathbf{P}_n = \mathbf{J}_{I1\oplus} \mathbf{P}_{IB} \mathbf{J}_{I1\oplus}^T + \mathbf{J}_{I2\oplus} [\mathbf{J}_{B1\oplus} \mathbf{P}_{BR} \mathbf{J}_{B1\oplus}^T + \mathbf{J}_{B2\oplus} \mathbf{J}_{R\oplus} \mathbf{J}_S \mathbf{P}_\rho \mathbf{J}_S^T \mathbf{J}_{R\oplus}^T \mathbf{J}_{B2\oplus}^T] \mathbf{J}_{I2\oplus}^T$ 
     $S_{new} = S_{new} \cup \{[\hat{\mathbf{n}}, \mathbf{P}_n]\}$ 
  }
   $\hat{\mathbf{q}}_{new} = \hat{\mathbf{x}}_R^B$ 
   $\mathbf{P}_{q_{new}} = \mathbf{P}_{BR}$ 
}

```

---

## VI. THE MSISPIC ALGORITHM

Once the pIC and the ScanGrabbing algorithms have been setup, it is very simple to localize the robot. This is the purpose of the MSISPIC algorithm (see Algorithm 4), which iteratively grabs two scans and register them using the pIC algorithm. It is worth noting that the pIC takes as input two consecutive scans ( $S_{new}$  and  $S_{ref}$ ) and its relative displacement which coincides with the position occupied by the robot at the end of the first scan ( $\hat{\mathbf{q}}_{ref}$ ). The output is an improved estimation of the robot displacement ( $\hat{\mathbf{q}}_{new}$ ). The iterative compounding of the relative displacement allows to track the global robot position.

## VII. EXPERIMENTAL RESULTS

The MSISPIC algorithm has been used with a dataset obtained in an abandoned marina located in Sant Pere Pescador,

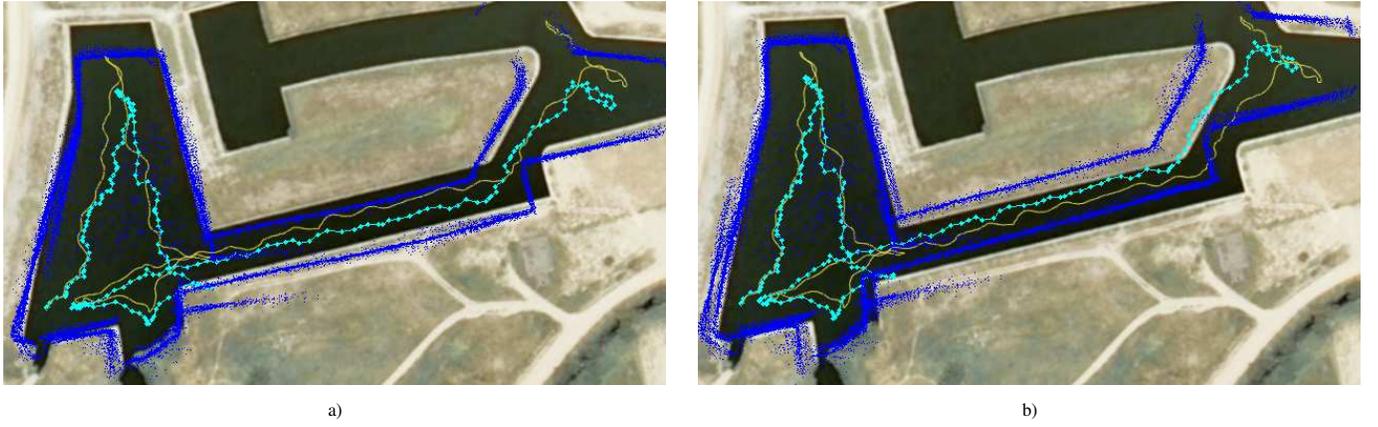


Fig. 10. Results: a) Map and trajectory generated with the MSISpIC output b) Map and trajectory corrected using the translation from MSISpIC output and rotation from the EKF updated with the MTi heading sensor. Comparison with GPS trajectory (yellow).

#### Algorithm 4 MSISpIC

```

MSISpIC() {
  [ $S_{ref}, \hat{q}_{ref}, P_{q_{ref}}$ ] = ScanGrabbing()
   $\hat{q}_{global} = \mathbf{0}$ 
  while(true) {
    [ $S_{new}, \hat{q}_{new}, P_{q_{new}}$ ] = ScanGrabbing()
     $\hat{q}_{pIC} = pIC(S_{ref}, S_{new}, \hat{q}_{ref}, P_{q_{ref}})$ 
     $\hat{q}_{global} = \hat{q}_{global} \oplus \hat{q}_{pIC}$ 
     $S_{ref} = S_{new}$ 
     $\hat{q}_{ref} = \hat{q}_{new}$ 
  }
}

```

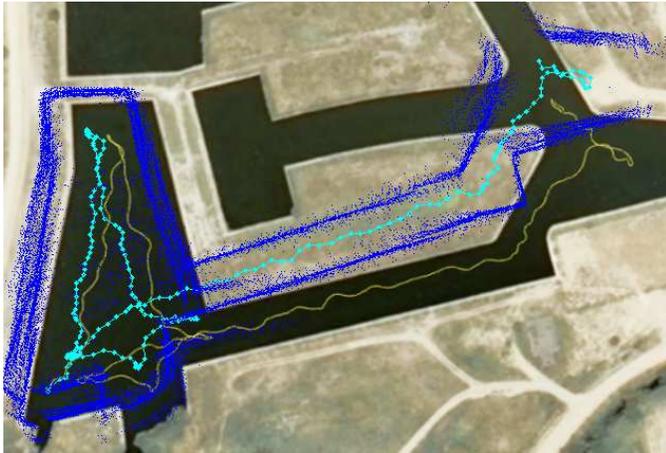


Fig. 9. Trajectory and map generated with odometry (cyan). GPS trajectory (yellow) used as a ground truth.

on the Catalan coast [18] [19]. This dataset is useful to test if the algorithm is capable to register the limited information provided by each scan in a large structure. The survey mission was carried out using ICTINEU<sup>AUV</sup> [8] traveling along a 600 meters path. The MSIS was configured to scan the whole 360° sector and it was set to fire up to a 50m range with a 0.1m resolution and a 1.8° angular step. Dead-reckoning was computed using the velocity reading coming from the

DVL and the heading data obtained from the MTi sensor, both merged using the described EKF. Standard deviation for the MSIS sensor was set as it is specified by the manufacturer, 0.1m in range and 1.8° in angular measurements. Fig. 9 shows the trajectory and the map estimated using the dead-reckoning method. Fig. 10 shows the trajectory and the map estimated with the MSISpIC algorithm. In these figures, the estimated trajectory is plotted on an orthophotomap together with the GPS ground truth for comparison. It can be appreciated that the dead-reckoning estimated trajectory suffers from an important drift which is considerably reduced when the MSISpIC algorithm is used. While the dead-reckoning map (Fig. 9) has worse resolution showing thicker walls, the one build with MSISpIC presents narrower and better defined walls. Both of them show duplicated walls since this is a consequence of the drift which is improved but not avoided with the MSISpIC. In Fig. 10.a it can be appreciated that the mapped size of the polygonal channel is smaller than the actual size. The same happens with the long, almost horizontal, water channel. This problem arises because during part of the trajectory, the robot traverses an area where the scan only observes one or two walls parallel to the robot path, being able to correct the lateral displacement but still drifting in the forward direction. It is worth noting that, even in the presence of structures in all the directions, scan matching algorithms are expected to drift due to its iterative formulation. Although not appreciated in Fig. 10.a, the drift in orientation is particularly dangerous since it quickly propagates to the position. To overcome this problem a variation of the algorithm has been tested. In this case, the MSISpIC estimates translation only while the EKF filter estimates the heading acquired from the heading sensor. The resulting map of such strategy is shown in Fig. 10.b. In this case, the size of the polygonal water channel is more close to the real one and the trajectory seems to be more close to the ground truth estimated with the GPS. Nevertheless, the experiment is not long enough to show a clear advantage.

#### VIII. CONCLUSIONS

This paper analyzes and compares two well known scan matching algorithms: the MbICP and the pIC. The two algorithms are tested with synthetic noisy data simulating a

post-processed output of a MSIS sensor in a static scenario. The study concludes that using unprecise sonar sensor data the statistical framework of the pIC algorithm provides much better results than the MbICP, which works in a metric framework. Next, a variation of the pIC algorithm called MSISpIC is presented, which is able to perform underwater scan matching using a MSIS. To deal with the motion induced distortion of the acoustic image, an EKF is used to estimate the robot motion during the scan. The filter uses a constant velocity model with acceleration noise for motion prediction and velocity (DVL) and attitude measurements (MTi) for updating the state. Through the compounding of the relative robot position within the scan, with the range and bearing measurements of the beams gathered with the sonar, the acoustic image gets undistorted. Assuming Gaussian noise, the algorithm is able to predict the uncertainty of the sonar measurements with respect to a frame located at the position occupied by the robot at the beginning of the scan, before applying the standard pIC algorithm. The proposed method has been tested with a dataset acquired during a survey mission in an abandoned marina located in the Girona coast. The results show substantial improvements in trajectory correction and map reconstruction.

#### ACKNOWLEDGMENTS

This research was sponsored by the Spanish government (DPI2005-09001-C03-01) and FREEsubNET (MRTN-CT-2006-036186). The authors want to thank Javier Mínguez and Luís Montesano from University of Zaragoza for their helpful advices to fulfill this paper.

#### REFERENCES

- [1] C. WangThorpe and A. Suppe, "Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds," *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pp. 416–421, 9–11 June 2003.
- [2] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1, pp. 206–211 vol.1, 27–31 Oct. 2003.
- [3] S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila, "Autonomous rover navigation on unknown terrains functions and integration," in *ISER '00: Experimental Robotics VII*. London, UK: Springer-Verlag, 2001, pp. 501–510.
- [4] J. Minguez, "Integration of planning and reactive obstacle avoidance in autonomous sensor-based navigation," *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 2486–2492, 2–6 Aug. 2005.
- [5] A. Grossmann and M. Poli, "Robust mobile robot localisation from sparse and noisy proximity readings," 1999. [Online]. Available: [citeseer.ist.psu.edu/article/gro99robust.html](http://citeseer.ist.psu.edu/article/gro99robust.html)
- [6] I. Cox, "Blanche-an experiment in guidance and navigation of an autonomous robot vehicle," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 2, pp. 193–204, Apr 1991.
- [7] J. A. Castellanos, J. D. Tardos, and J. Neira, "Constraint-based mobile robotlocalization," *Advanced Robotics and Intelligent Systems, IEE*, 1996.
- [8] D. Ribas, N. Palomer, P. Ridao, M. Carreras, and E. Hernandez, "Ictineu AUV wins the first SAUC-E competition," in *IEEE International Conference on Robotics and Automation, 2007*.
- [9] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*
- [10] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pp. 145–152, 2001.
- [11] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2d range scans," in *CVPR94, 1994*, pp. 935–938.
- [12] J. Minguez, F. Lamiroux, and L. Montesano, "Metric-based scan matching algorithms for mobile robot displacement estimation," *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 3557–3563, 18–22 April 2005.
- [13] L. Montesano, J. Minguez, and L. Montano, "Probabilistic scan matching for motion estimation in unstructured environments," *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, 2–6 Aug. 2005.
- [14] A. Burguera, Y. Gonzalez, and G. Oliver, "Probabilistic sonar scan matching for robust localization," *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3154–3160, 10–14 April 2007.
- [15] Y. Bar-Shalom and T. Fortman, *Tracking and Data Association*. Academic Press, 1998.
- [16] D. Ribas, J. Neira, P. Ridao, and J. D. Tardos, "Auv localization in structured underwater environments using an a priori map," in *7th IFAC Conference on Manoeuvring and Control of Marine Crafts*, Lisbon, Portugal, september 2006.
- [17] T. Fossen, *Guidance and Control of Ocean Vehicles*, J. Wiley and S. Ltd., Eds., 1994.
- [18] D. Ribas, "David ribas homepage," <http://eia.udg.edu/~dribas>, Online August 2008.
- [19] D. Ribas, P. Ridao, J. Tardós, and J. Neira, "Underwater SLAM in man made structured environments," *Journal of Field Robotics*, vol. Accepted for publication, 2008.

# SLAM and Map Merging

A. León, R. Barea, L.M. Bergasa, E. López, M. Ocaña and D. Schleicher

**Abstract**—This paper presents a multi-robot mapping and localization system. Learning maps and efficient exploration of an unknown environment is a fundamental problem in mobile robotics usually called SLAM (simultaneous localization and mapping problem). Our approach involves a team of mobile robots that uses Scan-Matching and Fast-SLAM techniques based on scan-laser and odometry information for mapping large environments. The single maps generated for each robot are more accurate than the maps generated only by odometry. When a robot detects another, it sends its processed map and the master robot generates a very accurate global map. This method cuts down the global map building time. Some experimental results and conclusions are presented.

**Index Terms**— Multi-robot SLAM, scan-matching, fast-slam, rao-blackwellised particle filter.

## I. INTRODUCTION

LEARNING maps and efficient exploration of unknown environments is a fundamental problem in mobile robotics. This problem is usually called SLAM (simultaneous localization and mapping problem) [1, 2, 3, 4, 5, 6], which includes estimating the robot's position on the map and building up a map using the sensory input and the estimated robot's pose.

The problem of exploration of an unknown environment has been extensively studied, first using single robot systems with a variety of sensors and later using teams of robots. The first multi-robot exploration systems implementations were simple extensions of the single robot implementations. Multiple robot systems are more complex than other distributed systems because they have to deal with a real environment, which is more difficult to model since it is dynamic, unpredictable, noisy, etc. Therefore, the extension to multiple robots systems brings several new challenges and difficulties [7][8]: robot coordination, integration of information collected by different robots into a consistent map and dealing with limited communication.

Multirobot exploration systems are usually classified as centralized and decentralized. Centralized systems obtain solutions close to the optimal but they are computationally intensive and its good functioning depends on the central

module. On the other hand, decentralized systems are flexible and robust, but frequently achieve significantly suboptimal solutions. Therefore, the difficulty of the coordination task strongly depends on the knowledge of the robots. If the robots know their relative locations and share a map of the area they explored so far, then effective coordination can be achieved by guiding the robots into different, nonoverlapping areas of the environment [9], [10], [11]. However, if the robots do not know their relative locations, then it is far less obvious how to effectively coordinate them, since the robots do not share a common map or reference frame [7].

Map merging task consists on building a consistent model of an environment with data collected from different robots. If the initial locations of the robots are known, map merging is a rather straightforward extension of a single robot mapping [12], [13], [14]. If robots do not know their relative locations, it is more difficult, since it is not clear how and where the robots' traces should be connected.

One of the hardest problems in robotic mapping is the loop closure [15]. When a robot navigates throughout a large cycle in the environment, it must face the hard data association problem of connecting correctly the data to its own map under large position errors. To scale to large-scale environments, one option consists on transform sequences of laser range-scans into odometry measurements using range-scan registration techniques [16], which reduces the well-known particle deprivation problem [17][18].

Rao-Blackwellized particle filters have been introduced as effective means to solve the simultaneous localization and mapping (SLAM) problem. This approach uses a particle filter in which each particle carries an individual map of the environment [19], [20]. The main problem of the Rao-Blackwellized approaches is their complexity, measured in terms of the number of particles required to build an accurate map. To solve this, Hahnel et al [15] combine Rao-Blackwellized particle filtering and scan matching with an improved motion model that reduces the number of required particles. Grisetti et al. [21] present an adaptive technique to reduce the number of particles in a Rao-Blackwellized particle filter for learning grid maps, they propose an approach to compute an accurate proposal distribution taking into account not only the movement of the robot but also the most recent observation. This drastic decrease the uncertainty about the robot's pose in the prediction step of the filter.

This paper presents a comparison of different grid-based SLAM algorithms over some application examples. A technique for merging maps from several robots is also described. The objective is to build up a highly accurate map of an unknown environment with unknown initial

R. Barea is with the Electronics Department, University of Alcalá, Spain (e-mail: [barea@depeca.uah.es](mailto:barea@depeca.uah.es)).

L.M. Bergasa is with the Electronics Department, University of Alcalá, Spain (e-mail: [bergasa@depeca.uah.es](mailto:bergasa@depeca.uah.es)).

E. López Guillén is with the Electronics Department, University of Alcalá, Spain (e-mail: [elena@depeca.uah.es](mailto:elena@depeca.uah.es)).

M. Ocaña is with the Electronics Department, University of Alcalá, Spain (e-mail: [mocana@depeca.uah.es](mailto:mocana@depeca.uah.es)).

position of robots using Rao-Blackwellized particle filtering and scan matching. Finally, some experimental results of our proposal and the main conclusions will be drawn and commented.

## II. PERFORMANCE COMPARISON OF GRID-BASED SLAM ALGORITHMS

This section makes a comparison of different SLAM algorithms. To do that, firstly, we will study the basic of the SLAM techniques then, we will compare them over some application examples and finally we will present some conclusions about its performance.

### A. Scan-Matching SLAM

The scan matching algorithm that we have implemented is an extension of the approach presented in [16], where the problem of concurrent mapping and localization can be treated as a maximum likelihood estimation problem, in which the aim is to find the most likely map given the data:

Let  $m$  be a map. At time  $t$ , the map is written:

$$m_t = \left\{ o_t, x_t \right\}_{t=0, \dots, T} \quad (1)$$

where  $o_T$  denotes a laser scan and  $x_T$  its pose, and  $t$  is time index.

The goal of mapping is to find the most likely map given the data, that is:

$$\underset{m}{\operatorname{argmax}} P(m | d_t) \quad (2)$$

where data  $d_t$  is a sequence of laser range measurements and odometry readings:

$$d_t = \left\{ o_0, a_0, o_1, a_1, \dots, o_T \right\} \quad (3)$$

where  $o$  denotes an observation (laser range scan),  $a$  denotes an odometry reading, and  $t$  are time indexes.

The map likelihood function  $P(m | d_t)$  can be transformed into the following product [18]:

$$P(m | d_t) = \eta \cdot P(m) \int \dots \int \prod_{t=0}^T P(o_t | m, x_t) \cdot \prod_{t=0}^{T-1} P(x_{t+1} | a_t, x_t) dx_1 \dots dx_T \quad (4)$$

where  $\eta$  is a normalizer and  $P(m)$  is prior over maps which, if assumed to be uniform, can safely be omitted. Thus, the map likelihood is a function of two terms, the motion model,  $P(x_{t+1} | a_t, x_t)$ , and the perceptual model,  $P(o_t | m, x_t)$ . If stationarity is assumed (i.e., neither model depends on the time index  $t$ ), the time index can be omitted and instead write  $P(x | a, x')$ , for the motion model and  $P(o | m, x)$  for the perceptual model.

### B. Grid-based Fast-SLAM

This algorithm adapts the Fast-SLAM algorithm to occupancy grid maps. This gives us a volumetric

representation of the environment that does not require any predefined landmark and it can therefore model arbitrary types of environments. The pseudo-code for grid-based Fast-SLAM [22] in each iteration is the following:

For  $i=0$  to  $M$  do

$$x_t^i = \text{motion model } (a_t, x_{t-1})$$

$$w_t^i = \text{model map } (o_t, x_{t-1}^i, m_{t-1}^i)$$

$$m_t^i = \text{update\_map } (o_t, x_{t-1}^i, m_{t-1}^i)$$

$$\tilde{X}_t = \tilde{X}_t + \langle x_t^i, m_t^i, w_t^i \rangle$$

endfor

For  $i=0$  to  $M$  do

$$\text{Draw } i \text{ with probability } \alpha \cdot w_t^i$$

$$\text{Add } \langle x_t^i, m_t^i \rangle \text{ to } X_t$$

Endfor

### C. Rao-Blackwellized mapping

The key idea of the Rao-Blackwellized particle filter [21] for SLAM is to estimate a posterior  $p(x_{1:t} | o_{1:t}, a_{0:t})$  about potential trajectories  $x_{1:t}$  of the robot given its observations  $o_{1:t}$  and its odometry measurements  $a_{0:t}$  and to use this afterwards to compute a posterior over maps and trajectories:

$$p(x_{1:t}, m | o_{1:t}, a_{1:t}) = p(m | x_{1:t}, o_{1:t}) \cdot p(x_{1:t} | o_{1:t}, a_{1:t}) \quad (5)$$

This can be done efficiently, since the posterior over maps  $p(m | x_{1:t}, o_{1:t})$  can be computed analytically given the knowledge of  $x_{1:t}$  and  $o_{1:t}$ .

To estimate the posterior  $p(x_{1:t} | o_{1:t}, a_{1:t})$  over the potential trajectories Rao-Blackwellized mapping uses a particle filter in which an individual map is associated to every sample. Each map is built given the observations  $o_{1:t}$  and the trajectory  $a_{1:t}$  represented by the corresponding particle. The trajectory of the robot evolves according to the robot motion and for this reason the proposal distribution is chosen to be equivalent to the probabilistic odometry motion model.

One of the most common particle filtering algorithms is the Sampling Importance Resampling (SIR) filter. A Rao-Blackwellized SIR filter for mapping incrementally processes the observations and the odometry readings as they are available. This is done by updating a set of samples representing the posterior about the map and the trajectory of the vehicle.

This is done by performing the following four steps:

- **Sampling:** The next generation of particles  $x_t^i$  is obtained from the current generation  $x_{t-1}^i$  by sampling from a

proposal distribution  $\pi(x_t | o_{1:t}, a_{1:t})$ .

- Importance Weighting: An individual importance weight  $w^i$  is assigned to each particle, according to

$$w^i = \frac{p(x_t^i | o_{1:t}, a_{1:t})}{\pi(x_t^i | o_{1:t}, a_{1:t})} \quad (6)$$

The weights  $w^i$  account for the fact that the proposal distribution  $B$  in general is not equal to the true distribution of successor states.

- Resampling: Particles with a low importance weight  $w$  are typically replaced by samples with a high weight. This step is necessary since only a finite number of particles are used to approximate a continuous distribution. Furthermore, resampling allows to apply a particle filter in situations in which the true distribution differs from the proposal.
- Map Estimating: for each pose sample  $x_t^i$ , the corresponding map estimate  $m_t^i$  is computed based on the trajectory and the history of observations according to  $p(m_t^i | x_{1:t}^i, o_{1:t}^i)$ .

#### D. Performance comparison

Firstly, we are going to compare the different SLAM algorithms using a manufactured rectangular map (figure 1). The objective of this initial test is to study the performance of these algorithms in environments with large corridors without points of interest (points where the symmetry of other types of information is conducive to accurate location).

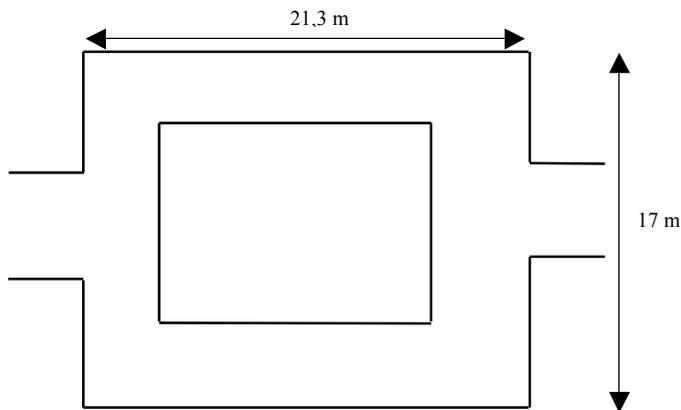


Fig. 1. Manufactured map.

Figures 2 and 3 show the map obtained after one and three map laps, respectively, using the scan-matching SLAM algorithm.

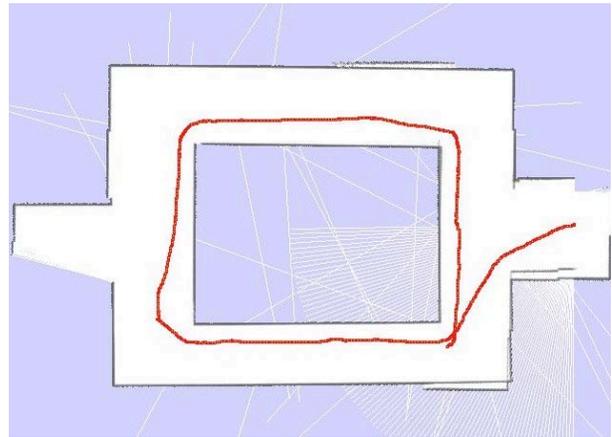


Fig. 2. Initial map using scan-matching SLAM.

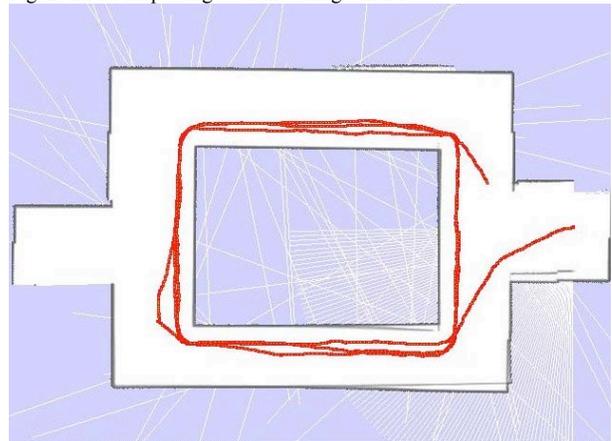


Fig. 3. Scan-matching map (3 laps).

Figures 4 and 5 show the map obtained after one and three map laps, respectively, using the grid-based Fast-SLAM algorithm.

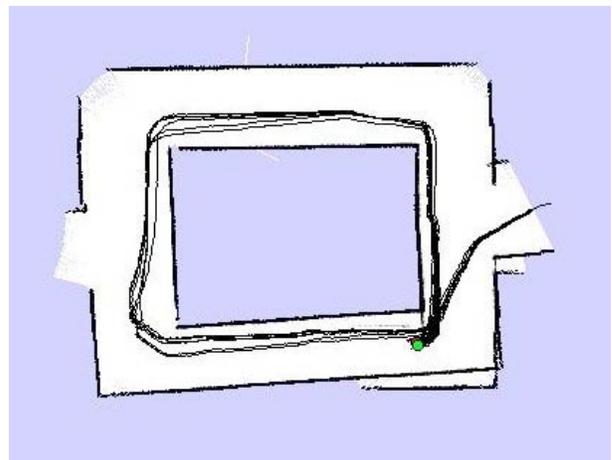


Fig. 4. Initial map using grid-based Fast-SLAM.

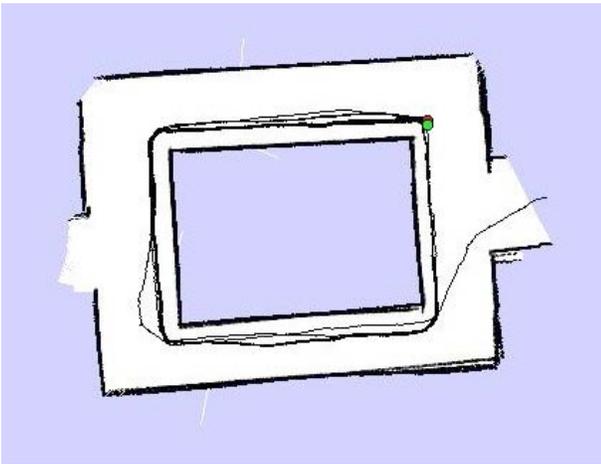


Fig. 5. Grid-based FastSlam map (3 laps).

Figures 6 and 7 show the map obtained after one and three map laps, respectively, using the Rao-Blackwellized particle filter for SLAM.

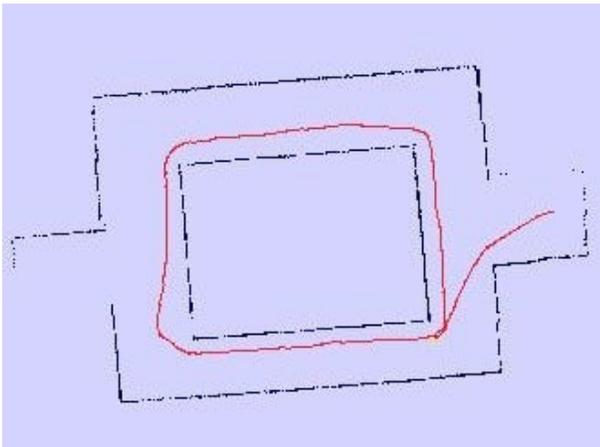


Fig. 6. Initial map using Rao-Blackwellized particle filter for SLAM.

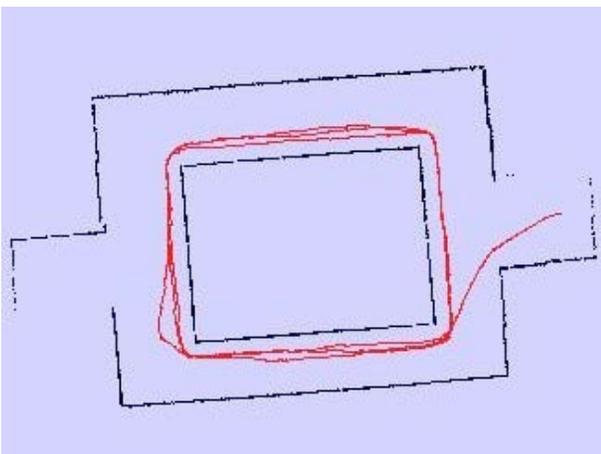


Fig. 7. Rao-Blackwellized Slam map (3 laps).

All cases under study show that errors decrease and results improve with the number of laps. This stands to reason, since the system progressively solves the mapped errors detected in the first lap.

Figure 8 shows the three maps obtained, superimposed over

the original map. The red unbroken line shows the original map, the green dot-and-dash line (·-·) traces out the map obtained using scan-matching, the orange dashed line (--) shows the map generated using grid-based Fast-SLAM while the blue dotted line (···) shows the map generated using Rao-Blackwellized mapping.

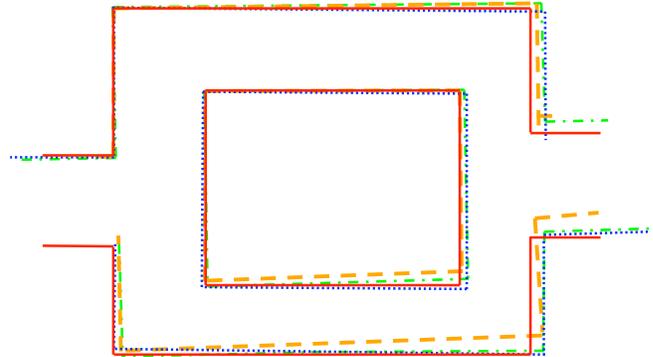


Fig. 8. Mapping errors (3 laps).

Table I shows the dimensional errors detected using the three algorithms. After three laps these errors decrease very slowly and then improvements are very low.

TABLE I  
DIMENSIONAL ERROR

	Real	Scan-Matching	Grid-based Fast-SLAM	Rao-Blackwellized
Length	21.3 m	22.3	21.7	22.4
Error length		4.6%	1.8%	5.1%
Heigh	17 m	17.1	16.8	16.7
Error heigh		0.5%	1.1%	1.7%

These results show that the detected errors are small and negligible in small-dimension environments. The best results would seem to be obtained by the grid-based Fast-SLAM algorithm, but the map geometry in fact shows that the generated map in this case is the worst of the three.

One of the problems that appear in environments of this type (especially for the case of scan-matching) is that if there is not outstanding information in the corridors (points of interest) the results record the overall corridor length smaller than the real one. This is because the maximum likelihood estimation method is used, in which the aim is to find the most likely map given the data. This occurs especially when we work in real environments and is due mainly to the environment-detection limitations of the sensor used. In our case this sensor is a laser SICK-LMS200 and in the corridors when the end is not detected, there are many points where the laser measurement is the same or very similar. The maximum likelihood estimation algorithm therefore generates significant estimation errors.

The next objective is to study a more complex environment. Figure 9 shows a manufactured map from the Electronics Department of University of Alcala (corridor 3 and 4). Figure 10 shows the map obtained using odometry and laser data

without any correction. The robot starts in Lab\_1, goes to Lab\_2 and returns to Lab\_1.

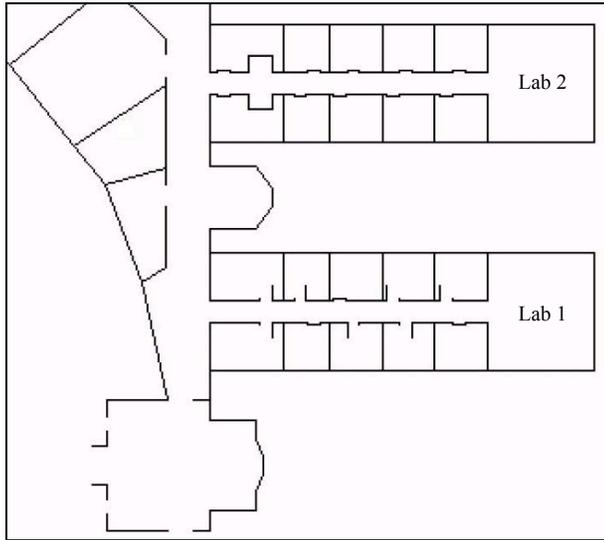


Fig. 9. Department of Electronics map (Corridor 3 and 4).

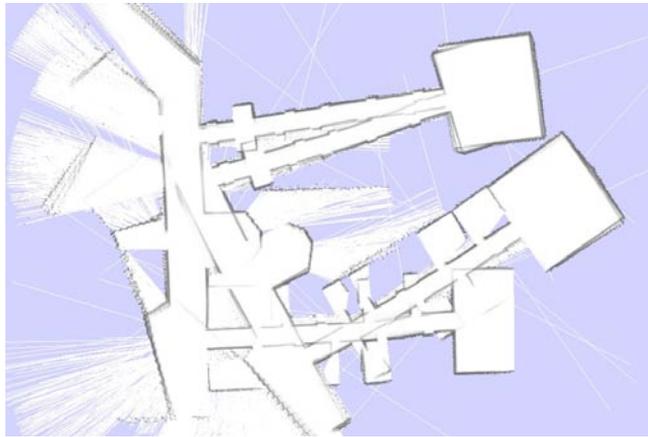


Fig. 10. Map obtained using odometry and laser data without correction.

Figures 11, 12 and 13 show the map obtained using scan-matching SLAM, grid-Fast-SLAM and grid-based Rao-Blackwellized SLAM, respectively. All algorithms solve existing errors using odometry and laser data, but a certain error exists in the orientation of the corridors.

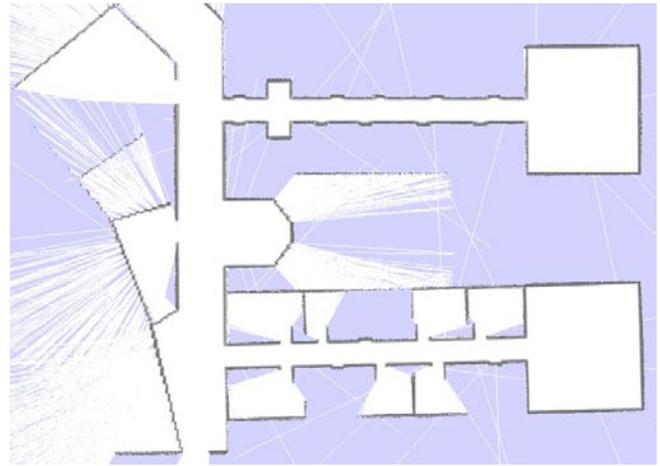


Fig. 11. Map obtained scan-matching SLAM

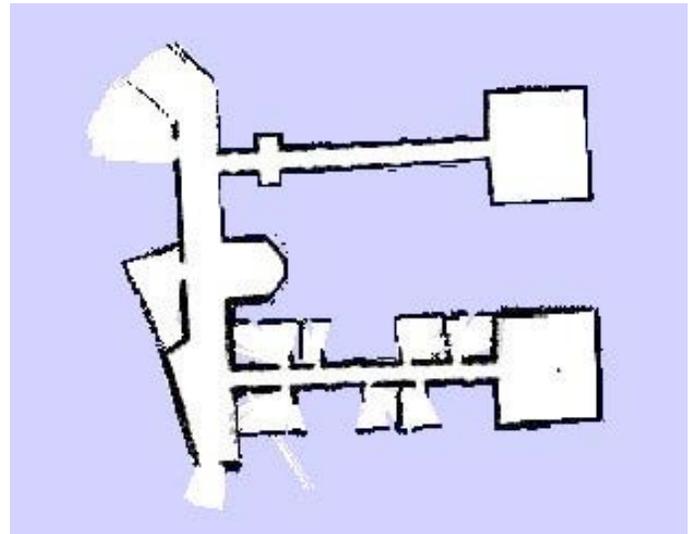


Fig. 12. Map with Fast-SLAM.

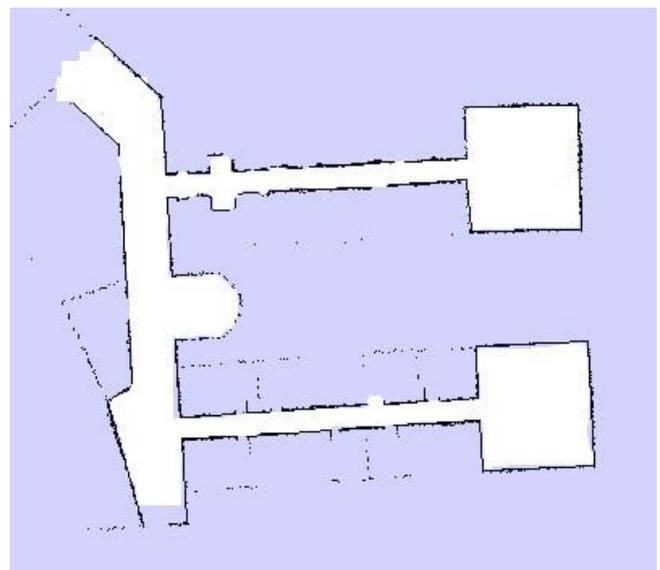


Fig. 13. Map with grid-based Rao-Blackwellized SLAM.

Another experiment to determine the errors introduced by these algorithms involves mapping a larger real environment as is the whole third floor of the Polytechnic building, as can be seen in Fig. 14.

The aim in this case is to compare results using scan-matching and grid-based Rao-Blackwellized SLAM. To do that a lap around the building will be carried out and the map generation errors will be studied. Figures 15 and 16 show the results obtained from one complete lap of the building. The start and the end point is a research laboratory in office 34. Robot trajectory has been superimposed on the real map to compare map-building errors.

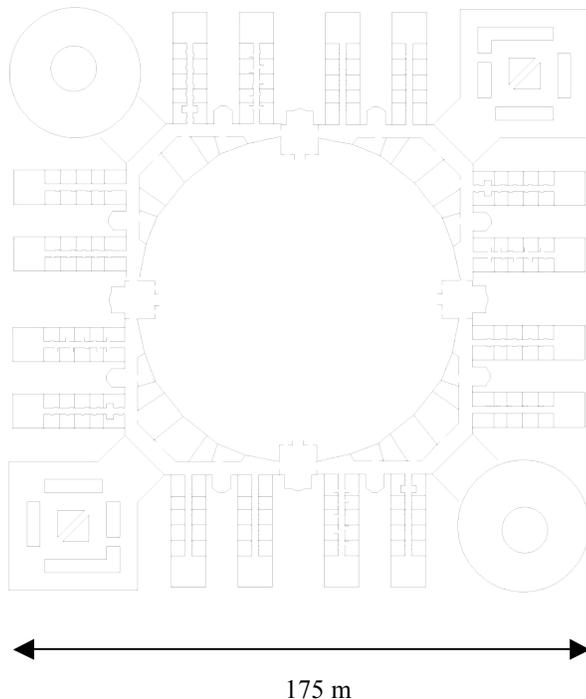


Fig. 14. 3<sup>rd</sup> Floor – Politechnic School.

Both cases show that significant position estimation errors occur after one lap of the building. Figure 15 (scan-matching) shows a position error of about 2 meters, while Figure 16 (Rao-Blackwellized SLAM) gives a 6 meter error. It should be bore in mind here that more than 400 meters have been traveled without any loop closing or point of interest that might allow correction of the existing map. Simulations suggest that results improve after several laps around the building. The results obtained can therefore be considered satisfactory.

### III. ARCHITECTURE

#### A. Robots

Four robotic platforms have been developed based on PeopleBot, pioneer DX and pioneer AT robots of ActivMedia Robotics [23] (see figure 17). Their architecture comprises four large modules: environment perception, navigation, human-machine interface and high-level services. The

perception module is equipped with encoders, bumpers, sonar ring, laser sensor and a vision system based on a PTZ (pan-tilt-zoom) color camera connected to a frame grabber. The human-machine interface is composed of loudspeakers, microphone, a tactile screen, the same PTZ camera used in the perception module, and a wireless Ethernet link. The high-level services block controls the rest of the modules and includes several tasks of tele-assistance, tele-monitoring, providing, reminding and social interaction [24].

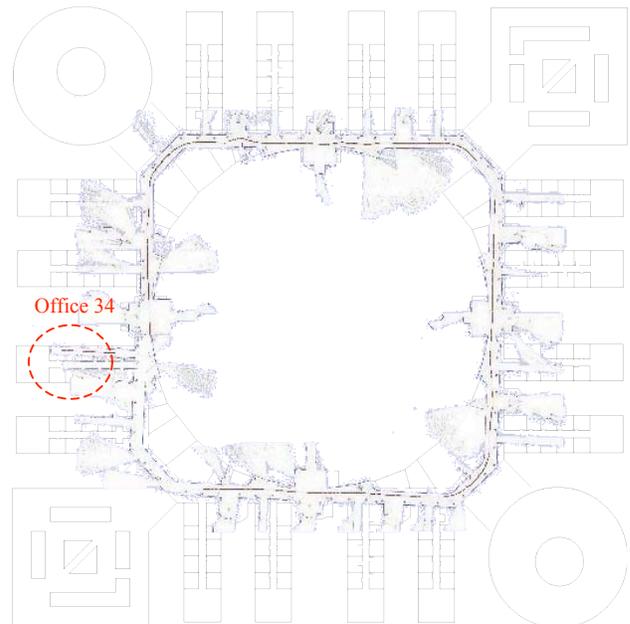


Fig. 15. Scan-matching SLAM map. Comparison of real and obtained map.

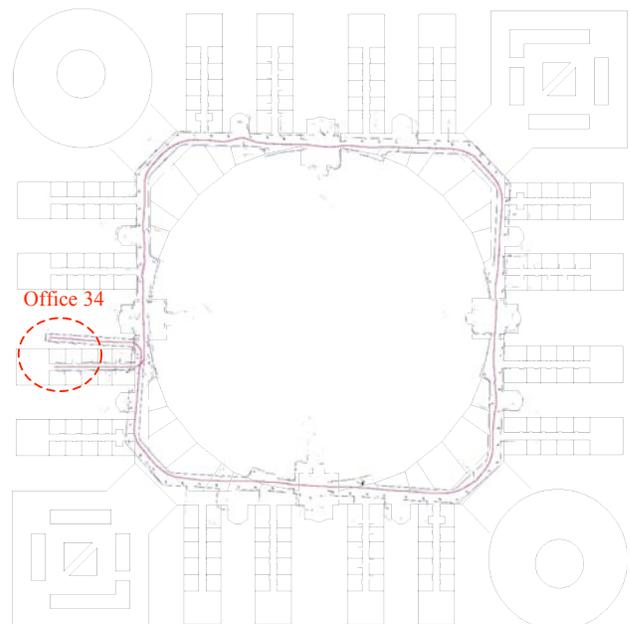


Fig. 16. Grid-based Rao-Blackwellized SLAM map. Comparison of real and obtained map.



Fig. 17. Robots.

### B. Navigation module

The navigation module combines information from the perception module for carrying out different tasks. The core of this module is CARMEN (Carnegie Mellon Robot Navigation Toolkit) [25] which is an open-source collection of software for mobile robot control. CARMEN is modular software designed to provide basic navigation primitives including: base and sensor control, obstacle avoidance, localization, path planning, people-tracking, and mapping.

This source has been modified to implement multi-robot localization, because CARMEN permits only single-robot working; there is also a different initial distribution for studying the robot localization (based on Montecarlo localization). This source implements the motion, perception and detection models. A virtual simulator has also been developed for testing the detection model using visual information and the localization process.

### C. Detection model

Each robot has the ability to sense each others. The detection model describes the probability that robot  $n$  is at location  $x$ , given that robot  $m$  is at location  $x'$  and perceives robot  $n$  with measurement  $r_m$ .

To determine the relative location of other robots, our approach combines visual information obtained from an onboard camera with proximity information coming from a

laser range-finder. Camera images are used to detect other robots and, together with laser range-finder scans, determine the relative position of the detected robot and its distance.

The robots are marked by a unique and colored marker to facilitate their recognition (green cylinder). The marker can therefore be detected regardless of the robot's orientation.

To find robots in the images obtained from a camera, our system first filters the image by employing local color histograms (HIS space color). A basic segmentation algorithm, based on the histogram, is then employed to search for the marker's color. If it is found, this implies that a robot is present in the image.

Once a robot has been detected, its size and position in the image are processed and a laser scan is made for calculating the relative position of one robot respecting the other (see figures 18 to 20). This multi-sensor technique has been proven to be robust in practical tests. Currently, images are analyzed approximately at 20 fps. This framerate is enough for this application.

When a robot detects the other one, a detection model is generated (based on a Gaussian function) representing the probability that the detected robot is at this point. This detection model carries out the adjustment of the particles by means of Collaborative Monte Carlo localization.



Fig. 18. Original image.



Fig. 19. Processed image.

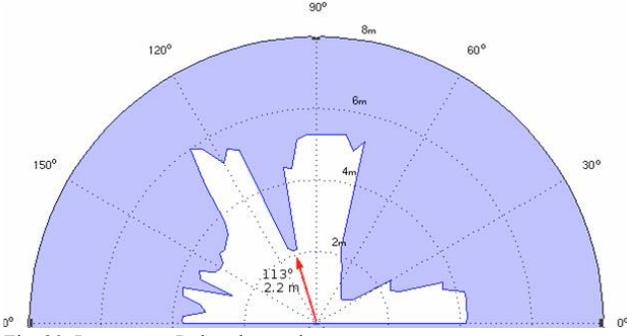


Fig. 20. Laser scan. Robot detected.

#### IV. MULTI-ROBOT MONTE CARLO LOCALIZATION

Monte Carlo Localization has been widely studied in [6,14]. MCL is a recursive Bayes filter that estimates the posterior distribution of robot poses conditioned on sensor data.

The key idea of Bayes filtering is to estimate a probability density over the state space  $x$  conditioned on the data. This posterior is typically called the *belief* and is denoted:

$$Bel(x_t) = p(x_t | o_t, a_{t-1}, o_{t-1}, a_{t-2}, \dots, o_0) \quad (7)$$

Where  $x_t$  is the vector state at time  $t$ ,  $o$  denotes *observations* (*perceptual data* such as laser range or vision measurements) and  $a$  represents *actions* (*odometry data* which carry information about robot motion).

Bayes filters estimate the belief recursively. The initial belief characterizes the initial knowledge about the system state. In the absence of such knowledge, it is typically initialized by a uniform distribution over the state space. In mobile robot localization, a uniform initial distribution corresponds to the global localization problem, where the initial robot pose is unknown.

To derive a recursive update equation, (7) can be transformed by Bayes rule to:

$$Bel(x_t) = \eta \cdot p(o_t | x_t, a_{t-1}, \dots, o_0) p(x_t | x_{t-1}, a_{t-1}, \dots, o_0) \quad (8)$$

$$\eta = p(o_t | a_{t-1}, \dots, o_0)^{-1} \quad (9)$$

Bayes filters assume that the environment is Markov, that is, past and future data are (conditionally) independent if one knows the current state. The Markov assumption implies:

$$p(o_t | x_t, a_{t-1}, \dots, o_0) = p(o_t | x_t) \quad (10)$$

$$p(x_t | x_{t-1}, a_{t-1}, \dots, o_0) = p(x_t | x_{t-1}, a_{t-1}) \quad (11)$$

Therefore, the belief can be denoted by:

$$Bel(x_t) = \eta \cdot p(o_t | x_t) \cdot \int p(x_t | x_{t-1}, a_{t-1}) Bel(x_{t-1}) dx_{t-1} \quad (12)$$

Where  $p(o_t | x_t)$  is called *perceptual model* and  $p(x_t | x_{t-1}, a_{t-1})$  represents the *motion model*.

The key idea of multi-robot localization is to integrate measurements taken at different platforms, so that each robot can benefit from data gathered by other robots than itself. Therefore, when a robot  $n$  is detected by robot  $m$  it is

necessary to introduce the detection model according with data obtained  $r_m$  in (12). In the absence of detections, the Markov localization works independently for each robot. A summary of the multi-robot Markov localization algorithm is:

- Initialize the belief  $Bel_n(x)$  according with initial data (typically uniform distribution).
- If the robot  $n$  receives an observation on (new sensory input)  $o_n$ , it is applied the perception model::

$$Bel_n(x) = \eta \cdot p(o_n | x) \cdot Bel_n(x) \quad (13)$$

- If the robot  $n$  do some action  $a_n$  (receives a new odometry reading), It is applied the motion model:

$$Bel_n(x') = \eta \cdot \int_x p(x' | x, a_n) \cdot Bel_n(x) \cdot dx \quad (14)$$

- And finally, if the robot  $n$  is detected by the  $m$ -th robot it is applied the detection model:

$$Bel_n(x') = \eta \cdot Bel_n(x) \int_x p(x_n = x' | x_m = x, r_m) \cdot Bel_m(x) \cdot dx \quad (15)$$

The idea of MCL is to represent the *belief* by a set of  $m$  weighted samples distributed according to  $Bel(x)$ :

$$Bel(x_t) \approx \{x^i, w^i\}_{i=1, \dots, m} \quad (16)$$

Where  $x^i$  is a *sample* of the random variable  $x$  (pose) and  $w^i$  is called *importance factor* and represents the importance of each sample. The set of samples, thus, define a discrete probability function that approximates the continuous belief  $Bel(x)$ .

The initial set of samples represents the initial knowledge  $Bel(x_0)$  about the state of the dynamical system. For instance, in global mobile robot localization, the initial belief is a set of poses drawn according to a uniform distribution over the robot's universe, annotated by the uniform importance factor  $1/m$ . If the initial pose is known up to some small margin of error,  $Bel(x_0)$  may be initialized by samples drawn from a narrow Gaussian centered on the correct pose.

The recursive update is carried out in three steps:

- Sample  $x_{t-1}^i \sim Bel(x_{t-1})$ . Each such particle  $x_{t-1}^i$  is distributed according to the belief distribution  $Bel(x_{t-1})$ .
- Sample  $x_t^i \sim p(x_t | x_{t-1}^i, a_{t-1})$ . In this case,  $x_t^i$  is distributed according to the product distribution  $p(x_t | x_{t-1}^i, a_{t-1}) \cdot Bel(x_{t-1})$ .
- The importance factor is assigned to the  $i$ -th sample:

$$w^i = \theta \cdot p(o_t | x_t^i) \quad (17)$$

The following example shows how collaborative multi-robot Monte Carlo localization improves single localization [15]. Robot 1 starts out with uniform belief and Robot 2 with Gaussian belief. Figs. 21a.b show the robots' initial position and Figs. 21c.d show the initial particle distributions.

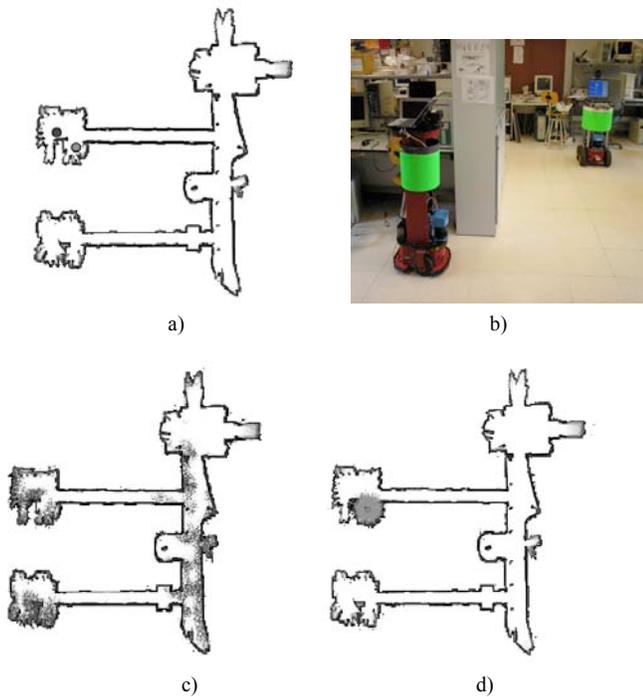


Fig. 21. Initial distributions.

If Robot 1 wanders across the top horizontal corridor (Fig. 22a), when Robot 2 detects Robot 1 (Fig. 22b), the detection model is sent to Robot 1 and this updates its belief distribution. Robot 1 is therefore well-located before reaching the corridor (Fig. 22c).

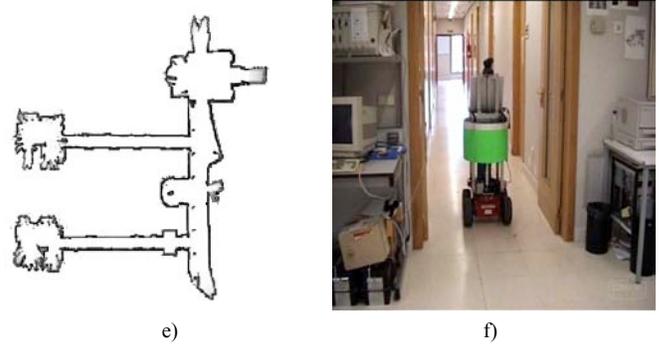
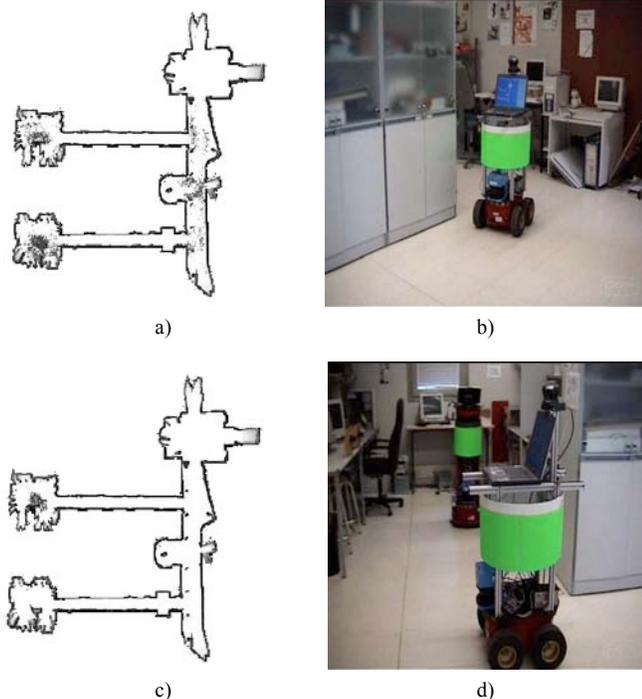


Fig. 22. Collaborative multi-robot Monte Carlo localization.

## V. MULTIROBOT MAP MERGING

This section describes how to build a map from data obtained by multiple robots. We are currently working with scan-match [16], grid-based Fast-SLAM [15] and grid-based SLAM with Rao-Blackwellized particle filter [21] for map merging; comparing the results. To do so, we have modified some codes and parameters in algorithms (GMapping and GridSLAM) obtained from OpenSlam.org [26] which provides SLAM researchers with a platform for publishing their algorithms.

In this initial work we are focused on developing multirobot map merging using a scan-match technique. Next, a merging map example is commented on, working with a scan-match technique (see figures 23 to 27). The goal is mapping corridors 3 and 4 of the Electronics Department of the Polytechnic School. Figure 23 shows the trajectories followed for each robot and figure 24 shows the global map using CARMEN (Montecarlo localization). Robot 1 explores across corridor 4 and Robot 2 explores corridor 3; each robot builds its partial map and calculates its pose at any time. When Robot 1 meets Robot 2, Robot 2 send its map to Robot 1. Robot 1 uses the partial Robot 2 map and the detection model (Robot 2's pose detected) to generate the global map (Fig. 27).

The main problem we run into here is the need for very precise inter-robot detection for map merging purposes. Any small orientation error would mean that the map built up from the slave robot data would be displaced vis-à-vis the master robot. We are currently working to improve inter-robot detection, making it much more precise in the interests of correct global map reconstruction.

Our final goal is to be able to map exteriors and implement other SLAM techniques for reconstructing the most trustworthy environment possible. One of the problems we are finding is that 2D laser information outdoors may be of little value in wider settings and we are therefore trying to merge laser with vision. We are thus working towards the introduction of visual information in localization and mapping process to improve SLAM's performance outdoors.

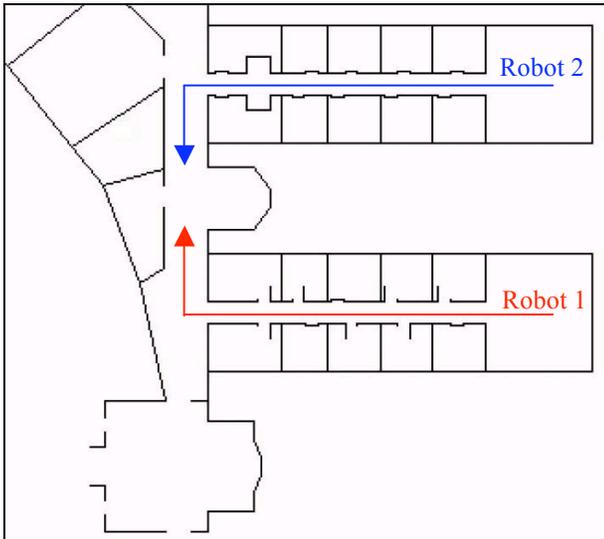


Fig. 23. Robots trajectories.

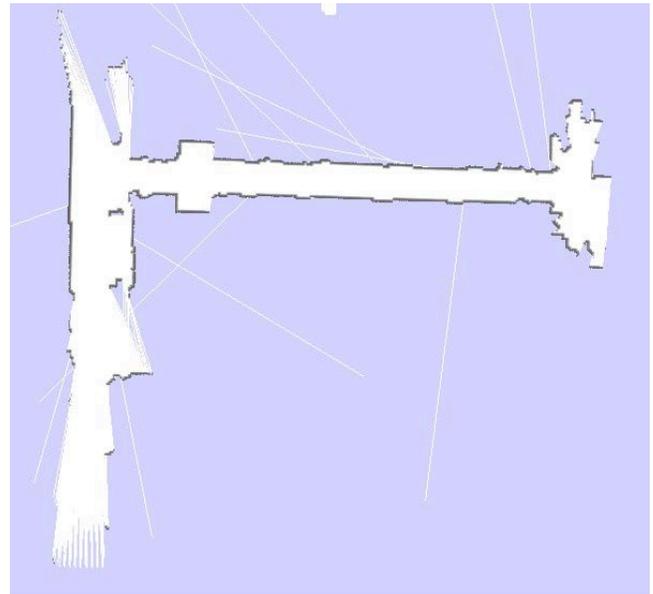


Fig. 25. Partial map built by robot 2.

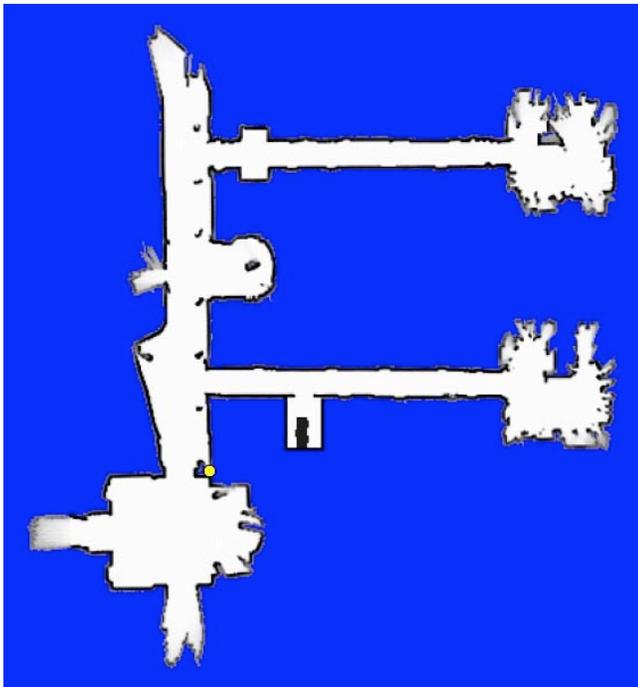


Fig. 24. Complete map obtained using CARMEN.

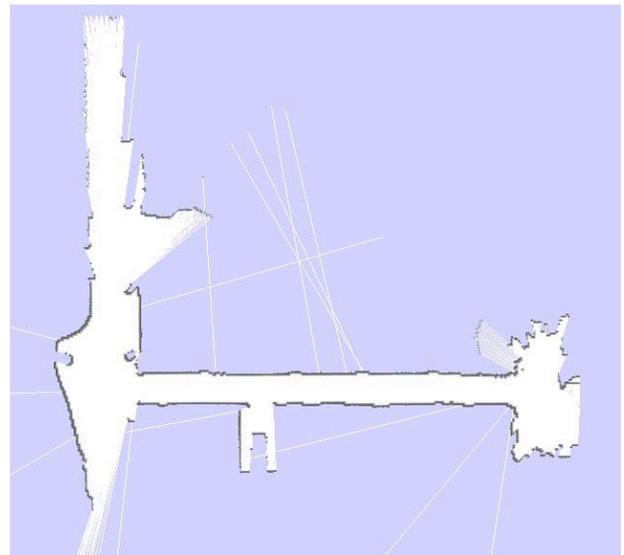


Fig. 26. Partial map built by robot 1.

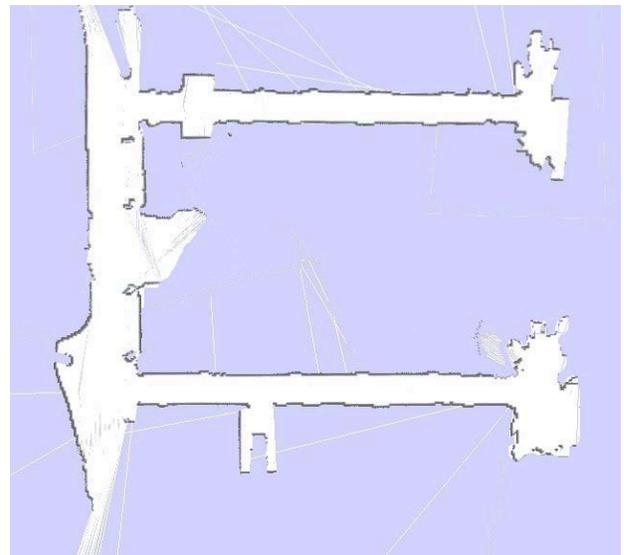


Fig. 27. Global map using scan-matching.

## VI. CONCLUSION

We present herein initial work in mobile robot mapping. We show and comment on results using several algorithms for SLAM (scan-matching, grid-FastSLAM and grid-based SLAM with Rao-Blackwellized particle filter) for a single robot and multi-robot map merging using a scan-match technique. The results show that it is possible to use a team of robots to explore and navigate in unknown environments.

Our future work will be focused on improve performance of our proposal for indoor environments and to generalize it for outdoor environments. According to this point, the results obtained show that it is necessary to use other sources of information (3D laser, vision, etc) to obtain similar results to those obtained indoors.

## ACKNOWLEDGMENT

The authors would like to express their gratitude to the Comunidad de Madrid and the University of Alcalá for their support through the projects RoboCity2030 (CAM-S-0505/DPI/000176) and SLAM-MULEX (CCG07-UAH/DPI-1736).

## REFERENCES

- [1] J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardós. The SPMAP: A probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15(5):948–953, 1999.
- [2] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localization and map building (SLAM) problem. In *ICRA'2000 Workshop on Mobile Robot Navigation and Mapping*, 2000.
- [3] H. Durrant-Whyte and T. Bailey, “Simultaneous Localization and Mapping (SLAM): Part I”, in *IEEE Robotics & Automation Magazine*, 2006.
- [4] T. Bailey and H. Durrant-Whyte, “Simultaneous Localization and Mapping (SLAM): Part II”, in *IEEE Robotics & Automation Magazine*, 2006.
- [5] M. Montemerlo, S. Thrun D. Koller, and B. Wegbreit, “FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges”. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 2003.
- [6] H. Chang.; C.Lee.; Y. Hu.; Y. Lu, “Multi-robot SLAM with topological/metric maps”, in *Intelligent Robots and Systems*, 2007.
- [7] D. Fox, J. Ko. K. Konolige, B. Limketkai, D. Schulz, B. Stewart, “Distributed Multirobot Exploration and Mapping”, in *Proceedings of the IEEE*, Volume 94, Issue 7, pp:1325 – 1339, July 2006.
- [8] Parker, L.E. “Current state of the art in distributed robot systems”, in *Dist. Auton. Syst.*, L.E. Parker, G. Bekey and J. Barhen (Eds.), vol. 4, pp. 3–12. Springer-Verlag, 2000.
- [9] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, “Collaborative multi-robot exploration”, in *Proc. IEEE Int. Conf. Robotics Automation (ICRA)*, 2000.
- [10] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, “Coordinated multi-robot exploration”, in *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 376–386, Jun. 2005.
- [11] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, “Coordination for multi-robot exploration and mapping”, in *Proc. Nat. Conf. Artificial Intelligence (AAAI)*, 2000.
- [12] K. Konolige, D. Fox, C. Ortiz, A. Agno, M. Eriksen, B. Limketkai, J. Ko, B. Morisset, D. Schulz, B. Stewart, and R. Vincent, “Centibots: Very large scale distributed robotic teams”, in *Experimental Robotics: 9<sup>th</sup> Int. Symp.*, Springer Tracts in Advanced Robotics (STAR), M. Ang and O. Khatib, Eds. New York: Springer Verlag, 2005.
- [13] S. Thrun, W. Burgard, D. Fox, “A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping”, in *Robotics and Automation*, 2000. *Proceedings. ICRA '00. IEEE International Conference*, April 2000.
- [14] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte, “Towards multi-vehicle simultaneous localisation and mapping”, in *Proc. IEEE Int. Conf. Robotics Automation (ICRA)*, 2002.
- [15] D. Hahnel, W. Burgard, D. Fox, D.S. Thrun, “An efficient fastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements”, in *Intelligent Robots and Systems*, 2003.
- [16] D. Hahnel, D. Schulz, W. Burgard, “Map building with mobile robots in populated environments”, In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002.
- [17] S. Thrun, “A probabilistic online mapping algorithm for teams of mobile robots”, in *International Journal of Robotics Research*, 20(5):335–363, 2001.
- [18] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. Technical Report CUED/FINFENG/ TR 380, Cambridge University, Department of Engineering, 2000.
- [19] K. Murphy. “Bayesian map learning in dynamic environments”. In *Neural Info. Proc. Systems (NIPS)*, 1999.
- [20] A. Doucet, J.F.G. de Freitas, K. Murphy, and S. Russel. “Raoblackwellized particle filtering for dynamic bayesian networks. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2000.
- [21] G. Grisetti, G. Stachniss, W. Burgard, “Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling”, in *Robotics and Automation*, 2005.
- [22] S. Thrun, W. Burgard, and D. Fox, “Probabilistic Robotics”. Cambridge, MA: MIT Press, 2005.
- [23] R. Barea, L.M. Bergasa, E. López, D. Schleicher, M. Ocaña and A. León. “Real Patient monitoring in health care working with robotic assistants”, in *Proc. Of the IEEE Internacional Symposium on Intelligent Signal Processing (WISP)*. Pp:711-716. October 2007.
- [24] R. Barea, E. López, L.M. Bergasa, S. Álvarez and M. Ocaña. “Detection Model in Collaborative Multi-Robot Monte Carlo Localization”. *IEEE Workshop on Distributed Intelligent Systems (DIS'06)*, pp. 49-54, 2006.
- [25] Carnegie Mellon Robot Navigation Toolkit. <http://www.cs.cmu.edu/~carmen/index.html>
- [26] [www.openslam.org](http://www.openslam.org).

# Localization through omnivision for a tour-guide robot

C. Gamallo, P. Quintía, C. V. Regueiro and M. Mucientes

**Abstract**—The localization of a mobile robot in a real environment is a complex task. In this paper, an algorithm that solves the global localization is presented. The proposal is based on a merit function that ranks different possible poses obtained from the acquired image, together with an iterative process for the minimization of that function using a particle filter. Landmark detection has been done with an omnidirectional camera pointing to the ceiling, combined with an infrared passband filter which extracts the lights. Several real experiments, both for global localization and the kidnapped robot problem, have been done in a museum. Results show a high accuracy, robustness, and real-time execution in this complex and crowded environment.

**Index Terms**—Global localization, omnivision, tour-guide robot, kidnapping problem.

## I. INTRODUCTION

**L**OCALIZATION is one of the most important tasks in the field of autonomous mobile robotics. Determining the location of a mobile robot means finding the Cartesian coordinates and angular orientation relative to an external frame. A localization algorithm must be reliable, robust and executable in real time.

Different types of sensors have been used for localization: laser [16], [9], ultrasonic, or infrared sensors, and vision. Nowadays, cameras are widely used in robotics. The main advantages of these sensors are the quantity of information that can be extracted from one acquisition. This is particularly interesting for localization, as different types of landmarks can be detected using information of shape, color, etc. Also, cameras have a good performance independently of the materials objects are made off.

Popular vision approaches are feature-based [5], [13], which exploit typical properties of the environment or any distinctive and recognizable objects (landmarks), and pixel-based techniques [1], [12], that compute the correlation between images, in order to estimate the mobile robot pose.

In this paper we present an approach based in a map of landmarks and an artificial vision system to estimate the pose of the robot. We use the omnidirectional camera shown in Fig. 1. It provides a very wide field of vision (FOV of about 185°) which covers half the space of the environment and, therefore, it can get a high amount of information in one acquisition. The

C. Gamallo and M. Mucientes are with the Intelligent Systems Group, Department of Electronics and Computer Science, University of Santiago de Compostela (Spain).

E-mail: cristina.gamallo@usc.es, manuel.mucientes@usc.es

C. V. Regueiro and P. Quintía are with the Computer Architecture Group, Department of Electronic and Systems, University of A Coruña (Spain).

E-mail: cvazquez@udc.es, pquintia@udc.es



Fig. 1. The vision system in the tour-guide mobile robot, based on a Pioneer 2-AT, placed at the Domus Museum in A Coruña (Spain). The omnivision camera is marked with a circle.

camera is pointing to the ceiling and elevated 1.5 m over the robot (1.8 m over the ground) (Fig. 1). Thus its movements are restricted to the  $x$ - $y$  plane and the noise or occlusion generated by moving people is minimized.

The landmarks that have been used are the lights placed on the ceiling of the environment (Fig. 3). These are easy to detect, repetitive and usually visible for long trajectories. On the other hand, all the buildings have these kind of landmarks, so there is no need for prior adaptation of the environment in order to use the proposed localization method. The main problem is their individual identification, because they usually are identical, enhancing the difficulties for data association.

The key point of our solution is a function that evaluates the similarity among the positions of the landmarks detected in the image acquired with the camera and the theoretical position of the landmarks projected according to the camera model. We present experiments in a dynamic museum environment (Fig 2) which show that our algorithm can estimate the global robot pose even in situations where odometry suffers from serious noise and, also, when the robot is kidnapped.

The paper is organized as follows. Section II contains an

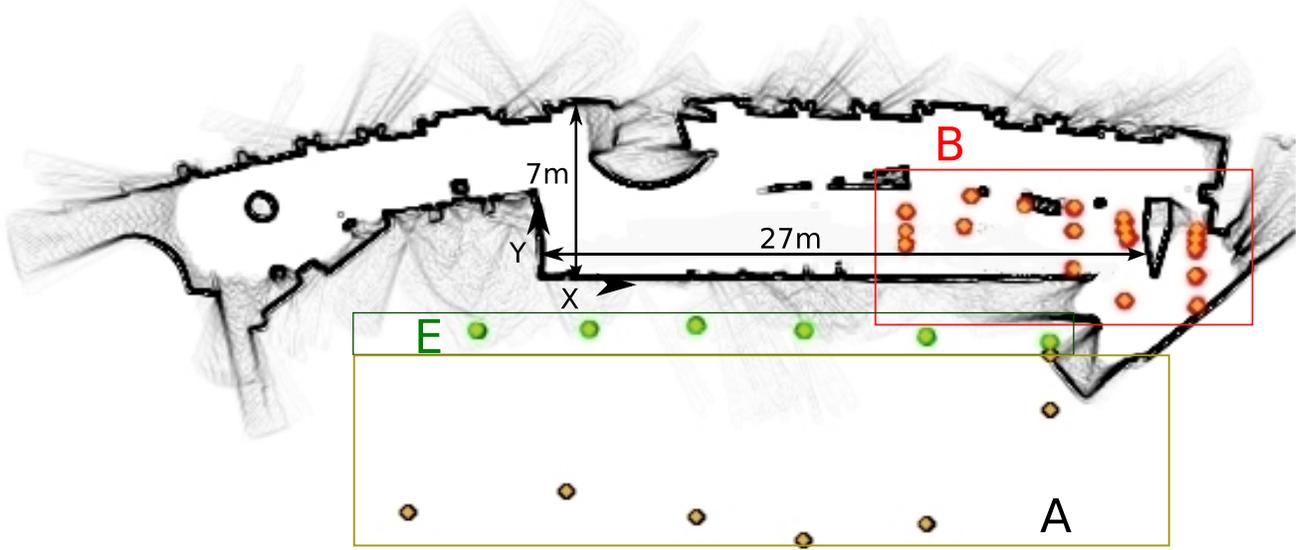


Fig. 2. 2D environment grid-map with landmarks positions (circles). The position of the landmarks is shown in table I



Fig. 3. Some of the lights that have been used as landmarks: (a) Type of lights placed in areas E and A of the map (Fig. 2); (b) Type of lights placed in area B of the map lights of the map (Fig. 2).

overview of related work. The next two sections describe in detail the omnivision and localization systems. Section V presents the experimental results in a real environment and, finally, the last section points out the conclusions and future work.

## II. RELATED WORK

In the last years, several vision-based algorithms have been developed as solution to the localization problem. Techniques vary substantially, depending on the sensors, their geometric models, and the representation of the environment.

The probabilistic approach is the most widely used in recent publications. For example, in [16] a Monte Carlo localization algorithm is presented to solve the global localization problem using a camera. They used a visual map of the ceiling, obtained by mosaicing, and localized the robot using a simple scalar brightness measurement as the sensor input. Nevertheless, this system is sensitive to bumps and, as a result of the small FOV of the camera, at some time instants none or few lights can be seen. This provokes more uncertainty in the knowledge of the pose of the robot. A similar approach is presented in [12]. They use an omnivision camera oriented to the ceiling too, but

TABLE I  
POSITION (METRES) OF THE LANDMARKS IN FIG. 2.

LABEL	X	Y	Z
Lights Type E			
E1	-3.45	-2.26	11.39
E2	1.34	-2.15	11.24
E3	5.9	-2.04	11.29
E4	10.56	-2.25	11.43
E5	15.8	-2.50	11.48
E6	21.1	-2.80	11.60
Lights Type A			
A1	-6.45	-10.00	11.39
A2	0.34	-9.15	10.50
A3	5.9	-10.24	10.50
A4	10.56	-11.25	10.50
A5	15.8	-10.50	10.50
A6	21.1	-5.70	10.50
Lights Type B			
B11	14.98	2.86	3.25
B12	14.98	2.00	3.25
B13	14.99	1.69	3.29
B21	17.71	3.51	3.24
B22	17.5	2.28	3.24
B31	20.16	3.21	3.16
B41	22.13	3.08	3.16
B42	22.13	2.08	3.16
B43	22.13	0.3	3.16
B51	24.34	2.47	3.14
B52	24.34	2.07	3.14
B53	24.54	1.77	3.14
B54	24.34	-0.97	3.24
B61	27.50	1.97	3.24
B62	27.50	1.67	3.24
B63	27.50	1.37	3.24
B64	27.50	0.17	3.24
R65	27.50	-1.27	3.24

it is based on information theory to get the global trajectory. The main problem of this work is the high computational cost.

In [1], [14], [11] a Monte Carlo localization is used, but

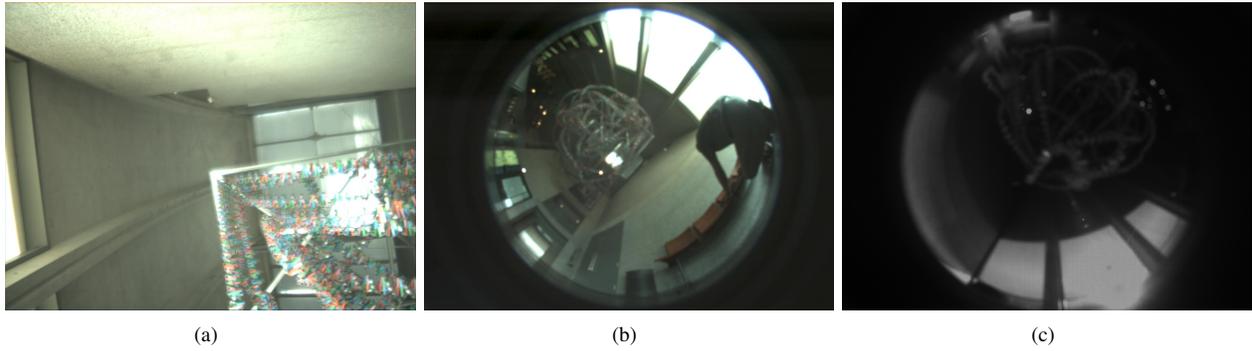


Fig. 4. (a) Original image, (b) omnidirectional image acquired without IRP filter and (c) omnidirectional image with IRP filter.

they create a database with images of every route and their poses. The robot can be localized by correlation between the captured images and the database images on real time. These systems have the drawback that they cannot work in other routes of the environment.

Menegatti et al. [10] have developed a system which uses a chromatic map of the floor to compute the robot pose. They obtained similar results to other proposed algorithms, but their system is limited to environments with natural color transitions and it is also light-sensitive.

The first work that used omnidirectional vision to localize a mobile robot was published in 1986 by Cao et al. [4]. Nevertheless, few related studies were published before the end of the nineties. Nowadays, such systems have become popular due to their low cost in addition to the benefit of having a very wide field of vision. There are two types of omnidirectional vision configurations: *catadioptric* (as in [3], [7], [13], [12]), where the camera images are obtained through a conic mirror and *dioptric*, where images are captured through a lens [11].

Other implementations use landmarks (beacons) of the environment to get the pose of the robot. For instance, in [13] the goals of a RoboCup field are used as marks and [3], [16] are based on features of the environment (corners, walls, lights ...) which were previously mapped.

Our model is similar to these ones but, in addition, we do not have the occluded landmarks problem and our process to discover landmarks is simple, fast and efficient. A similar approach (omnivision camera oriented to the ceiling) is used in [12], but it is based on information theory to get the global trajectory.

### III. VISUAL SYSTEM

The vision system consists of a color digital camera MDCS2, with a *fish-eye* lens FE185CO46HA-1 and a passband infrared filter (IRP) type HOYA IR85.

The fish-eye lens has a high resolution and a wide angle of view ( $185^\circ$  degrees), so that the amount of information captured at a time instant is high. Due to this wide field of view, the number of landmarks in an environment can be reduced because each of them can be detected from a higher number of places (Fig. 4(b)).

The infrared passband filter *IRP* dims the light visible components and only allows to pass the components near to

the IR range (Fig. 4(c)) which makes simpler the landmarks detection.

#### A. Camera Model

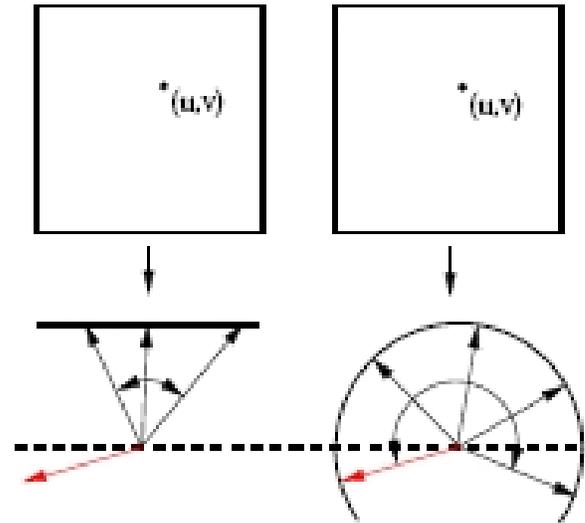


Fig. 5. The *Pin-Hole* camera model based on a flat retina (left) compared with the omnidirectional camera model based on spherical retina (right).

The camera model describes how a 3D scene is transformed into a 2D image (Fig. 6). The standard model is the *Pin-Hole*, which projects the scene on a flat retina (Fig. 5), but it is limited to cameras with  $FOV \ll 180^\circ$ . The other cameras require a model based on a spherical retina (Fig. 5). In our system, we have used a projection model developed by Pajdla and Bakstein [2] that describes the relation between the angle ( $\theta$ ) formed between the optical axis and the light ray, and the distance  $r$  from the image center  $(u_0, v_0)$  to the projection point of  $B(u_B, v_B)$  in the image (Fig. 6):

$$r = a * \tan \frac{\theta}{b} + c * \sin \frac{\theta}{d}, \quad (1)$$

where  $a$ ,  $b$ ,  $c$ , and  $d$  are parameters of the model. This function makes it possible to calculate the coordinates of the image  $(u, v)$  depending on the azimuth ( $\varphi$ ) and the elevation ( $\theta$ ) (Fig. 6):

$$\left. \begin{aligned} u &= u_0 + r * \cos\varphi \\ v &= \beta * (v_0 + r * \sin\varphi) \end{aligned} \right\} \quad (2)$$

where  $\beta$  is the ratio between the width and the height of a pixel.

### B. A Landmark Projection

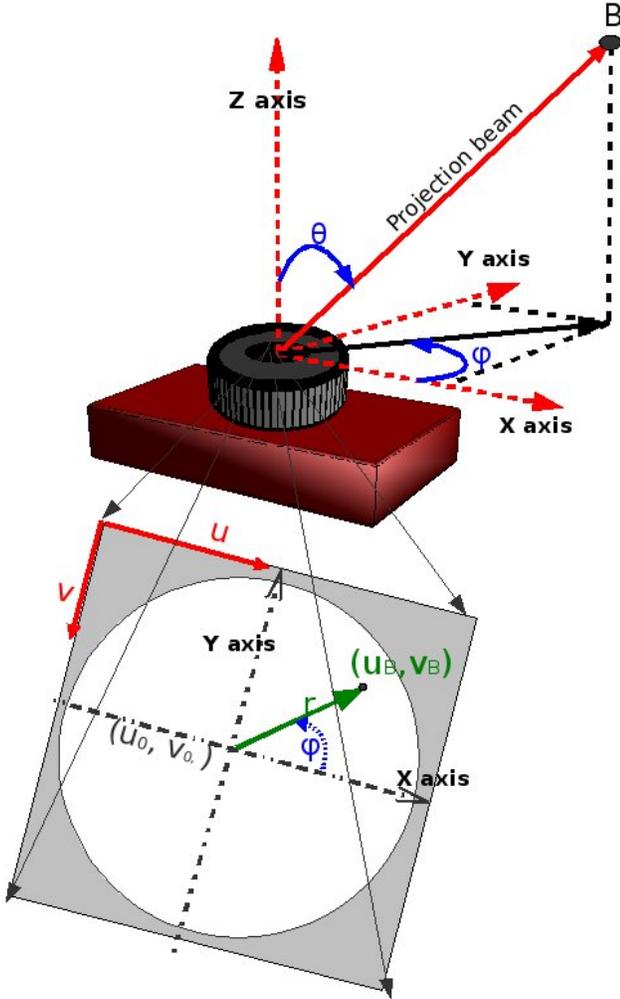


Fig. 6. Theoretical omnidirectional camera model and projection of a point  $B$ . Its projection ray is defined by the elevation ( $\theta$ ) and the azimuth ( $\varphi$ ), both in the camera coordinate system.  $r$  and  $\varphi$  are the polar coordinates of the projected point  $(u_B, v_B)$ .  $(u_0, v_0)$  are the coordinates of the image center.

A Landmark projection is the calculation of the image coordinates  $(u_B, v_B)$  for a landmark  $i$  ( $B$  in Fig. 6) in the world, given its coordinates in the world ( $\mathbf{B}_i^W$ ) and the coordinates of the camera ( $\mathbf{C}^W$ ).

First, we have to change the landmark coordinates ( $\mathbf{B}_i^W$ ) to the camera reference system ( $\mathbf{B}_i^C$ ) with the rotation matrix  $R_C$  between the camera and the world:

$$\mathbf{B}_i^C = R_C * \mathbf{B}_i^W - \mathbf{C}_W \quad (3)$$

From  $\mathbf{B}_i^C$ , we obtain the elevation ( $\theta$ ) and the azimuth ( $\varphi$ ) angles by applying the traditional Euclidean transformations

(Fig. 6). Finally to get the landmark projection ( $Proj(\mathbf{B}_i^C)$ ) represented as  $(u_B, v_B)$  in Fig. 6, we apply Eqs. 1 and 2:

$$Proj(\mathbf{B}_i^C) = (u_{B_i^C}, v_{B_i^C}) \quad (4)$$

### C. Ceiling Map Projection

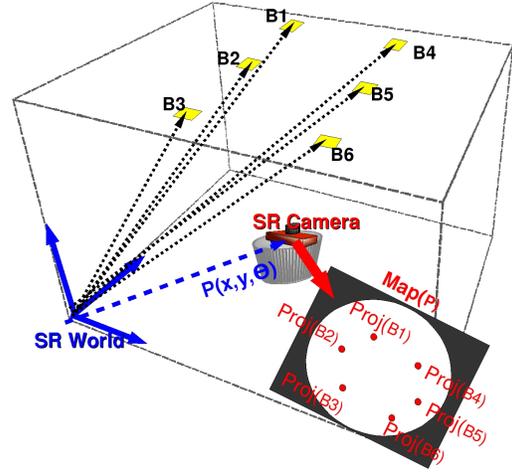


Fig. 7. A graphical example of Ceiling Map Projection for six landmarks.

Ceiling map projection,  $Map(P)$ , is an image that has been generated based on the theoretical camera model described in Sec. III-A. It is composed of a set of points  $\{(u_{B_i}, v_{B_i}), \dots\}$  that correspond with the coordinates of each landmark in the image reference system when the robot is placed at pose  $P$ . The process is detailed in Alg. 1, and a graphical example is shown in Fig. 7.

---

#### Algorithm 1 Calculate $Map(P)$ .

---

```

for all Landmarks  $B_i^W$  in the map (in world Cartesian
reference system) do
   $B_i^C = R_C * \mathbf{B}_i^W - \mathbf{C}_W$ 
   $Proj(B_i^C) = (u_{B_i^C}, v_{B_i^C})$  applying Eqs. 1 and 2
end for

```

---

### D. Landmark detection

The process for detecting landmarks consists of 5 phases: acquisition, preprocessing, segmentation, recognition and features extraction. The output of the system is an array of features for each landmark. In the preprocessing phase, the image (Fig. 8(a)) is transformed to facilitate the processing in the next stages. The techniques that have been used are binary thresholding (Fig. 8(c)) and morphological filtering (dilation) (Fig. 8(d)).

As segmentation techniques, the system uses a Canny filter and contour extraction (Fig. 8(e)). The next step is to extract the characteristics of each region:

- *Ratio*: number of pixels in the perimeter.
- *Centroid*: coordinates of the center of gravity.
- *Radio*: centroid distance to the center of the image.

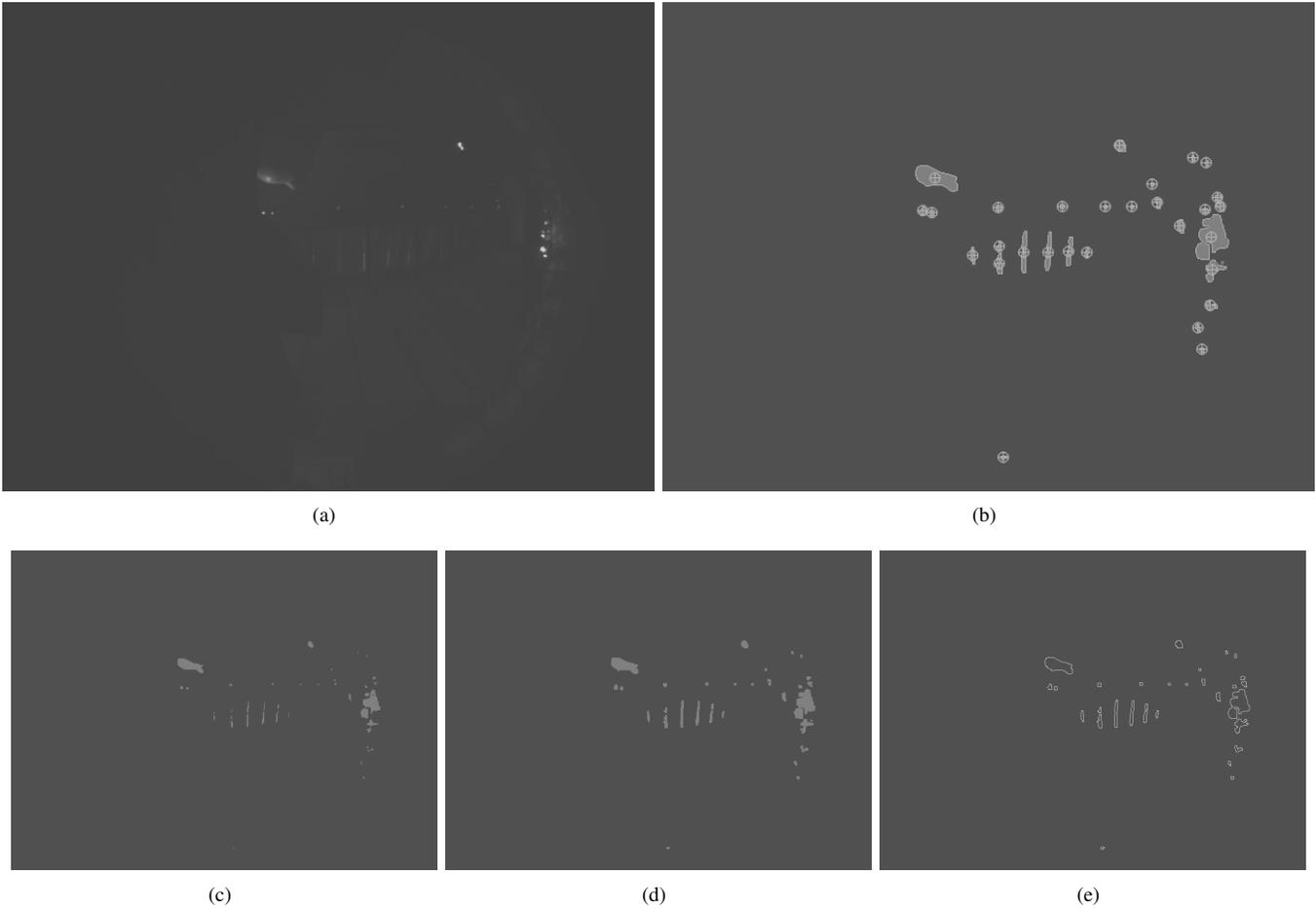


Fig. 8. Detecting landmarks in the omnidirectional filtered image: (a) original image, (b) binary thresholded, (c) dilated, (d) edges, (e) blobs and (f) saturated image.

- *Azimuth*: orientation of an object in the image with respect to the  $x$  axis ( $\varphi$  in Fig. 6).

If a light is pointing directly to the camera, then the acquired image will be saturated (Fig. 9(a)). In such cases, a big blob can be detected and the image has to be processed again using a higher threshold (Fig. 9(b)). This situation is very frequent in the region labelled as B (Fig. 2) because lights can be very close to the camera.

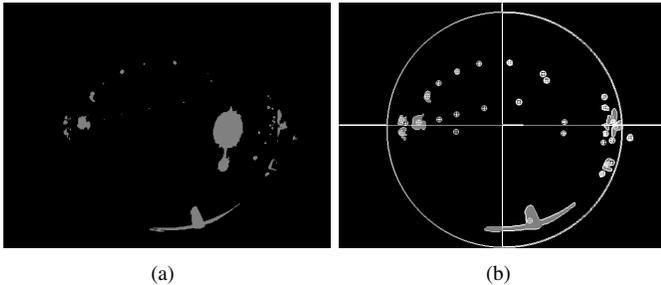


Fig. 9. Postprocessing phase: (a) saturated image and (b) postprocessed image.

#### IV. GLOBAL LOCALIZATION

The localization algorithm for omnidirectional vision is based on a merit function that evaluates each proposed pose

and an iterative process (based on a particle filter) for minimizing that function. The algorithm follows the same idea of a particle filter but it does not use any motion model. For each pose (particle) we build its ceiling map projection ( $Map(P)$ ) using Alg. 1, and compare it with the detected landmarks in the image. The general scheme of the method is shown in Fig. 11.

##### A. Merit Function

**Algorithm 2** Calculate the *Merit Function* ( $N_P$  and  $\varepsilon_P$ )

---

```

Map(P)
for all Beacons j in the image do
  for all Beacons i in Map(P) do
     $\varepsilon(B_{ij}^P) = |Proj(B_i^P) - Detected(B_j)|$ 
    if  $\varepsilon(B_{ij}^P) < THRESHOLD$  then
       $\varepsilon_P = \varepsilon_P + \varepsilon(B_{ij}^P)$ 
       $N_P = N_P + 1$ 
    else
       $\varepsilon_P = \varepsilon_P + THRESHOLD$ 
    end if
  end for
end for

```

---

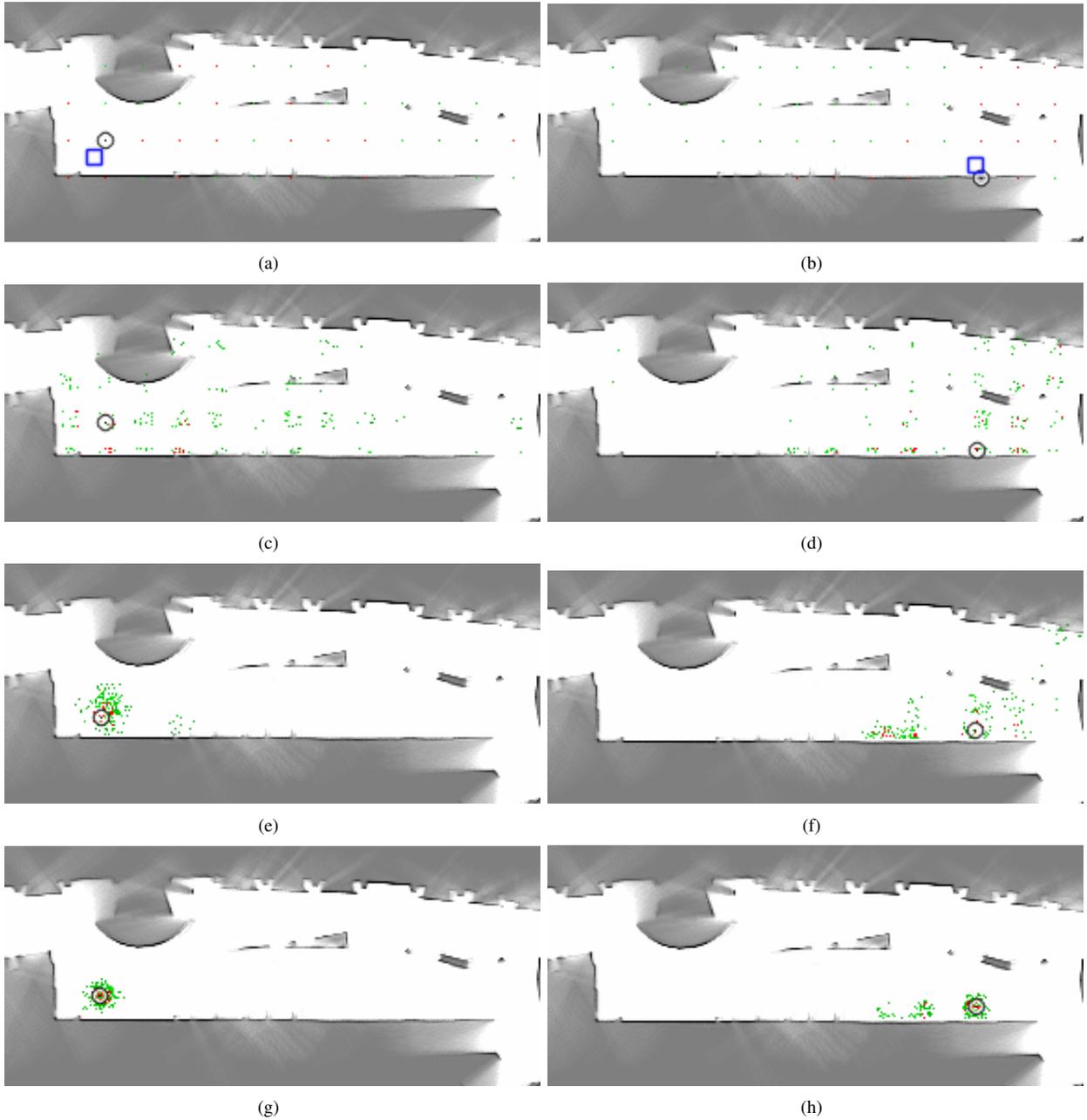


Fig. 10. Results for global localization in the Domus Museum for 2 different poses (left and right columns, respectively). (a,b) Top 200 poses in the initialization stage. Real poses are represented with a square and calculated poses (Eq. 6) with a circle. (c,d) First iteration of the resampling process. (e,f) Third Iteration. (g,h) Last iteration.

The *Merit Function* estimates the similarity between the image acquired with the camera and one ceiling map projection ( $Map(P)$ ). It is defined as:

$$M(P) = \frac{1}{N_P} * \varepsilon_P \quad (5)$$

where  $N_P$ , is the number of landmarks that have been matched between the image and  $Map(P)$ , and  $\varepsilon_P$  is the accumulated error for these associations, i.e., the sum of the errors among each of the detected landmarks on the image and  $Map(P)$  at one pose (see Fig. 12). Before the calculation of the *Merit Function*, a matching process has to be implemented. This matching process establishes the correspondence among the

landmarks detected in the real image ( $B_j$ ) and the theoretical pose of landmarks into the ceiling map projections ( $Proj(B_i^P)$ ). It is based on minimum distance, and an example is show in Fig. 12. The conditions that have to be fulfilled for an association between  $B_j$  and  $Proj(B_i^P)$ , represented by  $B_{ij}$ , are:

- 1)  $|B_j - Proj(B_i^P)| \leq THRESHOLD$
- 2)  $|B_b - Proj(B_i^P)| \geq |B_j - Proj(B_i^P)|, \forall b \neq j$
- 3)  $B_j \longleftrightarrow Proj(B_i^P)$  (uniqueness)

#### B. Minimization Process

The minimization process (Alg. 3) tries to find the pose that

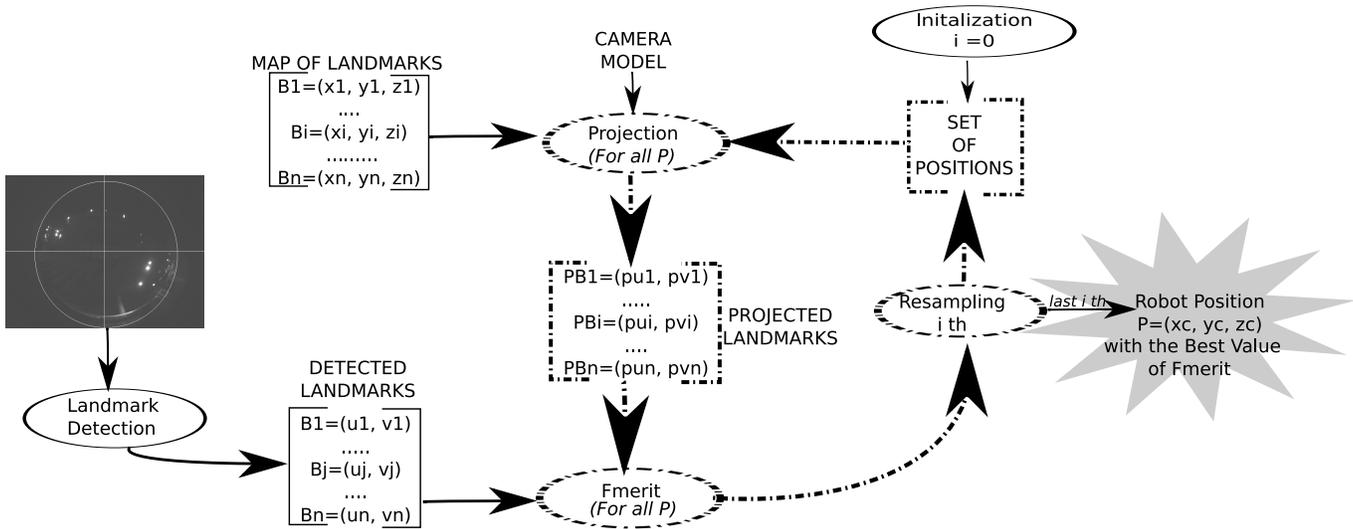


Fig. 11. General scheme of the global localization system for an omnivision image and a map of landmarks.

---

### Algorithm 3 Minimization of the Merit Function.

---

```

Initialize set of particles  $\zeta$ 
repeat
  for all  $P$  in  $\zeta$  do
    Calculate  $Map(P)$  (Alg. 1)
    Calculate  $N_P$  and  $\varepsilon_P$  (Alg. 2)
    Calculate  $M(P)$  (Eq. 5)
    Resample  $\zeta$ 
  end for
until  $Niter < 0$ 

```

---

produces the minimum value of the merit function. To explore all possible poses we use a particle filter, defining a particle as a pose in the environment. The algorithm has two stages, initialization and resampling:

1) *Initialization*: A set of particles ( $\zeta$ ) is uniformly distributed in the environment. Each particle represents a pose of the robot.

2) *Resampling*: This stage is executed iteratively until the particle set is stabilized. For each pose  $P$ , its merit function value ( $M(P)$ ) is calculated applying Alg. 2. Then, the best particles of the set  $\zeta$  are selected to pass to the next stage and the others are discarded. The lower the value of the *Merit Function*, the better the particle.

With this reduced set of particles, a subset of them is selected, random Gaussian noise is added to each of them, and the resulting particles are added to the particle set. In our experiment we have generated 5 new poses, from each pose which belongs to the 15% best particles of set  $\zeta$ . From each of the next 25% best particles, a new particle is added in the same way.

Figure 10 shows the evolution of the set of particles for two images captured from different poses at the Domus Museum. In the last row (Figs. 10(i) and 10(j)) the poses with the best values of the merit function are shown. In Fig. 10(j) particles are grouped in two sets which means that there is a symmetry in the poses of the landmarks.

The pose of the camera is the one that gets the minimum value of the *Merit Function*. This happens when the number of identified landmarks ( $N_P$ ) is the largest and the estimated error ( $\varepsilon_P$ ) is the smallest:

$$C^W = \operatorname{argmin}_P(M(P)) \quad (6)$$

## V. EXPERIMENTAL VALIDATION

The localization system has been tested using data acquired from an omnidirectional camera mounted on a the Pioneer 2-AT robot in an exposition hall (Fig. 2) of the Domus Museum (A Coruña, Spain). The set of images were acquired each second and labeled with the corresponding pose determined with a localization system that uses laser range data. This position information has been used as the ground truth in order to measure the errors of the proposed localization algorithm. We are going to describe two experiments: the first one is a global localization task, and the second one is a kidnapping problem. The experiments were executed using 200 particles and  $Niter$  was set to 6.

### A. Global localization

The experiment was built with a set of 60 images captured in a path of 24 meters long, and the robot moved along that path with a speed of up to 40 cm/s. Fig. 13(a) shows the estimated poses and the ground truth for the experiment.

The localization error (Fig. 14(a)) was calculated as the Euclidean distance between the pose of the robot given by the laser and the pose obtained by the proposed localization algorithm. The maximum error in the experiment was 2.42 m and the average error was 0.53 m.

The orientation error (Fig. 14(b)) was estimated from the absolute error in degrees between the orientation of the laser and the orientation obtained from the localization algorithm. Although the maximum achieved error was of 15 degrees, on average the error was of only 3 degrees.

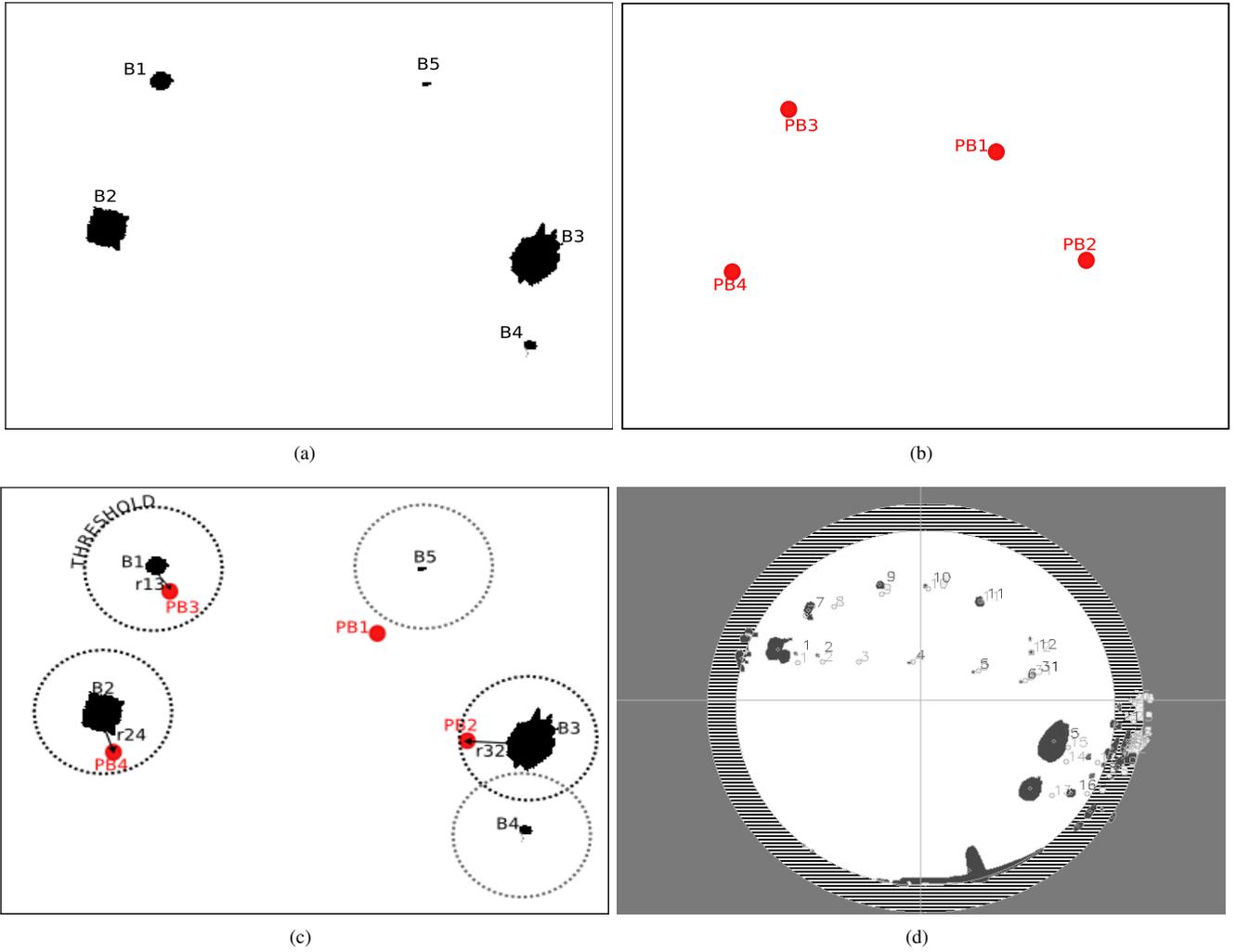


Fig. 12. Matching between a real image and a map ( $Map(P)$ ): (a) Real image. (b) Map. (c) Matching between detected landmarks (beacons)  $B_j$  on the image and the map projection  $Proj(B^P_i)$  (labeled with  $PB_i$ ). (d) Projected map (detected landmarks are shown in black)

The time required for the computation of the algorithm for each image (Fig. 14(c)) depends on the total number of poses checked and the number of iterations. In the experiment, the total number of particles was 702 in the initial phase and 200 for the next iterations. This means 2502 calculations of the *Merit Function* for each image, requiring 300 ms on average on a Pentium 4 3.06 GHz.

Fig. 14(d) shows the number of matched landmarks ( $N_p$ ) and the value of the *Merit function* ( $M(P)$ ) for each pose. We have to highlight that the localization system only needs a reduced number of landmarks for an accurate pose estimation. Moreover, the error in localization does not depend on the number of matched landmarks.

### B. Kidnapped robot

The second experiment was carried out to demonstrate that our system is able to overcome kidnapping situations. We have used a sequence of images that represent jumps of about 400 cm between them in a path of more than 40 m. The estimated trajectories are shown in Fig. 13(b). In this experiment the maximum pose error was 2.59 m and 16

degrees (Fig 15).

## VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a localization algorithm using omnivision. The algorithm is based on the minimization of a *Merit Function* through a particle filter. As the environment is too much crowded in normal conditions, the use of laser or sonar for localizing the robot is not possible. The use of an omnidirectional camera pointing to the ceiling solves this problems. However, irregularities on the floor produce swing movements in the camera, enhancing the difficulties. Nevertheless, and taking into account the experimental results, we can conclude that our system can solve the global localization problem and the kidnapping problem with an average pose error of 0.6 m and 3 degrees.

The most important problem of the system is the symmetry problem. It can produce errors because there are some poses in the environment from which the view of the landmarks configurations is similar, misleading the system. Moreover, the processing time for each image is quite low and, therefore, the system can be executed on real time. In summary, the system can localize the robot in a robust and accurate manner.

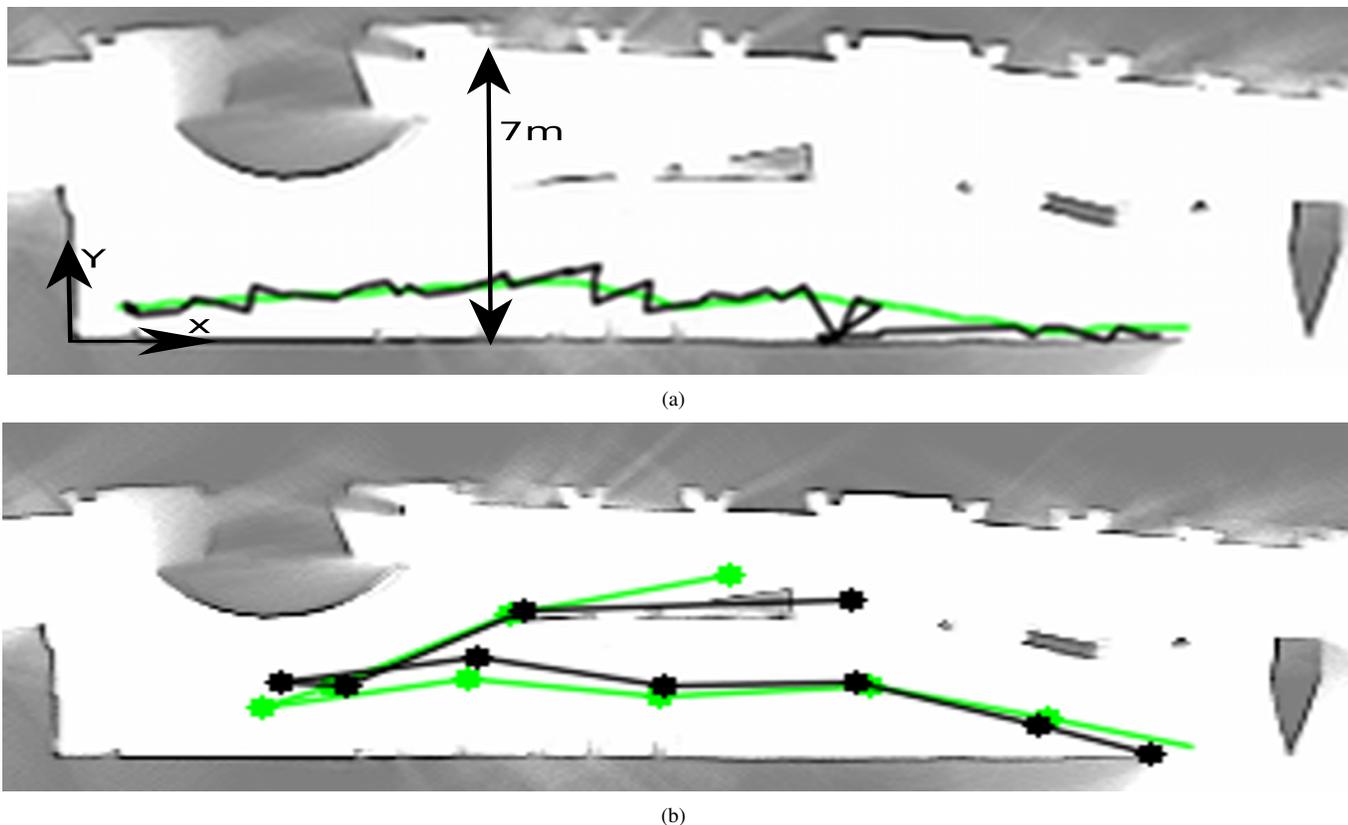


Fig. 13. Path of the robot plotted on the grid map created with laser data. Omnivision localization is shown in black and the ground truth (laser) in green. (a) Global localization experiment. (b) Kidnapping experiment.

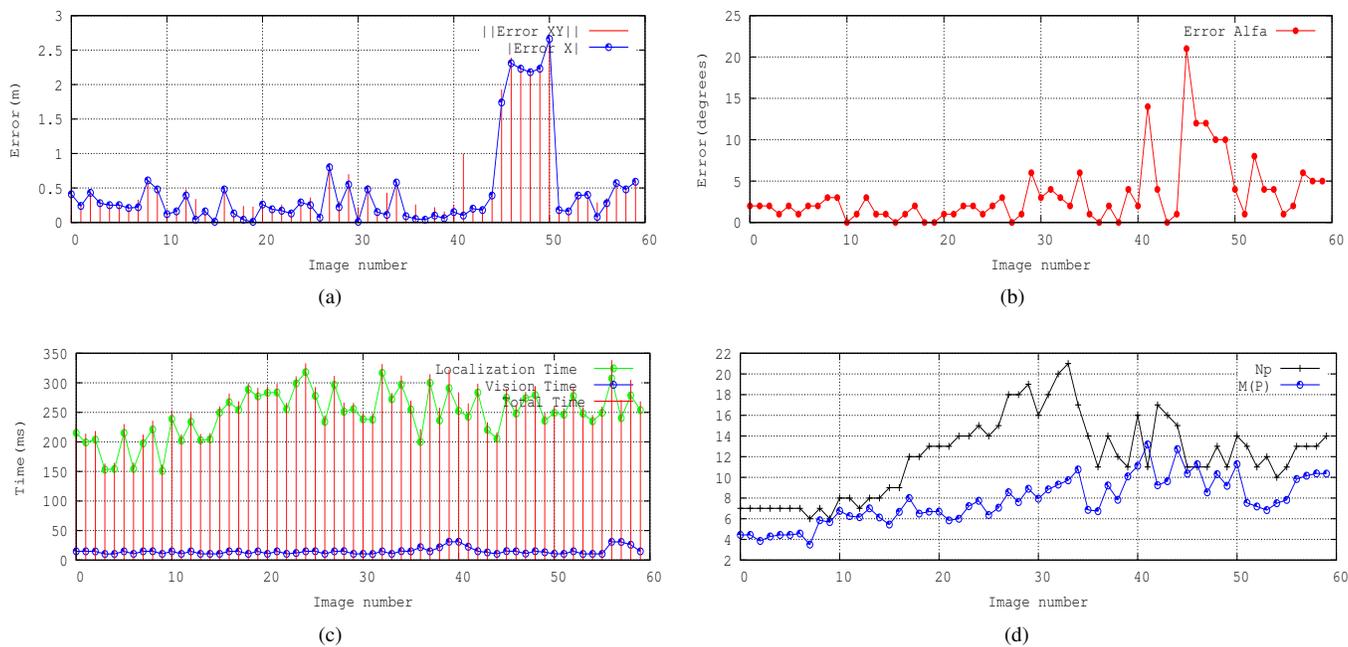


Fig. 14. Errors during global localization: (a) Pose error. (c) Orientation error. (d) Processing time. (e) Number of landmarks and Merit Function value.

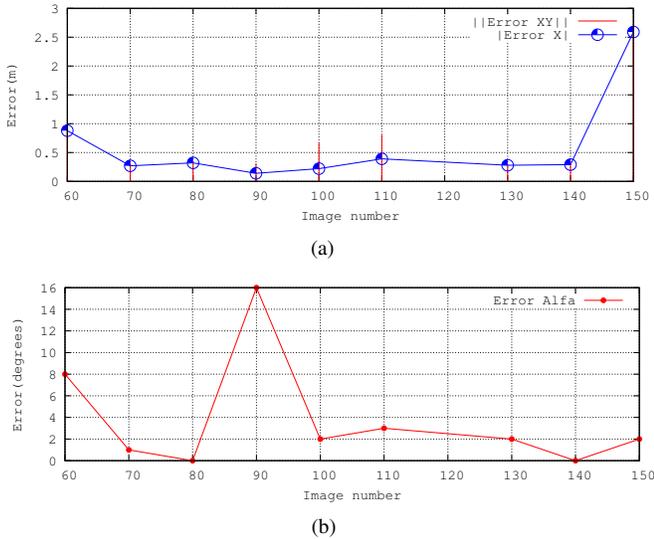


Fig. 15. Errors during kidnapping experiments: (a) Pose error. (b) Orientation error.

As future work, we are planning to reduce the localization error adding the robot action model used in the probabilistic algorithms ([15], [6]). Also, the fusion of information from other sensors like laser or odometry can help in this error reduction. The final objective is to produce a SLAM [8] system for omnivision that allows the localization in any environment without the restriction of having a previous map.

#### ACKNOWLEDGMENTS

Authors want to thank Tomas Pajdla and Hynek Bakstein for their help in the calibration of the camera. This work has been partially supported by the Galician Regional Government and the Spanish Government through grants PGIDITIC206011PR, TIN2005-03844, INCITE08PXIB262202PR and TIN2008-04008/TSI.

#### REFERENCES

[1] H. Andreasson, A. Treptow, and T. Duckett. Localization for mobile robots using panoramic vision, local features and particle filter. pages 3348–3353, 18-22 April 2005.

[2] H. Bakstein and T. Pajdla. Panoramic mosaicing with a 180/spl deg/ field of view lens. In *Omnidirectional Vision, 2002. Proceedings. Third Workshop on*, pages 60–67, 2002.

[3] Eric Brassart, Laurent Delahoche, Cyril Cauchois, Cyril Drocourt, Claude Pegard, and El Mustapha Mouaddib. Experimental results got with the omnidirectional vision sensor: Syclop. *omnivis*, 00:145, 2000.

[4] Zuo L. Cao, Sung J. Oh, and E. L. Hall. Dynamic omnidirectional vision for mobile robots. In *Journal of Robotic Systems*, 1986.

[5] D. Chekhlov, M. Pupilli, W. Mayol, and A. Calway. Robust real-time visual slam using scale prediction and exemplar based feature description. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–7, 17-22 June 2007.

[6] H. Choset, Lynch, M. Kevin, Seth Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press, 2005.

[7] A. Clerentin, L. Delahoche, E. Brassart, and C. Drocourt. Localisations absolues en milieu interieur bases sur la vision omnidirectionnelle. *TS. Traitement du signal*, 20:183–212, 2003. French.

[8] A. J. Davison, Y. Gonzalez, and N. Kita. Real-time 3d slam with wide-angle vision. In *IAV 2004*, July 2004.

[9] D. Fox, S. Thrun, W. Burgard, F. Dellaert, and C. Fernández. Particle filters for mobile robot localization. In *Sequential Monte Carlo Methods in Practice*, New York, 2001. Springer.

[10] E. Menegatti, A. Pretto, A. Scarpa, and E. Pagello. Omnidirectional vision scan matching for robot localization in dynamic environments. In *IEEE transactions on robotics*, volume 22, June 2006.

[11] A. C. Murillo, C. Sagüés, J. J. Guerrero, T. Goedemé, T. Tuytelaars, and L. Van Gool. Hierarchical localization by matching vertical lines in omnidirectional images. In *From Sensors to Human Spatial Concepts workshop, held together with IROS 06*, pages 13–19, 2006.

[12] Jose Pascual Roda, Juan Manuel Saez, and Francisco Escolano. Ceiling mosaics through information-based slam. *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pages 3898–3904, Oct. 29 2007-Nov. 2 2007.

[13] Y. Sun, Q. Cao, and W. Chen. An object tracking and global localization method using omnidirectional vision system. In *5th World Congress on Intelligent Control and Automation*, Hangzhou. P.R. China, 2004.

[14] H. Tamini, H. Andreasson, A. Treptow, T. Duckett, and A. Zell. Localization of mobile robots with omnidirectional vision using particle filter and iterative sift. In *European Conference on Mobile Robots (ECMR05)*, Ancona, Italy, September 2005.

[15] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2003.

[16] Sebastian Thrun, Michael Beetz, Maren Bennewitz, Armin Cremers, Frank Dellaert, Dieter Fox, Dirk Hähnel, Charles Rosenberg, Nicholas Roy, Jamieson Schulte, and Dirk Schulz. Probabilistic algorithms and the interactive museum tour-guide robot Minerva. *International Journal of Robotics Research*, 2000.

# Pedestrian tracking using inertial sensors

Raúl Feliz, Eduardo Zalama, Jaime Gómez García-Bermejo

**Abstract**—A personal navigation system is proposed in the present paper. This system is based on the use of inertial sensors and will be a complement for the Global Positioning System in places where GPS signals are not available. Such places are indoors and outdoors without a clear visibility of the sky (urban canyons, places under dense foliage...).

A known initial position is required for this method for it to be able to estimate the trajectory of the pedestrian using inertial sensors attached to the foot.

The developed method and the approximations taken into account are explained. Finally, the results of several tests carried out are shown.

**Index Terms**— JoPhA, Inertial sensors, GPS, Odometry, MTi, IMU, INS.

## I. INTRODUCTION

Today, LBS (*Location Based Services*), such as car navigators or personal navigators, are based on the use of GPS receptors. However, limitations of the GPS system are well known to users, jamming being the main problem, as shown in (Fig. 1).

GPS signals are not always available because they can be blocked by high buildings, canyons or forests among others. It can be a great problem in certain situations, such as military maneuvers or even for emergency responders. Usually, pedestrians are not in places with a high visibility of the sky, so the reception of the GPS signal will not be as good as the

system needs. A soldier can be in hidden places, and a firefighter can be indoors. Both are places in which a good placement of aerials is not possible. The combined use of inertial sensors and GPS will yield a high accuracy in the estimation of the pedestrian's location even when outdoor and indoor placements are crossed. This work tries to extend the studies done in vehicles and robots with inertial sensors to personal navigation.

The INS (Inertial Navigation System) gives a solution for personal navigation by using inertial sensors such as accelerometers and gyroscopes which have no relation with the place in which pedestrians are located. The IN Systems are a perfect solution for personal navigation due to the frequent GPS signal jamming both indoors and outdoors. The main disadvantage is the increasing error of the system over time and a method able to correct it is required.

Many studies done to determine the location and displacement of people have been focused on the use of inertial sensors and, in some cases, magnetic sensors. Some methods are based on step counters and estimate the distance covered using an average step length [1], [2], [3]. Other methods are based on doubly integrating the acceleration measurements during the walking movement [4], [5], [6]. Yet others try to determine the direction in which the pedestrian is walking to estimate a trajectory. In many studied cases, results obtained using expensive gadgets have not been better than those obtained with simple commercial pedometers.

Simple pedometers focus on counting steps. Based on this step count and an average step length, a pedometer unit can estimate distance traveled [2], [3]. Pedometers do not have the ability to differentiate between different types of gait such as running, shuffling or side stepping. Overshoots tend to occur at slower velocities. Undershoots tend to occur at higher velocities. Pedometers must be calibrated for the stride length of the user and they produce large errors when the user moves in any other way than his or her normal walking pattern.

A very sophisticated pedometer-like approach was introduced by [1]. This system uses a two-axes accelerometer and a two-axes magnetometer located on the user's boot. Step length is estimated from accelerometer readings that are passed through a neural network, and advanced Kalman Filter techniques are aimed at reducing the effect of magnetic disturbances.

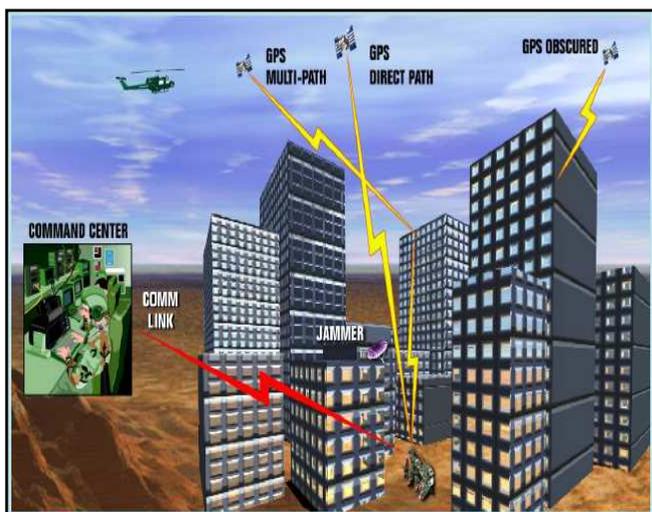


Fig. 1: Problems in the reception of a GPS system



Fig. 2: Xsens MTi-G sensor

Another way of implementing absolute position estimation is computer vision[7], [8]. Images are compared and matched against a pre-compiled database. Computer vision has the advantage that the environment does not need to be modified, but the approach requires potentially very large databases. Work is also being done on so-called Simultaneous Location and Mapping (SLAM) methods, which don't require a precompiled database. However, SLAM systems are not as reliable, may accrue errors over time and distance, and poor visibility and unfavorable light conditions can result in completely false position estimation.

More sophisticated solutions actually measure the length of every stride in real-time. One such solution using ultrasonic sensors attached to the user's boots is explained in [10]. Ultrasonic sensors require a direct line of "sight" between the boots, which may be a problem on rough terrain. Another approach measures the RF phase change between a reference signal located in a waist pack and the one coming from a transmitter located on each boot [11]. A significant drawback of these technologies is that position estimation is restricted to 2-D environments since these systems cannot determine altitude changes and assume that any change is horizontal. Another potential problem is that these technologies use active emissions, which are undesirable for military applications, and they are vulnerable to external interference from the environment or from other units.

Finally, in the current methods, acceleration measurements must be integrated twice to yield position [12]. For this, specific sensors have to be attached to the pedestrian's body. There are many possibilities depending on the part of the body on which the sensor is worn. Authors have attached it to the head [13], to the leg [14], to the waist [15], or to the foot [4], [5], [6], [16]-[19].

The developed method is based on the last point. As explained below, the sensor will be attached to the foot, and acceleration measurements will be corrected when the foot is set on the ground. The main proposals for improving the developed solutions is the use of the compass and gyroscope in the determination of the trajectory, improvements in the correction of acceleration measurements, and the use of the integrated barometer in the height measurements (3-dimensional movement).



Fig. 3: Xsens MTi-G sensor attached to the instep.

## II. PROPOSED METHOD

### A. Procedure

The procedure followed to determine the position of the pedestrian is the double integration of the momentary accelerations measurements. For this work, the inertial unit *MTi* of *Xsens Technologies*, and its variant *MTi-G*, which integrates a GPS and a barometer (Fig. 2), have been used. The *MTi* is a miniature, gyro-enhanced Attitude and Heading Reference System (AHRS). Its internal low-power signal processor provides drift-free 3D orientation as well as calibrated 3D acceleration, 3D rate of turn and 3D earth-magnetic field data. The *MTi* is an excellent measurement unit for stabilization and control of cameras, robots, vehicles and other (un)manned equipment. Moreover, it is an excellent measurement unit for the orientation measurement of human body segments.

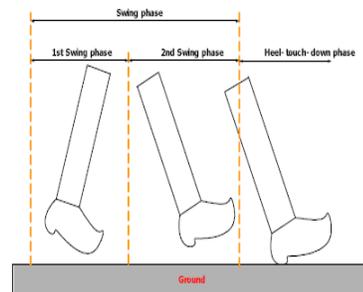


Fig. 4: Foot movement phases

It is well known that the position values obtained by this method are reliable for just a short period of time. This is due to the accelerometer's inherent drift error, which means that when the double integration of the acceleration measurements is performed to get, first the speed and then the position, an error that is accumulated over time is not insignificant, so the estimated position will be far from the actual position.

The method followed to avoid this accumulated error, or at least to reduce it as much as possible, has been to estimate it in each step a pedestrian walks. This ensures a couple of points: first of all, the distance computed by this method will be very close to the actual travelled distance, and the second and most important point is that the small error produced, when speed and position estimation are computed for any step, will not influence the speed and position estimation for the next step. So the inertial system's inherent accumulated error is corrected.

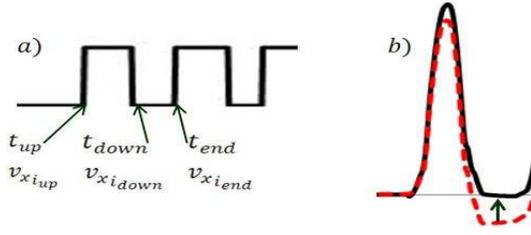


Fig. 5: a) time spent in every step phase. b) Linear speed representation for each step. Without correction (red line) and corrected (black line).

This is why it has been decided to attach the inertial sensor to the instep (Fig. 3). When a person walks, the movement of the foot can be divided into two phases, as seen in (Fig. 4). The first one is the stance phase, which is when the sole is on the ground. The other is the swing phase, which is when the foot is in movement. In the stance phase, linear and angular velocity of the foot must be nil. There is no movement in this phase, so any kind of velocity higher than zero means that there is an error produced by a wrong measurement of the acceleration values by the inertial sensor.

In this phase, linear velocities will be corrected, so the error will not be accumulated for the next step. The linear velocity values are reset to zero and all the measurements obtained for this step are recalculated according to the estimated error. The correction procedure is represented in (Fig. 5).

### B. Error correction description

The detection of the swing and stance stages should be done at the first point. For this task, values of the angular velocity in each of the three axes provided by the gyroscope integrated into the IMU (Inertial Measurement Unit) have been used. Acceleration due to gravity is always present in the linear acceleration measurements. Specifically, total angular velocity has been used.

$\{\omega_{x_i}\}_{i=1\dots n}$ ,  $\{\omega_{y_i}\}_{i=1\dots n}$  and  $\{\omega_{z_i}\}_{i=1\dots n}$  being  $n$  angular velocity measurements in the three axes given by the IMU during the instants of the data acquisition time. Total angular velocity will be:

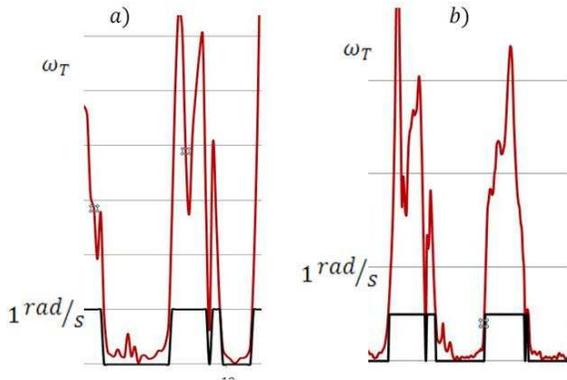


Fig. 6: a) Wrong stop detection. b) Wrong movement detection.

$$\omega_{total_i} = \sqrt{\omega_{x_i}^2 + \omega_{y_i}^2 + \omega_{z_i}^2} \Big|_{i=1\dots n} \quad (1)$$

In ideal conditions, this velocity should be zero in the stance phase. In fact, however, the total angular velocity will not be zero, but it should be lower than a given threshold. If this is true, it is considered that the sole is on the ground. The threshold used in this paper has been  $\tau = 1 \text{ rad/s}$ . A result of one of the tests done is shown in (Fig. 5), where total angular velocity signal  $\{\omega_{total_i}\}_{i=1\dots n}$  vs. step detection signal  $\mathcal{S} = \{s_i\}_{i=1\dots n}$  are shown.

$$\mathcal{S} = \{s_i\}_{i=1\dots n} = \begin{cases} 0 & \forall \omega_{total_i} \leq \tau \\ 1 & \forall \omega_{total_i} > \tau \end{cases} \quad (2)$$

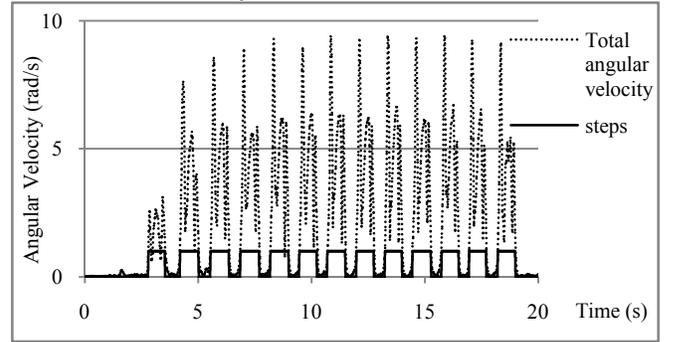


Fig. 7: Angular speed vs. step detection

It is necessary to filter the signal in order to eliminate small fluctuations (momentary values of  $\omega_{total} \leq \tau$  when the movement is actually in the swing phase, and viceversa (Fig. 6)). This filter takes into account the values of the previous and subsequent measurements, and the  $X$  and  $Y$  axes angular acceleration (corresponding to the pitch and roll angles).

Once a signal is obtained, the step-phase of the foot is known and, consequently, the integration and correction of acceleration measurements can be done (Fig.7). Accelerations have to be referred to global coordinates for their integration. This is necessary as the measurements given by the IMU are referred to local coordinates.

Global coordinate conversion is carried out using the rotation matrix which relates local coordinates to global ones. Rotation matrix is defined by the *Euler angles*, which are given by the *MTi/MTi-G* either directly or indirectly. The disadvantages of the direct version are the singularities when the values are close to  $\pm 90^\circ$ . To avoid these singularities, quaternion mode has been used in the data acquisition. This is one of the advantages of using the *MTi/MTi-G*. Working with quaternion allows the mentioned singularity to be avoided and the conversion from quaternion to Euler angles is very easy. Moreover, it is possible to obtain the rotation matrix from the quaternion values.

Then, acceleration in global coordinates are integrated to compute linear velocities for subsequent correction.

The correction method would be as in the following short

mathematical description:

1. Let  $\mathcal{V}_x = \{v_{x_i}\}_{i=1..n}$  be the linear velocity values in the  $X$ -axis to be corrected.
2. The method is applied to each step as follows:
  - a. Let  $t_{up}$  be the start time of the step's movement phase (swing), so that the value of the velocity at this moment will be  $v_{x_{i_{up}}}$ ,  $t_{down}$  will be the moment when the stance phase (in which the foot is on the floor again) starts, and  $v_{x_{i_{down}}}$  will be the value of the velocity at this moment. Finally,  $t_{end}$  will be the moment in which the stance phase ends and the next step starts, with its velocity value  $v_{x_{i_{end}}}$  being a binary signal that satisfies:

$$\mathcal{S} = \{S_i\} = \begin{cases} 0 & \forall i = i_{down} \dots i_{end} \\ 1 & \forall i = i_{up} \dots i_{down} \end{cases} \quad (3)$$

- b. The mean velocity value is computed for the stance phase:

$$\mu = \frac{1}{i_{end} - i_{down}} \sum_{i=i_{down}}^{i_{end}} v_{x_i} \quad (4)$$

- c. The linear velocity values are corrected as follows:

$$v'_{x_i} = v_i - \mu \Big|_{i=i_{down} \dots i_{end}} \quad (5)$$

$$v'_{x_i} = v_i - \mu \cdot \frac{t_i - t_{up}}{t_{down} - t_{up}} \Big|_{i=i_{up} \dots i_{down}} \quad (6)$$

The procedure to obtain  $\mathcal{V}_y = \{v_{y_i}\}_{i=1..n}$  and  $\mathcal{V}_z = \{v_{z_i}\}_{i=1..n}$  would be the same. The graphic representation of the described procedure is shown in (Fig. 8).

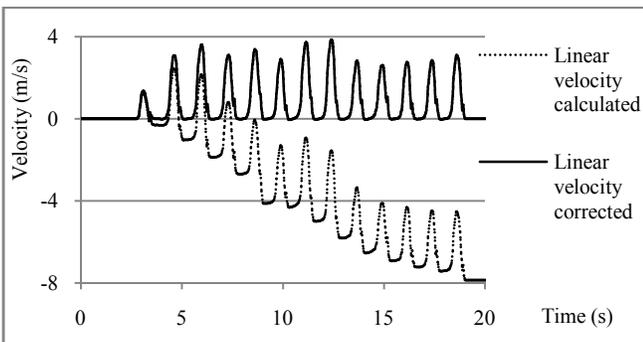
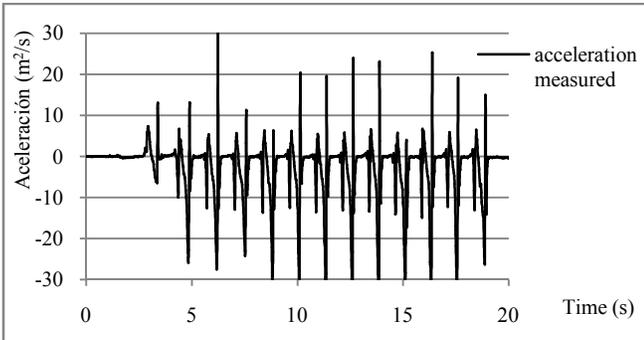


Fig. 8: a) Global x-axis acceleration. b) Obtained and corrected speed from the acceleration shown in (Fig.8-a).

Each step fits a wave of the signal represented. It can be clearly seen how the accumulated error results in the velocity always having a negative value from the sixth step. This would mean that the subject would be moving, which is impossible, due to the given form of the signal. The zone between two consecutive waves (steps) corresponds to the stationary phase. As mentioned above, it is known that in this period the velocity has to be null, so this condition will be used to correct the error.

### C. Yaw/Heading selection.

The Yaw angle, referred to global coordinates, will determine the trajectory followed by the inertial sensor when it is in motion. In the case of the *MTI/MTI-G* sensor, this angle can be obtained through two procedures. On the one hand, the integrated compass provides the angle formed by the sensor and the magnetic north. This is very reliable outdoors, but not so much in the presence of metallic environments (such as metal doors, handrails, frames...) which are present indoors. In this case, the compass will be affected, leading to erroneous measurements.

On the other hand, the integration of angular velocities given by the gyroscope will determine the angular variations between the current instant and an initial known state. The advantage is that the computed angle variation is not affected by magnetic fields in metallic environments. But the main drawback of this procedure is that angular velocity measurement is only reliable over short intervals of time (just a few seconds). This happens because of the inherent error. When the angle variations are computed by integrating the angular velocity measurement, the degree of error increases. In the linear velocity and displacement calculation, this was not a problem because the values were corrected when the foot was stationary. The same procedure is not feasible here, because the foot can rotate when it is on the floor, so it would not be appropriate to assume that the angular velocity at this phase is null.

## III. TESTS

### A. Outdoor tests

The outdoor tests are characterized by the reliability of the measurements obtained through the integrated compass of the sensor.

**Test 1:** In this test, a rectangular trajectory of about 30 m in length is covered. Both trajectory determination methods (gyroscope and compass measurements) are compared. The results of this comparison are shown in (Fig. 9).

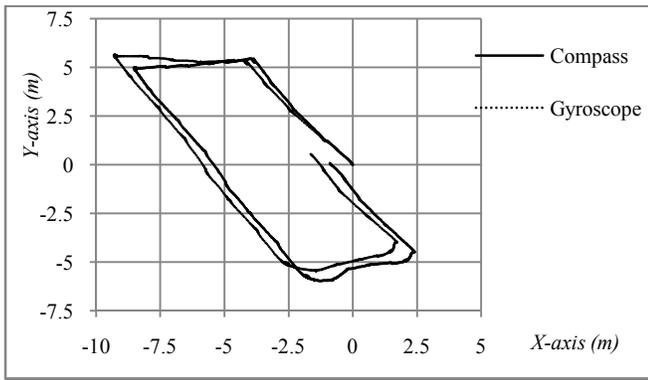


Fig. 9: Trajectory followed by the sensor in the XY plane.

The exact result has not been achieved in either the first nor the second case, as the trajectory was not closed. However, the result shows a higher accuracy in the compass test. The proximity of surrounding metals, such as cars or handrails may have influenced this. The error percentage versus traveled distance is:

- Gyroscope error: 2.5 m → 7.81%
- Compass error: 0.56 m → 1.75%

**Test 2:** In this test, the distance has been increased to 90m. The trajectory is rectangular, as in the previous test, and the comparison of both methods is shown in (Fig. 10).

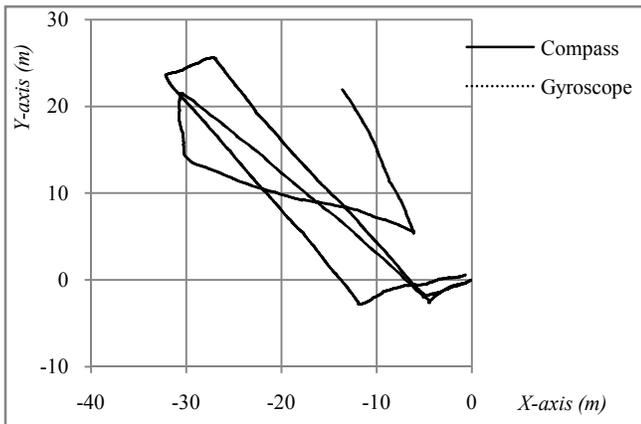


Fig. 10: Trajectory followed by the sensor in the XY plane.

The trajectory followed by the sensor, according to the angle given by the compass, is quite similar to the expected one. This is due to the absence of metallic environments which disturb the measurement. In the case of the gyroscope, once again, the long distance has meant that the result was affected by the presence of an accumulated error, the point where the trajectory turns 90 degrees being significant.

- Gyroscope error: 25.9 m → 28.63%
- Compass error: 0.9 m → 1.06%

**Test 3:** In this test, a block of 680m long has been covered. Only the trajectory computed by the angle given by the compass has been represented. The trajectory obtained according to the gyroscope will be wrong, due to the long distance traveled and the accumulated error. The trajectory has

been superimposed over a satellite image obtained from Google Earth. The result is shown in (Fig. 11).

The trajectory was not closed. This means that an error is present. However, this error, in comparison with the distance traveled, is acceptable.

- Compass error: 15,75 m → 2.31%



Fig. 11: Trajectory followed by the sensor in the XY plane.

*Indoor tests*

There will be two types of indoor test. Those done in places with metallic objects around and those without the presence of such objects.

**Test 4:** The trajectory followed in this test is a square of about 20 meters long in an environment where metals and computer equipment are present (Fig. 12).



Fig. 12: Laboratory where the test was done.

The results are shown in (Fig. 13).

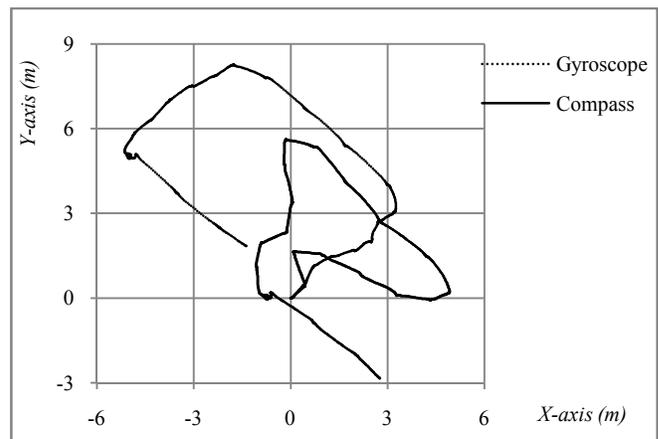


Fig. 13: Trajectory followed by the sensor in the XY plane.



Fig. 14: Corridor without surrounding metallic objects

In this case, the computed trajectory using the angle determined by the compass is far from the expected square. This is because, in the trajectory followed in the test, multiple elements that might distort the measurement of the compass, such as metal gates and computer equipment, have been crossed or passed around. So in this case, it is hoped that the measurement of the gyroscope is more reliable as it is not affected by magnetic fields. Moreover, the short distance traveled means that the error does not affect it disproportionately.

- Gyroscope error:  $2.1m \rightarrow 9.19\%$
- Compass error:  $3.7m \rightarrow 16.28\%$

**Test 5:** In this test, a straight corridor is covered. The tester goes to the end of the corridor, turns 180 degrees and returns to the starting point. The distance traveled is about 30 m. (Fig.14).

The test was done in an indoor environment characterized by the lack of metallic objects around the sensor. The trajectory followed is shown in (Fig 15)

In this case, the measurement of the compass has been reliable even in an indoor environment. The result was predictable because, as mentioned earlier, the scenario was favorable. This is due to the lack of elements that distort the measurement (metallic doors, computers, loudspeaker magnets ...). By contrast, the gyroscope's angle measurement has not led to the expected results, due to the accumulated error, although the trajectory's representation in the first 15 meters is the expected one. The biggest error occurred on the 180-degree turn in mid trajectory.

- Gyroscope error:  $3.8m \rightarrow 12.75\%$
- Compass error:  $0.93m \rightarrow 3.11\%$

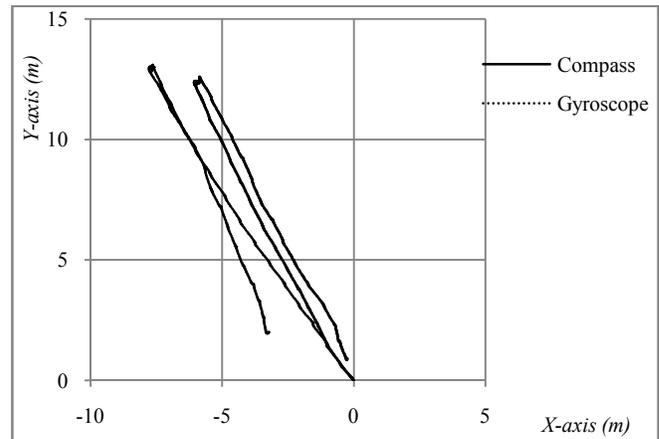


Fig. 15: Trajectory followed by the sensor in the  $XY$  plane according to the compass and the gyroscope.

### B. Test at different heights

This test aimed to show how the sensor works when it describes a 3-dimensional trajectory. This trajectory not only belongs to the  $XY$  plane. It also describes variations in height, such as movements between different floors in a building.

**Test 6:** In this test, an indoor staircase of 5m in height was climbed. The staircase has two flights and a thick metallic handrail, as shown in (Fig. 16).



Fig. 16: Staircase where the test was done.

Both trajectories, obtained according to the accelerometer and the barometer, are shown in (Fig. 17).

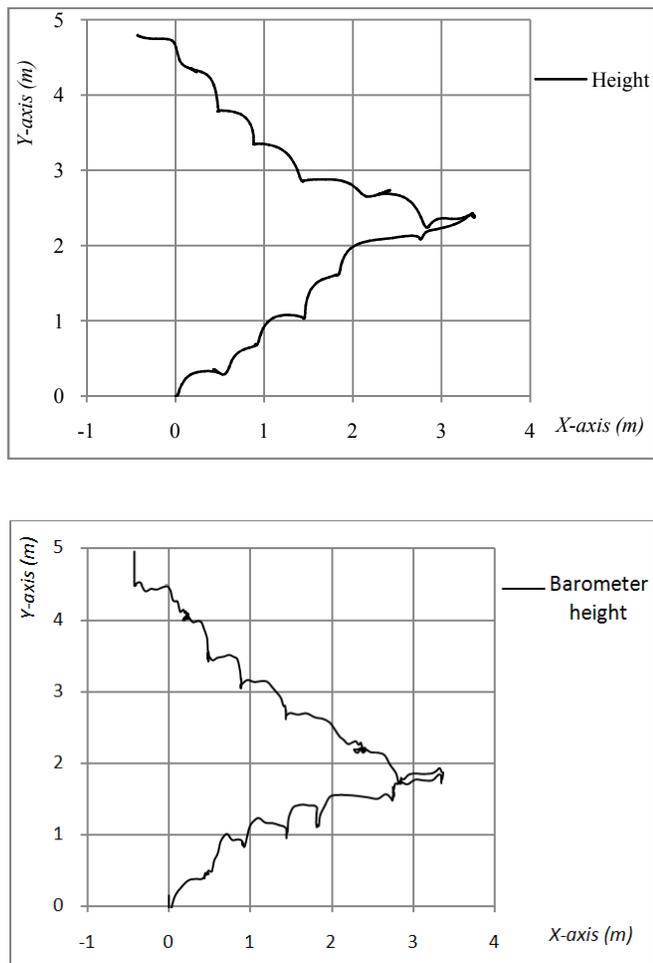


Fig. 17: a) Trajectory followed by the sensor in  $XZ$  plane according to the double integrated  $z$ -axis acceleration. b) Trajectory followed by the sensor in  $XZ$  plane according to the measurement given by the barometer.

The  $XY$  trajectory is affected by the proximity of the handrails, which reduces the accuracy of the compass. In the  $XZ$  plane, the height obtained by both methods is quite similar. The computed signal obtained according to the double integration of  $z$ -axis acceleration shows a smoother evolution. This signal shows the movement described by the foot in a precise way. However, this method has attached an accumulated error, typically inherent in accelerometers, and another error due to the difficulty of subtracting the gravity acceleration from the accelerometer's measurements. On the other hand, the signal obtained from the barometer is rough, and has a slower response time. However, the advantage of this sensor is the lack of accumulated error in all measurements.

Therefore, the variation in height will be sufficiently precise to differentiate jumps in the sensor's measurements to decide on which floor inside a building the person is.

#### IV. CONCLUSION

The aim of this work has been the research and development of a system that will enable personal navigation in areas where the GPS signal is compromised and, at the

same time, will allow indoor navigation. The system is designed to cover three points:

- Ability to obtain a better estimate of a pedestrian's position in real time.
- 3D location of the user in a building (being able to discern the floor where the user is located in a building with several floors).
- To achieve a continuous transition between outdoor and indoor areas.

The method developed for the estimation of the pedestrian's movement starts from the idea of the correction of acceleration measurements during the stance phase (the one in which the sole is on the ground), an idea which was proposed previously by other authors. A number of improvements have been done in this respect.

One is the development of a filter which will allow us to determine if the foot is on the ground or in movement with a higher accuracy. While other authors just correct measures at the low angular velocity point, this filter is able to identify the start and the end of the stance phase and consequently the acceleration measurement correction is done in the whole period. The stance phase's length depends on the walking speed.

The *MTi/MTi-G* sensor provides the possibility of working with highly reliable measurements. Furthermore, it provides the possibility of using calibrated and non calibrated measurements. The calibrated ones are corrected by an *Xsens* own Kalman filter to achieve higher accuracy.

A barometer is used in the test for different heights. When the global  $Z$ -axis acceleration is integrated twice, the problem of the accumulated error is present. It is not possible to subtract the acceleration due to gravity with a high accuracy as the conversion from local coordinates to global ones is not perfect at all. This leads to an accumulated error, especially over long distances.

The sensor integrates a compass and a gyroscope. This means the heading calculation can be done in two ways, as mentioned before. On the one hand, compass measurements, and on the other hand, gyroscope measurements. Moreover, *MTi* has a high accuracy, about a thousandth of all measurements.

Looking at the results obtained, even over long distances, the accuracy of the trajectory estimation has been demonstrated. Nevertheless, there are issues that should be improved in order to achieve a higher reliability in the estimation of trajectories. In future work, these issues will hopefully be developed:

- 1) Study of surrounding magnetic fields in order to estimate the reliability of the compass in the trajectory computation. If the reliability is not verified, gyroscope measurements will be used.
- 2) Comparison between compass and gyroscope measurements to detect wrong behaviors in either, and the development of a Kalman filter which integrates both measurements.

- 3) Integration of the Inertial Unit with a GPS/AGPS system. When a GPS measurement is available, the position estimated by the IMU will be updated in order to eliminate the accumulated error.
- 4) To increase the accuracy when the gravity acceleration value is subtracted from accelerometer measurements. The presence of this value, or a residual part of it, will lead to a wrong estimation of velocity and position.
- 5) Development of a voice-system which allows pedestrians to interact with it in a natural way. The user will be able to help the navigation system by using his/her voice, giving clues like turning left, going straight on... or setting landmarks (door, window...), which allows the position to be estimated in the case of passing through the same place again.
- 6) Use of computer vision to recognize patterns such as doors, stairs, windows...will make an environment simulation (using SLAM) possible with landmarks which allow the position to be estimated in the case of passing through the same place again.

## V. APPLICATIONS

The navigation system is focused on the following applications:

- To achieve a precise location of soldiers indoors, in urban canyons or places under dense foliage ...
- Location of civilians in commercial buildings, finding departments in universities...
- Service based location, as tourist guides in museums.
- Navigation system for blind people.
- Applications for public institutions: tracking the position of firefighters, police officers or to recognize different rooms inside a particular building...

## ACKNOWLEDGMENT

This work has been supported by Deimos Aplicaciones Tecnológicas S.L, for the development of the Personal Navigation System for Lázaro ref.: FIT-340000-2007-169 and Ministerio de Industria, Turismo y Comercio convocatoria profit 2007.

## REFERENCES

- [1] S.Y. Cho, C.G. Park, "MEMS Based Pedestrian Navigation System." *Journal of Navigation*, vol. 59, pp. 135-153, 2006.
- [2] S.E. Crouter, P.L. Schneider, M. Karabulut, D.R. Bassett Jr, "Validity of 10 Electronic Pedometers for Measuring Steps, Distance, and Energy Cost," *Medicine and Science in Sports and Exercise*, 35(8):1455-1460, 2003.
- [3] P.L. Schneider, S.E. Crouter, O. Lukajic, y D.R. Bassett Jr., "Accuracy and Reliability of 10 Pedometers for Measuring Steps over a 400-m Walk," *Medicine & Science in Sports & Exercise*. 1779-1784, 2003.
- [4] L. Ojeda, J. Borenstein, "Non-GPS Navigation for Emergency Responders." *2006 International Joint Topical Meeting: Sharing Solutions for Emergencies and Hazardous Environments*. Salt Lake City, UT, 2006.
- [5] L. Ojeda, J. Borenstein, "Non-GPS Navigation for Security Personnel and Emergency Responders." *Journal of Navigation*, Vol. 60 No. 3, 2007.
- [6] Y. Xiaoping, E. Bachmann, H. Moore, J. Calusdian, "Self-contained Position Tracking of Human Movement Using Small Inertial/Magnetic Sensor Modules". *IEEE International Conference on Robotics and Automation*. Roma 10-14, 2007.
- [7] Y. Liu, Y. Wang, Y. Dayuan, Y. Zhou, "DPSD algorithm for AC magnetic tracking system." *IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, pp. 101-106, 2004
- [8] M. Kourogi, T. Kurata, "Personal positioning based on walking locomotion analysis with self-contained sensors and a wearable camera", *Proc. of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 103-112, 2003.
- [9] C. Galindo, et al. "Vision SLAM in the measurement subspace." *Proceedings of the IEEE International Conf. on Robotics and Automation*, Barcelona, Spain, pp. 30-35, 2005.
- [10] J. Saarinen, J. Suomela, S. Heikkila, M. Elomaa, and A. Halme, "Personal navigation system" *Proc. 2004 IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, vol. 1, pp. 212-217, 2004.
- [11] T. Brand, R. Phillips, "Foot-to-Foot Range Measurement as an Aid to Personal Navigation." *59th Institute of Navigation Annual Meeting*. Albuquerque, NM. (2003).
- [12] S.H. Shin, C.G. Park, J.W. Kim, H.S. Hong, "Adaptative step length estimation algorithm using low-cost MEMS inertial sensors", *IEEE Sensor Applications Symposium* (2007)
- [13] S. Beauregard, "A Helmet-Mounted Pedestrian Dead Reckoning System," in *Proceedings of the 3rd International Forum on Applied Wearable Computing (IFAWC 2006)*, O. Herzog, H. Kenn, M. Lawo, P. Lukowicz, and G. Tröster, Eds. Bremen, Germany: VDE Verlag, March 15-16, pp. 79-89, 2006.
- [14] J. Won Kim, H. Jin Jang, D.H. Hwang, Ch. Park, "A Step, Stride and Heading Determination for the Pedestrian Navigation System", *Journal of Global Positioning Systems*, Vol. 3, No. 1-2: 273-279, 2004.
- [15] M. Kourogi, N. Sakata, T. Okuma, T. Kurata, "Indoor/Outdoor Pedestrian Navigation with an Embedded GPS/Self-contained Sensor System", *IEIC Technical Report (Institute of Electronics, Information and Communication Engineers)*, vol.106, p75-80, 2006.
- [16] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Computer Graphics and Applications*, vol. 25, no. 6, pp. 38-46, 2005.
- [17] S. Beauregard, "Pedestrian Dead Reckoning: A Basis for Personal Positioning," in *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication (WPNC'06)*. University of Hannover, Hannover, Germany: Shaker Verlag, March 16 2006, pp. 27-35.
- [18] F. Cavallo, A.M. Sabatini, V. Genovese, "A step toward GPS/INS personal navigation systems: real-time assessment of gait by foot inertial sensing", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005. (IROS 2005). 2005.
- [19] K. Sagawa, Y. Satoh, and H. Inooka, "Non-restricted Measurement of Walking Distance" , *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, Volume 3, Nashville, TN., October 2000
- [20] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in *Magnetism*, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.
- [21] B. Smith, "An approach to graphs of linear forms," unpublished.
- [22] E. H. Miller, "A note on reflector arrays," *IEEE Trans. Antennas Propagat.*, to be published.
- [23] J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays," *IEEE J. Quantum Electron.*, submitted for publication.
- [24] C. J. Kaufman, Rocky Mountain Research Laboratories, Boulder, CO, private communication, 2004.
- [25] Y. Yoroazu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Transl. J. Magn. Jpn.*, vol. 2, pp. 740-741, August 1987 [*Dig. 9th Annual Conf. Magn. Jpn.*, p. 301, 1982].
- [26] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.



# Developing a Low-Cost Autonomous Indoor Blimp

P. González<sup>1</sup>, W. Burgard<sup>2</sup>, R. Sanz<sup>1</sup> and J.L. Fernández<sup>1</sup>

**Abstract**— This paper describes the design of an autonomous blimp-based robot and its navigation system. The robot was based on a commercial kit and its dimensions were suitable for use in indoor environments. Our main goal was to develop a simple and safe model for evaluating different autonomously controlled navigation techniques. Due to the special requirements of this application, two specific electronics boards to control the blimp and to communicate with the PC ground station were designed and two different altitude controllers and also a controller to maintain distance from obstacles were implemented. Finally, comparative results on both altitude controllers are presented.

**Index Terms**— Blimp, autonomous robot, robot navigation, altitude controller, control avoidance controller, fuzzy logic control.

## I. INTRODUCTION

Non-rigid airships, also known as blimps, are basically unmanned aerial vehicles (UAVs) that use gas (usually helium) balloons. In contrast to a rigid airship, a blimp has no internal structure to maintain the shape of its hull envelope. Rather, its shape is maintained by a higher pressure of the gas. The only rigid components are the driving elements, the fins and the gondola attached to the envelope.

Unmanned blimp robots can be used in both indoor and outdoor environments. The buoyancy force provides an energy-free form of lift, offering a non-traditional approach to long-duration missions for which conventional aircrafts are not well-suited. Miniaturization of sensors and actuators and the development of long-duration batteries have also opened up opportunities for further progress in the development of these small-scale autonomous vehicles.

The first rigid airships, which were constructed in the early 20th century, consisted of a balloon with a metal frame covered by fabric and filled with a gas (helium or hydrogen). These airships were mainly used in wars for military aerial exploration and transportation. Nowadays, however, they are mainly used for advertising and aerial filming. Nevertheless, they have great potential in terms of applications such as search and rescue missions, traffic monitoring, urban planning,

inspection of power lines and pipelines, mineral and archaeological site prospection, law enforcement and telecommunication relay systems [1]. Blimps are well-suited for these applications because their ability to remain stationary for long periods of time in the air enables data to be gathered. Blimps can also be used for research purposes in a variety of applications including ecological, biodiversity and climate research and monitoring in different environments [2].

Our primary interest was the development of a low-cost blimp designed to operate autonomously in indoor environments where different control strategies and navigation paradigms are tested and evaluated. The design of a blimp imposes certain restrictions, primarily because of its limited payload capability, given that a blimp relies on its neutral buoyancy to stay afloat. A key challenge was to build an electronic board that was sufficiently light to be carried on board the blimp. Electronic components were selected to fit our main navigation requirements including limited autonomous navigation capabilities.

This paper is structured as follows. Section 2 discusses related works. The commercial blimp selected and other main components are described in detail in Section 3. Section 4 describes the design and implementation of the navigation system, which basically consists of a fuzzy logic obstacle avoidance controller and an altitude controller. Two different control techniques were experimented with for the altitude controller: a simple proportional–integral–derivative (PID) linear controller and a fuzzy logic controller. The experimental results for these systems are also compared in this section. Finally, Section 5 summarizes our conclusions.

## II. RELATED WORKS

Several researchers have recently developed autonomous robotic systems based on blimps and studied appropriate control paradigms. Much of this research is devoted to large-scale systems, with payloads of kilograms and therefore capable of including a number of sensors (such as cameras) and remaining airborne for long periods of time. Elfes and colleagues [1] presented arguments that favor blimps over airplanes and helicopters as ideal platforms for standard aerial exploration missions. Kantor and colleagues [2] discussed the use of solar energy as a renewable source of power for airships using an outdoor blimp. Hygounenc and colleagues [3] focused on flight control and terrain mapping issues in cooperation between ground and aerial robots.

This work was partially funded by Spanish Ministerio de Ciencia y Tecnología (DPI2005-06210) and the Xunta de Galicia (PGIDIT06PXIC 303194PN).

<sup>1</sup>Dept. of Systems Engineering and Automation, University of Vigo, Spain.

<sup>2</sup>Dept. of Computer Science, University of Freiburg, Germany.

Other authors described the use of blimps in indoor environments. Motoyama and colleagues [4] designed an autonomously controlled indoor blimp and an action-value function for motion planning based on a potential field method, evaluating its effectiveness in a simulated environment [5]. Geoffrey and colleagues [6] used a commercial indoor blimp, concluding that the vertical motor was severely underpowered in tele-operated control. These researchers used a commercial wireless board to send the sensor measurements to the ground computer and a separate board to control the motors. A PC running Linux was used to process sensor data and send control signals to the blimp and a wireless communication unit (WCU) was used for sensor data communication and blimp-side servo control. Hydrogen and helium were used to increase the payload capacity.

We used the same type of blimp as Geoffrey and colleagues [6], but with a different vertical motor, as a more powerful motor and a big propeller were necessary to control altitude so as to have full altitude control in each indoor environment in which the vehicle was tested. We also developed a lighter specific board to control the motors and sensors and thus had more payload capacity for additional components. We only used helium, because hydrogen is flammable and so is very dangerous in indoor environments.

An important navigation problem is automatic control of altitude and of horizontal movement. If the blimp can be maintained at a specific altitude, it can be moved in a horizontal plane. Kadota and colleagues [7] used PID controllers to control blimp altitude and horizontal movement, arguing that blimp trajectory could be unstable in the vertical direction. We also used PID and fuzzy logic controllers to control blimp altitude and evaluated the performance of the control systems developed using the controllers in two different environments.

A second important navigation problem for blimps (and for autonomous mobile robots in general) is obstacle detection and collision avoidance. Green and colleagues [8] used an infrared sensor to detect obstacles: when the collision avoidance system detects an obstacle, the blimp turned 180 degrees to avoid collision. The collision avoidance system does not take account of blimp dynamics, however; consequently, we used an ultrasonic sensor to measure the distance to potential obstacles and implemented a fuzzy logic controller to avoid collisions.

### III. BLIMP DESIGN

The design of an autonomous blimp has certain restrictions arising from the assembled hardware. Various issues were evaluated simultaneously in terms of making appropriate choices. For a blimp system, the higher the volume of the envelope—that is, the higher the ascending force—the higher the possible payload. However, a blimp for indoor applications has to be fairly small. For our blimp system, the goal was to minimize both the size of the blimp and the weight of the necessary onboard hardware.

Due to the limitation on payload, onboard hardware could not be equipped with sensors to measure absolute vehicle position and this limited the autonomous navigation capabilities of our vehicle (as described below).

The main characteristics of the selected blimp components and the design of other elements are described immediately below.

#### A. Blimp selection

Payload, which depends on balloon volume, is a critical aspect that constrains the choice of other onboard components such as the battery.

The blimp that we selected was a hobby radio-controlled (RC) blimp from Plantraco, which comes with an RC transmitter than can be connected via USB to a PC and controlled via a connection to a TCP socket. It has a 52" non-rigid hull made of a light material, achieving 200 grams of estimated payload capacity (Figure 1) [9] [10]. This size is very appropriate for indoor laboratory applications. It can fly in a corridor with people and enter rooms through standard-sized office doors. The payload capacity of this blimp was considered adequate for our purposes.

The blimp has a tri-turbofan gondola with three light DC micro-motors: a vertical motor allows altitude to be regulated and the other two motors control speed and rotation. Since the original vertical motor does not have enough power to properly control altitude, we replaced it with a Futaba 3003 servo motor and a bigger propeller blade. With all these changes the gondola components weighed only 55 grams, leaving 74 grams free for other hardware components including the battery (Table I).

TABLE I  
BLIMP COMPONENT WEIGHTS

Component	Weight (g)
Envelope (52"x37")	60.5
Gondola components	55.0
Fins and propellers	10.5
Onboard hardware	39.0
Battery	35.0
Total blimp weight	200.0



Fig. 1. The commercial blimp selected for this research, with a tri-

turbofan gondola at its base. Balloon size was 52". The vertical motor controls altitude and the side motors control horizontal movements.

The choice of battery was another key aspect because of the weight issue: a very light battery that still provided enough current and autonomy was required. We selected a 350mA lithium polymer battery based on a new technology. Weighing only 35 grams, it supplies a maximum current of 3A and provides around 40 minutes of autonomy for the blimp. The nominal voltage level of 11V was too high for the electronic components, and so a power management circuit to provide the required voltages (8V and 5V) was designed. A circuit to control voltage level was also added to avoid damage to the battery when the voltage was under 9V.

### B. Ultrasonic sensors

Two ultrasonic sensors provided the autonomous blimp with information on the environment. An ultrasonic, lightweight SRF05 sensor (with a resolution of 1 mm and a very narrow beam) was mounted facing downwards at the bottom of the gondola to measure the distance from the blimp to other objects. Sensor measurements were integrated by means of a Kalman filter which sequentially estimated blimp altitude. The other ultrasonic sensor, located in the forward-facing part of the balloon, was used by the obstacle avoidance controller. In this case, we selected a lightweight SRF10 ultrasonic sensor with an operating range of up to 6 meters that could be connected to the microcontroller via a standard I2C bus interface.

### C. Electronic components

An electronic board was custom-designed for this application because no commercial board met with our requirements (Figure 2). An appropriate communication system was also designed and implemented. The main reasons motivating this approach were the following:

- A new motor speed control was necessary, firstly, because the blimp could not be controlled using the original circuit, which only worked at maximum speeds, and secondly, because we could not control the new vertical motor with the original board.
- Good wireless communication coverage with a lightweight circuit was desirable, so lighter components for communications between the vehicle and the PC ground station were selected, given that a wireless access point or a wireless router could not be used because of the limited payload capability.
- A bidirectional communication link to send and receive data from the blimp was required in order to be able to send data to and from the blimp and the PC ground station (in other words, we needed to close the control loop).

For the above reasons, two specific boards were designed: a gondola onboard card and a PC interface card.

*Gondola onboard card.* This board included all the electronic components necessary to control the three motors

and the onboard wireless communication link (see Figure 3). The main components are:

*Power supply unit.* It consists of a power regulator which provides two stabilized voltage levels: 8V and 5V and a battery circuit which prevents total discharge of the battery.

*Microcontroller unit (MCU).* It is based on a PIC microcontroller, which remains the best balance of cost, processing power, complexity, and power consumption. The PIC 16F873 microcontroller from Microchip appears to be a good choice. It has USART, analog ports, and I2C ports required to connect sensors and motor drivers. Some other inputs and outputs are used to communicate with the wireless transceiver.

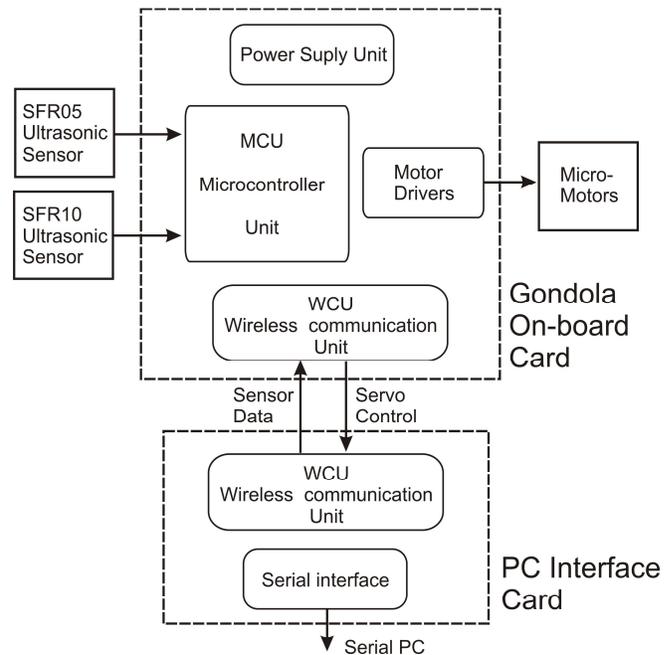


Fig. 2. Block diagrams of the gondola onboard and PC interface cards.

*Motor drivers.* They are necessary to control the speed of each motor. The drivers are based on a L293B circuit, which offers 1A per channel and can modulate the voltage and control the motor speed. The pulse-width modulation (PWM) is directly controlled by the microcontroller.

*Wireless communication unit (WCU).* It is employed to transmit data between the blimp and the PC ground station. It consists of a serial wireless modem that works in the 400 MHz band, and it is based on an ER400TRS transceiver. It also includes a buffer to protect the transceiver. Data are transferred in real time at a speed of 19,200 bps. The WCU sends sensor data to the ground PC and receives servo positions from the ground PC to control the blimp (Figure 2).

*PC interface card.* The blimp was remotely controlled using a customized computer board that also contained a WCU. We also developed a software package module with generic functions that enabled easy control of the blimp by the programmer.

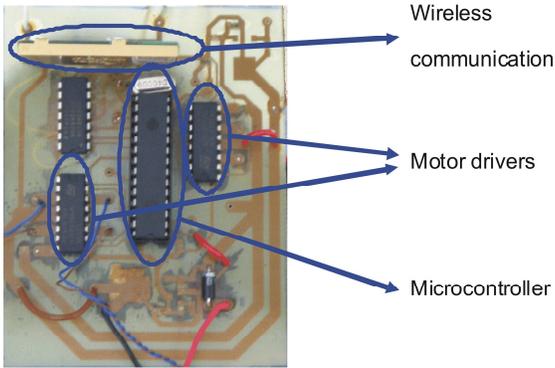


Fig. 3. Electronic components of the gondola onboard card.

The ground computer could both write and read from the WCU using these functions. The design of the function interface was such that the programmer was not required to deal with serial port characteristics and the communication protocol between the blimp and the computer. The interface had ten user-friendly functions so that the programmer could develop software to move the blimp in few minutes.

#### D. Onboard software

The onboard software processed the sensor measurements and sent them to the PC ground station. These data were received by the PC ground station and used to compute new control signals for the gondola micromotors. The WCU in the PC interface card sent control data to the onboard WCU, which were used by the microcontroller to set the corresponding commands for the micromotors.

### IV. BLIMP NAVIGATION SOFTWARE

The navigation software provided limited autonomous operation of our blimp due to:

- The lack of odometry.
- Misalignment caused by wind gusts and temperature changes.
- The non-linear nature of propeller action.
- Blimp operation in three dimensions.

The control system for our autonomous blimp was designed basically to keep the blimp moving in a straight line if its path was clear, maintaining the desired horizontal speed and reference altitude. When a head-on obstacle was detected the control system attempted to maintain a certain distance from it. Assuming a low speed, the control problem could be decoupled in two sub-systems describing motions in both vertical and horizontal planes [11].

Although some blimp control approaches are based on the vehicle mathematical model (see [11] and [12], for example), we preferred not to implement an analytical controller because the blimp has complex dynamics due to its nonlinear characteristics and environmental influences (air gusts, temperature, altitude references, etc). We implemented both

PID and fuzzy logic controllers. PID parameters can be experimentally adjusted on the basis of well-known methods and fuzzy logic controllers can be easily tuned on the basis of trial and error.

Our control system was composed of two different controllers that would ensure safe autonomous navigation in indoor environments: an altitude controller and a collision avoidance controller. Both controllers were implemented in the PC ground station using the distance measurements sent by the WCU, with computed control signals sent back to the onboard microcontroller.

The reference blimp altitude was specified in the program user interface, with the altitude control algorithm endeavoring to automatically maintain the blimp at this altitude. Controlling vertical motion reduced blimp movements by one degree of freedom. Two different altitude controllers were implemented and tested.

The collision avoidance controller enabled control of the horizontal movements of the blimp so as to avoid frontal collisions. Only a fuzzy controller was implemented for this purpose because the results for PID controllers were poor. When the front sensor did not detect an obstacle in its path, the controller navigated the blimp along a straight line; when an obstacle was detected, the controller kept the blimp at a certain distance from the obstacle.

#### A. PID altitude controller

The current altitude of the blimp was measured by the SRF05 ultrasonic sensor and then sent to the PC. Sensor errors were corrected with a Kalman filter, thus obtaining the estimated altitude. The filter parameters depended on sensor characteristics, the dynamic model of the blimp and the previous measurements, but in our case they were tuned experimentally. We did not make use of the Kalman filter variance because of real time restrictions in controlling motor action.

The first altitude controller implemented was a PID-type controller. Control actions were calculated as:

$$u_{PID} = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

where  $K_p$ ,  $K_i$  and  $K_d$  were parameters experimentally calculated using the Ziegler-Nichols method and  $u_{PID}$  was the command signal to the vertical propeller, responsible for up-and-down movements of the aerial vehicle.

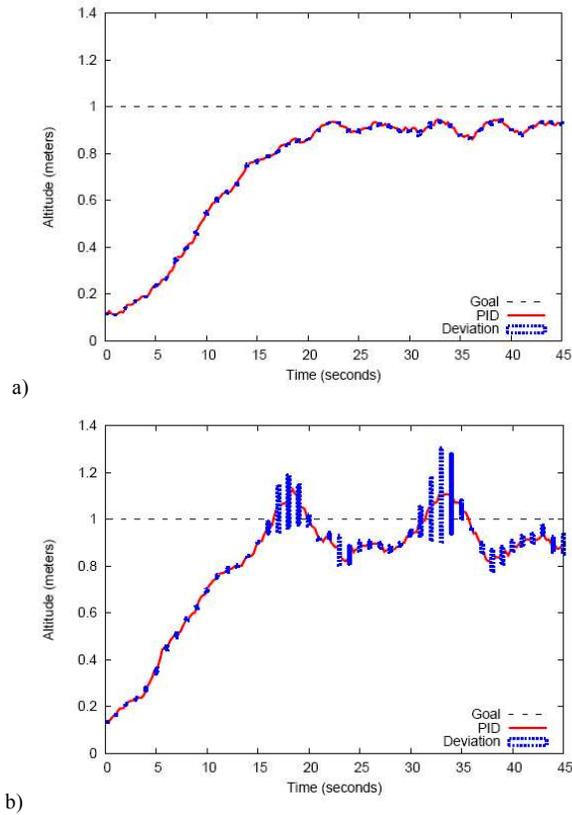


Fig 4. Behavior of the PID altitude controller when the altitude reference is set to 1 meter. a) Experiments in environment 1. b) Experiments in environment 2.

Blimp altitude in Figure 4 is depicted in red, while average values are represented in blue. The figure shows the blimp PID altitude controller behavior for two different scenarios. The first was a corridor in Building 79 of Freiburg University (Figure 5), while the second was a computer laboratory in the same university (Figure 6). The computer laboratory had a door that, when opened, could alter the draught inside. Eight similar experiments were performed in both environments, all commencing with the blimp lying on the ground and then moving to its target position, one meter above the ground.



Fig 5. Experimental environment 1: a corridor in Building 79, Freiburg University.

The experiments have shown diverse results using the same controller in these two environments. An overall good performance can be observed during the experiments carried out in environment 1. However, the same PID controller in environment 2 has shown an oscillating behavior with an important deviation from the mean. Although it can reach the altitude reference, there is a significant error as it is shown in Figure 4b.

Note that the controller response in the second environment could be improved by tuning PID parameters. In fact, these parameters had to be recalculated to take account of changes in environmental conditions while the blimp was navigating.



Fig 6. Experimental environment 2: a computer laboratory in Freiburg University.

### B. Fuzzy altitude controller

A non-linear altitude controller based on fuzzy logic was also implemented. Fuzzy logic uses fuzzy sets to model

designer knowledge about the system to control, with knowledge representation modeled using fuzzy rules. This kind of controller has several advantages because it does not need to recalculate parameters when environmental conditions changes.

A fuzzy controller is composed of a knowledge base and an inference engine. The knowledge base contains rules and linguistic variable descriptions, while the inference engine generates a control action as a function of state variable values in a given time instant.

The altitude fuzzy logic controller in our blimp had two inputs: altitude error and estimated current vertical speed. Altitude error was the difference between the desired altitude and current altitude (Figure 7). A change in altitude error indicated whether the aerial vehicle was approaching the reference altitude or moving away from it. The controller output was the vertical motor command. Note that the fuzzy controller structure differed from the PID controller in that the latter has only one input, which ensures better and more accurate altitude control.

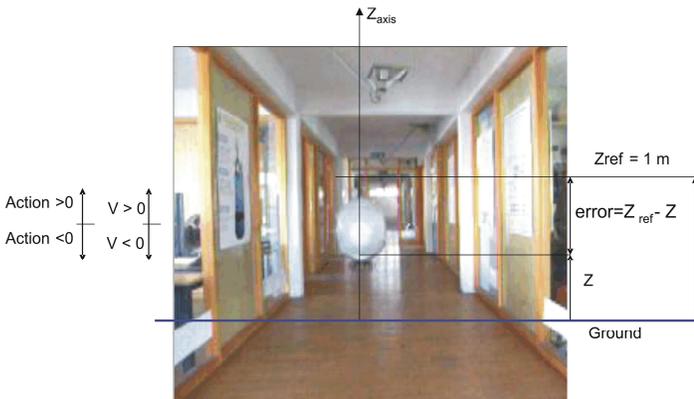


Fig. 7. Variable relations regarding altitude control of the blimp. Zref is the altitude reference; Z is the estimated height using a Kalman filter. Other variables are shown on the left.

The fuzzy logic controller is characterized by a set of linguistic variables and fuzzy if-then rules. All input and output linguistic variables have a finite number of linguistic values with membership functions that are empirically defined after exhaustive simulation studies. The linguistic values representing the linguistic variable *altitude error* and *vehicle speed* are *very negative* (VN), *negative* (N), *zero* (Z), *positive* (P) and *very positive* (VP). Linguistic values describing the output linguistic variable are *very negative 3* (VN3), *very negative 2* (VN2), *very negative* (VN), *negative* (N), *zero* (Z), *positive* (P), *very positive* (VP), *very positive 2* (VP2) and *very positive 3* (VP3). Their corresponding membership functions are shown in Figure 8.

Fuzzy rules describe the controller behavior in terms of relationships between input and control variables. A rule is usually of the type:

*If  $x_1$  is  $A_1$  and  $x_2$  is  $A_2$  then  $y$  is  $B$ ,*

where  $x_i$  and  $y$  are, respectively, input and control linguistic variables, and  $A_i$  and  $B$  are linguistic terms.

The altitude controller can be described with a small set of rules, for example:

*If altitude error is negative and vertical speed is very positive then motor command is positive.*

Figure 9 depicts, in tabular form, the fuzzy rule set used to generate the motor command. These rules endeavor to maintain the blimp at a specific height and represent human expertise on how to control the system. Note that the rule descriptions were developed on the basis of multiple experiments.

Figure 10 depicts one of the situations described in the table of Figure 9. In this situation, the blimp is descending and the altitude error is significant. In the position shown in the figure, the input variables have the linguistic labels VN for velocity and VP for altitude error, respectively. So, the command signal value must be VP3, as shown in the fuzzy rule table depicted in Figure 9.

Note that one set of rules describes when motor action must be null. These rules have a dead band to avoid continuous motor action.

Experimental results for the fuzzy altitude controller are depicted in Figures 11a and 11b. The blimp showed only slight oscillations in both environments and deviations from the mean, shown in blue, were very small. A significant number of tests carried out in different circumstances led to similar good results in both environments.

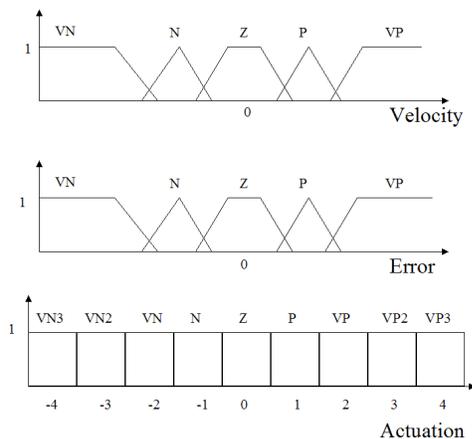


Fig 8. Fuzzy linguistic variables for the fuzzy altitude controller.

		Error				
		VN	N	Z	P	VP
Velocity	VN	Z	P	VP	VP2	VP3
	N	N	Z	P	VP	VP2
	Z	VN	N	Z	P	VP
	P	VN2	VN	N	Z	P
	VP	VN3	VN2	VN	N	Z

Action

Fig 9. Fuzzy rules for altitude control.

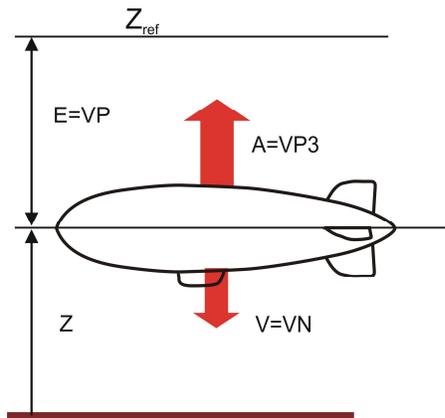
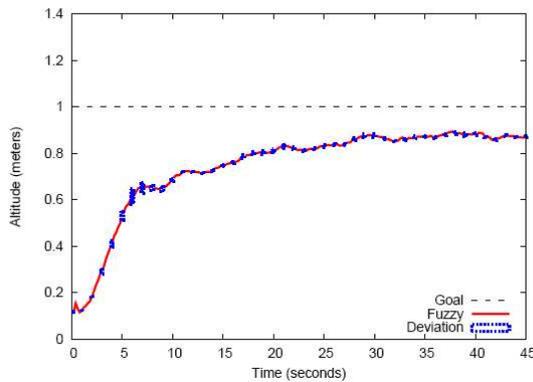
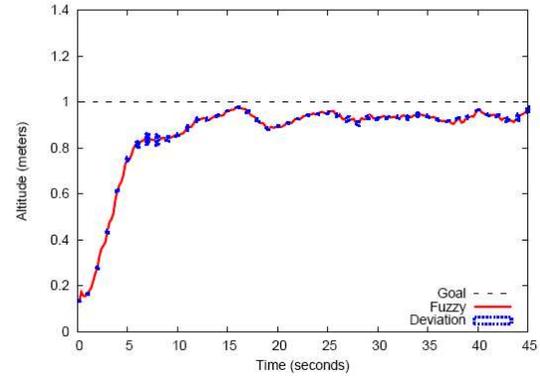


Fig. 10. Behavior of the fuzzy altitude controller when the blimp descends.



a)



b)

Fig. 11. Behavior of the fuzzy altitude controller when the altitude reference is set to 1 meter. a) Experiments in environment 1. b) Experiments in environment 2.

### C. Collision avoidance controller

The collision avoidance system should cause the vehicle to stop reliably when the frontal distance sensor detects an obstacle in the vicinity. In such circumstances, the horizontal navigation speed is changed by the collision avoidance controller. Only the frontal sensor no longer detects an obstacle close to the blimp, horizontal blimp speed is reset to a certain value. Different approaches can be used to implement the collision avoidance controller.

We again chose a fuzzy logic approach, based on a PID controller which demonstrated oscillating behaviors in almost all the experiments carried out in both environments. The goal of this controller was to keep the blimp at a safe distance from frontal obstacles. Controller inputs were frontal distance and estimated speed and controller output was a speed index for the horizontal motors.

Fuzzy rules for the collision avoidance controller are shown in Figure 12. Figure 13 depicts a possible situation in which the blimp is very close to an obstacle detected by the frontal sensor. The blimp is heading straight for a vertical wall; speed is  $V=VN$  and the distance error value is  $E=VP$  (near the wall). In this situation, which is critical for the blimp as it is likely to collide with the wall, the fuzzy collision avoidance controller transmits the maximum control command ( $A=VP3$ ) to the motors, thereby transmitting the maximum opposite power to ensure that the blimp avoids the collision. However, if this speed is used for a long time the blimp develops inertia and it becomes impossible to stop it in the desired reference. When the speed is reduced so as to be comprised within the linguistic variable  $V=N$ , the blimp does not need maximum power and control action is reduced to  $A=VP2$ . These few rule examples illustrate how the controller works.

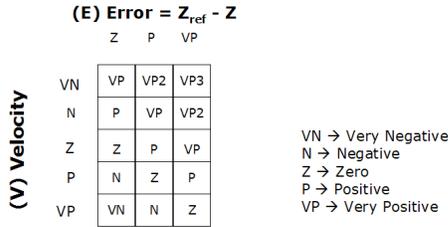


Fig. 12. Fuzzy rules for the collision avoidance controller.

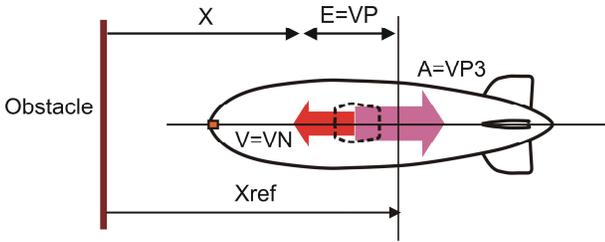


Fig. 13. Behavior of the fuzzy collision avoidance controller when an obstacle is detected.

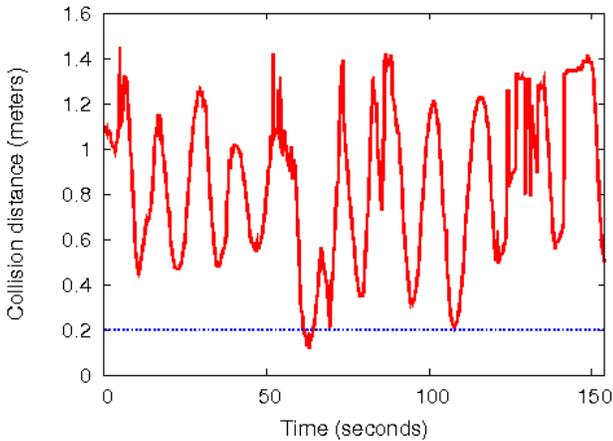


Fig. 14. Behavior of the blimp in a corridor. The blue line is the safety distance from the floor.

Figure 14 shows the blimp distance to a vertical wall while it is navigating in environment 1. Initially, it maintained a constant horizontal speed. When it approached a wall at the end of this corridor, the obstacle avoidance controller reduced the speed accordingly. In the experiment, the vehicle maintained a security distance of about 0.2 meters away from the obstacle without crashing at any time.

V. CONCLUSIONS

In this paper we have described the construction of a small blimp based on a commercial kit, with the main goal of designing and implementing different control navigation techniques. It was necessary to change the vertical motor of the commercial blimp and develop onboard control hardware, including a wireless radio connection with a PC ground station. Due to the vehicle's small payload only two distance

sensors were used for navigation; the small payload also did not permit inclusion of an inertia sensor or miniature wireless camera usually used for global position calculation [6].

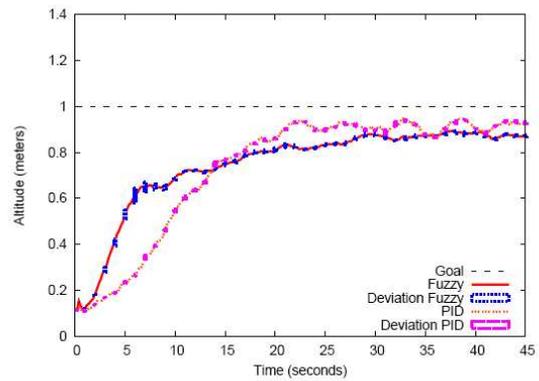
We also implemented onboard and PC programs to control the blimp with the PC ground station, with the communication software module capable of receiving onboard sensor data and sending commands to the blimp (as well as facilitating future application developments).

Two different controllers were implemented. An altitude controller maintained the blimp at a certain distance from the floor and a second controller avoided obstacles in the path of the blimp. The altitude controller was implemented using two different approaches: PID and fuzzy logic. A first experiment was designed to compare the two controllers operating in the same conditions, so as to determine which controller was better suited to controlling blimp altitude.

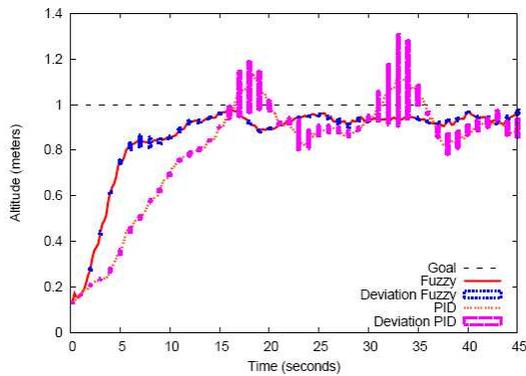
In the first scenario, both controllers were tested on an alternating basis in one environment (a corridor), an experiment that was repeated eight times. Similarly, the blimp was tested in a second environment (a computer room) with the same perturbations on the blimp for both controllers.

Comparative results show that the fuzzy logic controller produced balanced behavior in either of the two environments. The PID altitude controller performed slightly better than the fuzzy logic controller in environment 1 (see Figure 14a) but its performance in environment 2 was significantly poorer than that of the fuzzy logic controller. A better PID controller in other environments would require online adjustment of PID parameters. Using a fuzzy logic controller, on the other hand, does not affect the behavior of the blimp too much and the controller parameters do not require modification.

A second fuzzy logic controller, designed to avoid head-on collision with obstacles while the vehicle was navigating in the indoor environment, showed good experimental performance.



a)



b)  
Fig. 13. Comparative results for the PID and fuzzy logic controllers. a) Experiments in environment 1. b) Experiments in environment 2.

Note that we could not measure the absolute position of our vehicle with the onboard sensor due to the low payload of the blimp. This limited blimp navigation in a desired direction without global path planning. For this reason we are developing a new design with a bigger balloon than described in this paper that will enable new vehicle control functions to be incorporated [13].

#### ACKNOWLEDGMENTS

We would like to thank the people of the University of Freiburg (Germany) for their assistance during Pablo González's visit to the Autonomous Intelligent Systems Laboratory.

#### REFERENCES

[1] A. Elfes, S. Siquiera, M. Bergerman, and J. Guimaraes, "A semi-autonomous robotic airship for environmental monitoring missions," in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, 1998.

[2] G. Kantor, D. Wettergreen, J. P. Ostrowski, and S. Singh, "Collection of environmental data from an airship platform," in *Proceedings SPIE Vol 4571, Sensor Fusion and Decentralized Control in Robotic Systems*, 2001.

[3] E. Hygounenc, I.-K. Jung, P. Soueres and S. Lacroix, "The autonomous blimp project of LAAS-CNRS: Achievements in flight control and terrain mapping," *The International Journal of Robotics Research*, vol. 23, n. 4-5, pp. 473-511, 2004.

[4] K. Motoyama, H. Kawamura, M. Yamamoto, and A. Ohuchi, "Development of autonomous blimp robot with intelligent control", in *Entertainment Computing: Technologies and Applications*, Eds. R. Nakatsu and J. Hoshino, Kluwer Academic Pub. 2003, pp. 191-198.

[5] K. Motoyama, H. Kawamura, M. Yamamoto, and A. Ohuchi, "Design of evaluation function in motion planning for autonomous balloon robot", in *Proceedings of 2003 Asia Pacific Symposium on Intelligent and Evolutionary Systems*, 2003.

[6] G. A. Hollinger, Z. A. Pezzementi, A. D. Flurie, and B. A. Maxwell, "Design and construction of an indoor robotic blimp for urban search and rescue tasks", *Swarthmore College Senior Design Thesis*, Spring 2005.

[7] H. Kadota, H. Kawamura, M. Yamamoto, T. Takaya, and A. Ohuchi, "Vision-based positioning system for indoor blimp robot", *Proceedings of the 4th International Conference on Advanced Mechatronics-Toward Evolutionary Fusion of IT and Mechatronics*, 2004.

[8] W. E. Green, K. W. Sevcik and P. Y. Oh, "A competition to identify key challenges for unmanned aerial robots in near-earth environments", in *Proceedings. 12th International Conference on Advanced Robotics ICAR '05*, July 2005.

[9] Balloon shop. <http://www.heo.com>.

[10] Plantraco shop. <http://www.plantraco.com>.

[11] S. van der Zwaan, M. Perrone, A. Bernardino, and J. Santos-Victor, "Control of an aerial blimp based on visual input", in *Proceedings of the 8th International Symposium on Intelligent Robotic Systems SIRS'00*, 2000.

[12] N. Rooz and E. N. Johnson, "Design and modeling of an airship station holding controller for low cost satellite operations," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2005.

[13] A. Rottmann, M. Sippel, T. Zitterell, W. Burgard, L. Reindl, and C. Scholl. "Towards an Experimental Autonomous Blimp Platform", in *Proceedings of 3rd European Conference on Mobile Robots (ECMR '07)*, September 2007.



# A Cognitive Approach to Multimodal Attention

Raúl Arrabales, Agapito Ledezma and Araceli Sanchis

**Abstract**—An adaptive attention mechanism is a requirement when an autonomous robot has to deal with real world environments. In this paper we present a novel cognitive architecture which enables integrated and efficient filtering of multiple modality sensory information. The proposed attention mechanism is based on contexts that determine what sensorimotor data is relevant to the current situation. These contexts are used as a mean to adaptively select constrained cognitive focus within the vast multimodal sensory space. In this framework, the focus of attention can be directed to meaningful complex percepts, thus allowing the implementation of higher cognitive capabilities. Sonar, contact, and visual sensory modalities have been used in the perception process, and the motor capability of the physical agent is provided by a differential wheel drive system. The testing of this artificial attention approach, carried out initially in the domain of counterpart recognition and chasing, has demonstrated both a great decrease in computation power requirements and ease of multimodal integration for cognitive representations.

**Index Terms**—Physical agents, Attention, cognitive modeling, mobile robotics.

## I. INTRODUCTION

**D**ESIGNING an autonomous control system for a mobile robot implies a decision on what inputs will be handled and what repertory of actions can be executed at any given time. The option of considering all the available sensory information as input for the core control of the robot is usually both unnecessary and extremely expensive in computational terms. Analogously, not all possible robot behaviors are appropriate at any given time. Instead of considering all physically plausible behaviors, the robot control system should take into account its current situation and assigned mission in order to build a shorter list of eligible behaviors. A simplistic definition of artificial attention can be drawn from the problem described above. Hence, let us say that an efficient artificial mechanism for attention would solve the problem of filtering relevant sensory information and selecting relevant behaviors.

According to the former definition, we need to specify what relevant means in terms of implementing an efficient attention mechanism. Relevant sensor data and relevant behaviors are those that could be both useful to accomplish the mission and adapted to the world in which the robot is situated. Attention has been typically applied to artificial vision systems taking the human visual attention mechanisms and its related eye movement control (foveation) as inspiration [1]. Visual attention has been extensively applied in robotics, e.g. [2]. However, much less effort has been put in pure multimodal attention mechanisms [3]. Usually attention mechanisms for robots focus



Fig. 1. Mobilerobots Pioneer 3 DX robot.

in great degree on visual sensory information; nevertheless, some salient examples incorporate data from other sensors in the attention mechanism. For instance, laser range finders [4]. In this work we present a purely multimodal attention mechanism, which permits a straightforward and graceful inclusion of new additional sensors of different modalities. The proposed mechanism for multimodal integration is not only intended to exclusively serve agent's attention capability, but also to provide a rich, complex, and coherent percept representation that can be directly used by other cognitive functions like associative learning and decision making.

Currently, sonar, contact, and vision modalities have been already incorporated in the perception subsystem. The actuators subsystem consists exclusively on a two-motor set forming a single differential wheel drive that provides the required indoor mobility. The testing platform is based on a Mobilerobots Pioneer 3 DX robot (P3DX) equipped with an onboard laptop computer, frontal centered fixed single camera, eight-transducer frontal sonar ring, and frontal and rear bumper rings (see Fig. 1). A counterpart recognition and chasing task has been selected as preliminary testing domain for the proposed cognitive attention mechanism. Both simulated and real environments have been setup as described below. In the simplest scenario, two P3DX robots are used: *P3DX-Chaser* is the robot running the autonomous control architecture which implements the proposed attention mechanism, and *P3DX-Target* is a similar robot base tethered or remotely controlled by a human. The mission consigned to *P3DX-Chaser* is to keep heading towards *P3DX-Target* maintaining a safe constant distance between the two robots. In order to accomplish the chasing goal, *P3DX-Chaser* has to pay attention to complex percepts such as “a moving target which is a Pioneer 3 DX robot”, while ignoring other percepts which are irrelevant to current mission. Being able to deal with such complex percepts when focusing attention is one of the main goals of this work in cognitive artificial attention.

Raúl Arrabales is with University Carlos III of Madrid.

E-mail: rarrabal@inf.uc3m.es

Agapito Ledezma is with University Carlos III of Madrid.

E-mail: ledezma@inf.uc3m.es

Araceli Sanchis is with University Carlos III of Madrid.

E-mail: masm@inf.uc3m.es

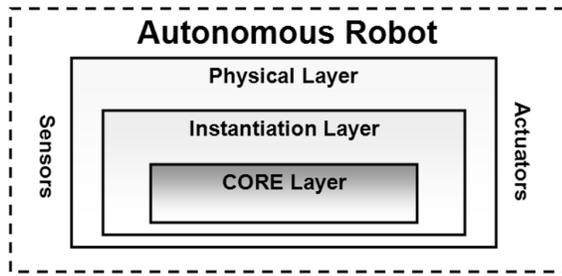


Fig. 2. CERA Control Architecture Layers.

In the next sections we discuss the implementation of an attention mechanism able to fulfill the requirement of selecting relevant sensorimotor information. Section II provides an introduction to the software architecture and how the attention mechanism is incorporated into a layered control system. Section III covers the definition of the attentional contexts that are used to form sets of sensory and motor data. Section IV is dedicated to explain how the proposed mechanism allows the integration of different modality sensory information into the same context. Section V illustrates the application of the proposed technique to the domain of counterpart recognition and chasing. Finally, we conclude in section VI with a discussion of the benefits and possible areas of application of the attention mechanism in the field of cognitive robotics.

## II. ARCHITECTURE OVERVIEW

Typically, autonomous robot control architectures are structured in layers. Each layer usually represents a different level of control, from lower reactive levels to higher deliberative levels. The proposed attention mechanism has been integrated into a three level control architecture called CERA (Conscious and Emotional Reasoning Architecture). CERA is composed of a lower level, called Physical Layer, a mission specific level, called Instantiation Layer, and a higher level, called Core Layer, where higher cognitive functions are implemented (see Fig. 2). The details about CERA are discussed elsewhere [5].

A number of processing steps that take place within this architecture can be identified as specifically belonging to the attention mechanism. Concretely, if we look at the perception cycle, the following steps are performed (see Fig. 3):

- Sensory data is acquired by sensors (for instance, a bump panel contact is reported).
- Contextualization parameters (like relative position vectors and timestamps) are calculated for each perceived object or event.
- Sensor Preprocessors build single percepts using both sensory data and their associated contextualization parameters.
- Groups of single percepts showing contextual affinity are eventually combined into complex multimodal percepts.

CERA Physical Layer provides the required functionality in order to interface with the robot hardware. In other words, it provides access to sensors and actuators. Additionally, as the CERA architecture has been designed to host the proposed attention mechanism, the physical layer is also in charge of

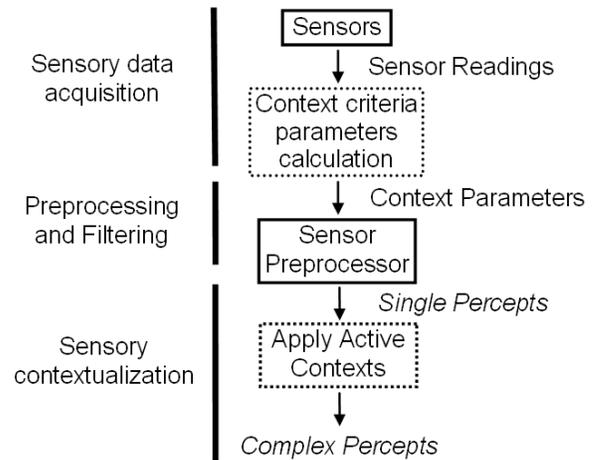


Fig. 3. Perception cycle overview.

calculating the contextual parameters of percepts and actions. From the point of view of the attention mechanism, the CERA Physical Layer is the domain of single percepts and simple actions. As the Physical Layer is specific to a given hardware it has to be changed or adapted if the underlying physical robot is replaced by a significantly different model. The composition of percepts and actions forming complex percepts and complex actions takes place in the CERA Instantiation Layer. This is the place where mission-specific contexts are to be applied, and therefore mission-specific complex percepts and behaviors are generated. As the Instantiation Layer is designed specifically for a given problem domain it can be replaced by a different problem instantiation without changing the existing Physical and Core layers. Finally, the CERA Core Layer is where a machine consciousness model is implemented based on several modules that represent higher cognitive functions. One of these functions related to consciousness is attention.

The attention module implemented in the Core Layer is designed to activate the most appropriate contexts at any given time, i.e. an attentional bias is induced from the Core Layer enforcing particular contexts. Complex percepts that are obtained in the perception cycle depend on the active contexts established by the Core Layer. Therefore, at any given time, the robot can only perceive those objects or events that are relevant to the mission (top-down attentional bias). However, a mechanism for bottom-up attention is always in place, so critical single percepts like bumper contact notifications are not ignored. One of the benefits of integrating the attention mechanism into a layered control system, where priorities for perceptions and actions can be established, is that the implementation of a combination of top-down and bottom-up attentional bias can be naturally enabled.

## III. DEFINITION OF ATTENTIONAL CONTEXTS

Our proposed artificial attention mechanism is inspired in the concept of context as defined in the Global Workspace Theory (GWT) [6]. The GWT is a cognitive account for consciousness, and therefore it covers attention as a key characteristic of conscious beings. However, the GWT do not

provide any algorithmic description of attention but just a metaphorical explanation. A theater spotlight simile is used to represent the focus of consciousness. This spotlight illuminates only a small part of the scene, which is considered the conscious content of the mind. The scene is actually built upon the subject's working memory. The movement of the spotlight, i.e. the selection of contents that will be used for volition and action, is directed by unconscious contextual systems. The aim of the work described in this paper is to design and test an implementation of such contextual systems, which are able to adaptively direct attention toward the interesting areas of the robot sensorimotor space.

From the point of view of perception, contexts are sets of percepts retrieved from the sensors. Percepts are considered the minimal information units obtained by the robot sensory machinery [5]. Therefore, a sensory context can be used to build a complex percept composed of related single percepts. From the point of view of behavior, contexts define sets of actions available for execution. Hence, we can define behavioral contexts as possible compositions of related actions. In order to generate an efficient robot behavior, both sensory contexts and behavioral context have to be adaptively generated.

#### A. Visual Field Segmentation

The first stages in visual sensor data processing are concerned with attentional context definition. Concretely, instead of applying a full preprocessing task to the entire image captured by the camera sensor, each incoming frame is fragmented into smaller regions. Subsequently, only one selected fragment (foveal region) is further processed, thus reducing to a great extent the processor requirements of visual sensor preprocessor. Additionally, as explained below, this strategy allows the robot to focus attention in specific visual regions also in further processing stages. Nevertheless, before the preprocessing stage, when context criteria are evaluated, all visual data packages (frame segments) are equally processed.

It is known that this strategy is similar to the way human visual system processes the foveal region, which is much richer in resolution and detail than retinal periphery. Humans use the fovea to fixate on an object and specifically process its image while maintaining a much less demanding process for peripheral regions [7]. This very same strategy has also been successfully applied in other artificial systems, e.g. [8].

#### B. Context Criteria

We have designed the process of context formation as the application of predefined criteria in order to calculate the degree of relation between the potential elements of a given context. Basically, a context should be constructed in a way that it can become a meaningful representation of the reality, i.e. the interplay between agent and situation must be enforced by a proper definition of both sensory and behavioral contexts. The very basic factors that need to be considered in the correct representation of robot situation in the world are time and location. Nevertheless, other factors can be considered depending on the problem domain and internal state representation richness. In the work described here, color

and movement properties have been considered as additional criteria; therefore, four criteria have been used for context formation in the experiments described below.

The time criterion refers to the exact moment at which a stimulus is perceived. Therefore, it should be taken as an important criterion to relate one percept to another. Given that different sensors and their associated device drivers can take different time intervals to process the sensory information, a mechanism for time alignment is required. It has been demonstrated that such a time alignment mechanism is present in biological brains [9][10]. Although visual and auditory stimuli are processed at different speeds, the time gap between different processed signals, whose physical originators were acquired at the same time, is automatically removed by the brain [11]. An analogous artificial mechanism has been implemented in the proposed architecture.

Location is another fundamental criterion for context formation as the representation of the position of objects in the world is a requirement for situatedness. Furthermore, the location of an object relative to the robot body (or any other reference frame) is required for generating adaptive behaviors. The relative location of any element in the sensory world is necessary for the integration of complex percepts; additionally, it allows the selection of a given direction of attention toward the most relevant places. The presence of space coding neurons and the use of reference frames (like somatotopic or head-centered) has been demonstrated in the mammal brain [12][13].

In a world where color patterns can be associated with particular objects, this property of entities should be taken into account. Similarly, some objects are mobile while others remain static; consequently, movement is a property that should also be considered as criterion for relevant context formation. Particularly, the task of counterpart recognition has been simplified in the research under discussion by characterizing other peer robots as autonomously moving red and black objects. The presence of specialized areas for color and movement detection has been demonstrated in human's brain visual cortex [14].

Following the principles presented above, we have used time, location, color, and motion as fundamental contextualization criteria for the formation of:

- Sensory contexts as composition of single percepts (complex percepts), and
- behavioral contexts as composition of simple actions.

In order to generate these contexts, both single percepts (which are built from data packages obtained from sensors) and simple actions (which are defined as part of the robot control system) are required to incorporate estimated time, location, motion, and color parameters (see Fig. 4). Motion properties could be obviously derived from time and location parameters; however, we have decided to use a natively visual motion detection approach in which motion properties are directly obtained from visual input analysis. In our proposed architecture there are specialized modules designed to calculate time, color, motion, and location parameters: the Timer module maintains a precision clock (less than 1 millisecond resolution) that represents the robot's age, the Proprioception

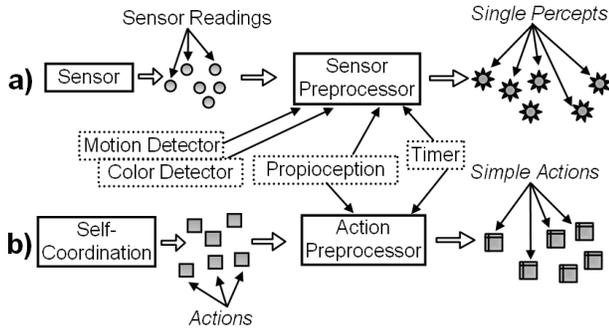


Fig. 4. Creation of single percepts and simple actions

module maintains all the required information to calculate the exteroceptive sensors position. This information is necessary to estimate the relative location of an object or event detected by an exteroceptive sensor. A Color Detection module is in charge of providing a color histogram representation associated to visual data retrieved by the camera. Similarly, a Motion Detection module continuously calculates the differences between the last data retrieved by the camera and current visual data.

Time, location, color, and motion parameters provided by the Timer, Proprioception, and Color and Motion Detector modules are used by the preprocessor modules in charge of generating single percepts and simple actions. A Sensor Preprocessor takes a given sensor reading as input, then calculates the relative position of the source of the reading and the instant when it took place using the information provided by the Timer and Proprioception. In case of visual sensor readings, also color and motion detectors are activated and histogram and motion vectors are calculated. Finally, the sensor preprocessor creates a single percept packing together the proper sensor reading with its contextualization information. The Action Preprocessor takes as input an action generated by the Self-Coordination module (this module and the way it works is described elsewhere [5]), and applies the same approach as in the Sensor Preprocessor in order to build the Simple Action representations.

More parameters should be added to single percepts if other contextualization criteria are to be applied. In the work described in the present paper, the following parameters have been used:

- **Timestamps:** two different timestamps are recorded in single percepts. The first timestamp is set when the sensory data is collected from the sensor. Usually this timestamp is directly assigned by the sensor hardware and retrieved in the control system through the sensor driver. The second timestamp is set when the percept is actually used in the control system. The time span between these two timestamps can be significant when a sensor is incessantly notifying readings and there is not enough onboard processing power to dispatch all the incoming data. Actually, the time span value can be used to discard too old sensory data which is not significant to the current robot state. Similarly, two timestamps are logged in the case of simple action. The first one is

set when the simple action is created and enqueued in the control system. The second timestamp is set when the action enters the core execution cycle, i.e. when the action is actually dequeued and dispatched (begins physical execution). The time span between these two timestamps can be used to detect delays in the execution queue and eventually abort too old actions.

- **J-Index:** for the representation of the location parameter of both single percepts and simple actions we have decided to use the robot body center of mass as reference frame. The term J-Index refers to a structure able to represent or map the relative position of an object or event within a biological brain [15]. We have adapted and enhanced the original definition of the J-Index representation with the aim of representing both the relative position and relative dimensions of the object. Hence, our J-Indexes are implemented as a composition of several n-dimensional vectors. The main vector is called the j referent vector, and is used to calculate the relative position of the geometrical center of the percept's source or the geometrical target of an action. Depending on the nature of the sensor that is reporting the sensory data, more positional vectors can be calculated in order to estimate the size of the percept (examples for sonar range finder, camera, and bump panel arrays are described below).
- **Color Histogram:** Each data package provided by the visual sensor (corresponding to a frame segment) is assigned a color histogram, where the frequency of image color components is represented. Obviously this parameter can only be set for visual sensory information. Any other more demanding visual processing concerned with color, like texture recognition is not defined as contextual parameter because it will be limited to the scope of foveal region (when the robot is fixating on a particular object); therefore, it must be part of the sensor preprocessing task.
- **M-Index:** The result of the application of the motion detection module on an incoming visual data package is a movement vector called M-Index, whose value is zero when no movement has been detected. Although motion could be detected using other sensory modalities, like sonar, we have decided to use only vision for the time being. Nevertheless, when complex percepts are built, the robot own movement is taken into account to calculate the relative motion of observed objects.

The timestamp parameters are easily acquired using the robot's control system precision timer. However, the J-Index parameters require more elaboration, particularly in the case of movable sensors. In the case discussed here, we have used P3DX robots (see Fig. 5a) with fixed position sensors: a frontal sonar array (see Fig. 5c) and frontal and rear bump panels (see Fig. 5b). In the experiments that we have carried out so far, J-Indexes have been calculated for sonar readings, bump panels contact and release notifications, and visual segments. The J-Indexes are calculated as a function of the transducer (fixed) position and orientation (relative to the robot front).

Although the J-Index parameter can be primarily repre-

sented by a three-dimensional vector, for the task of following a counterpart robot in a flat surface, a two-dimensional  $j$  referent vector can be considered, where  $(X,Z) = (0,0)$  represents the subjective reference frame of the robot (see Fig. 5b and 5c). Nevertheless, a  $Y$  coordinate (height) is usually calculated even though it is not used.

The calculation of the  $j$  referent vector is different depending on the sensor. In the case of bump panels, as they are located at angles around the robot (see Fig. 5b), the  $j$  referent vector is calculated using (1). Where,  $BR$  is the bump panel radius, i.e. the distance from the center of mass of the robot to the bumper contact surface (see Fig. 5b).  $BA$  is the bump panel angle to the front of the robot (Pioneer 3 DX bump panels are located at angles  $-52^\circ$ ,  $-19^\circ$ ,  $0^\circ$ ,  $19^\circ$ , and  $52^\circ$ ).  $BH$  is the height at which the bumpers are mounted.

$$j = (X, Y, Z) = \begin{pmatrix} BR * \cos(BA) \\ BH \\ BR * \sin(BA) \end{pmatrix} \quad (1)$$

Additionally, two more vectors are calculated to be associated to a bumper percept: the left- $j$  referent and the right- $j$  referent (see Fig. 6). These two vectors represent the dimensions of the percept (the width assigned to the collision).

In order to calculate the  $j$  referent vector corresponding to a given sonar reading, (2) is used. Note that the calculation of  $j$  referent vectors is dependent on the type of sensor being considered.

$$j = (X, Y, Z) = \begin{pmatrix} (R + SR) * \cos(SA) \\ SH \\ (R + SR) * \sin(SA) \end{pmatrix} \quad (2)$$

Where,  $R$  is the maximum range measured by the sonar transducer,  $SR$  is the distance from the center of mass of the robot to the sonar transducer, and  $SA$  is the angle at which the particular sonar transducer is located. Note that sonar transducers are located at angles  $-90^\circ$ ,  $-50^\circ$ ,  $-30^\circ$ ,  $-10^\circ$ ,  $10^\circ$ ,  $30^\circ$ ,  $50^\circ$ , and  $90^\circ$  to the front of the robot (see Fig. 5c). Therefore, each transducer is able to measure the free space available within a three-dimensional  $15^\circ$  wide cone (this cone aperture corresponds to the SensComp 600 transducer).

Taking into account that the ultrasonic beams emitted by the sonar transducers take the form of a symmetric three-dimensional cone, at least one additional  $j$  referent vector has to be calculated in order to estimate the dimensions of the single transducer sonar percept, i.e. the open space perceived in front of that particular sonar transducer. The main  $j$  referent vector calculated using (2) represents the cone bisector. Additionally, two more vectors: the left- $j$  referent vector and right- $j$  referent vector represent the lateral 2D boundaries of the percept (see Fig. 7). The representations of J-Indexes for both sonar and bumpers have been designed as described above with the aim of implementing an attention algorithm. Although some of the calculated reference vectors are expendable, they are useful to pre-calculate the regions of the world affected by a given percept. Besides, this representation is also particularly useful for the subsequent task of counterpart robot chasing.

In the case of visual sensory information, each segment is assigned a  $j$  referent vector which corresponds to the relative

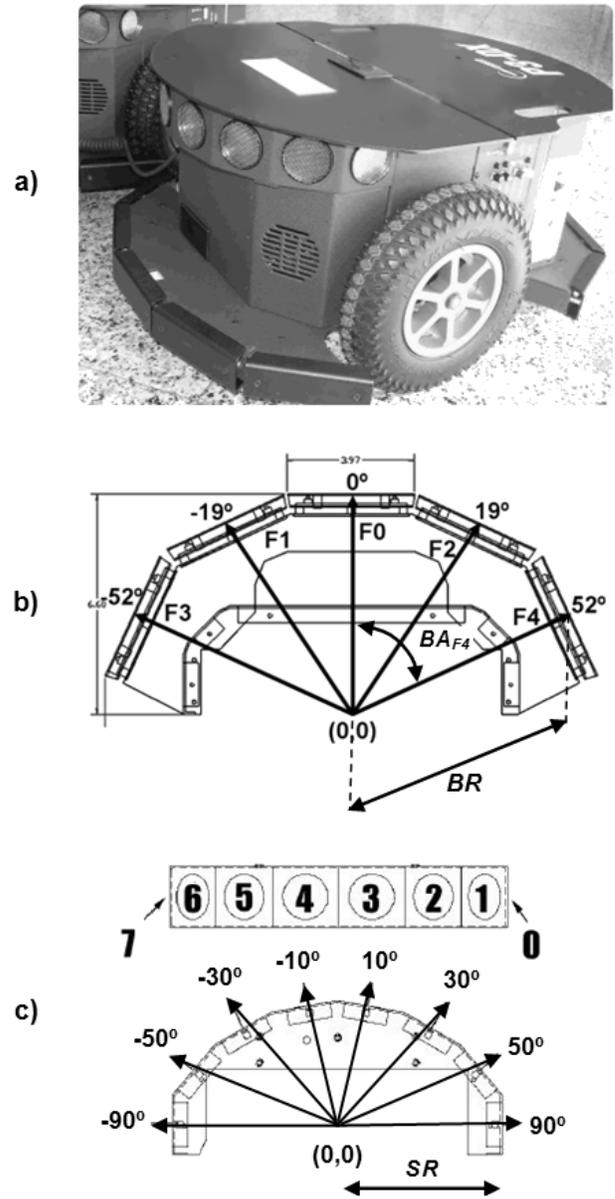


Fig. 5. MobileRobots Pioneer 3 DX Robot, frontal bumper panel, and frontal sonar ring.

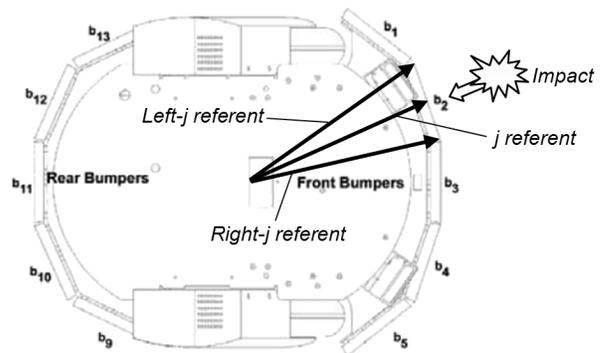


Fig. 6. Vectors calculated to build the J-Index of a single bump panel contact percept.

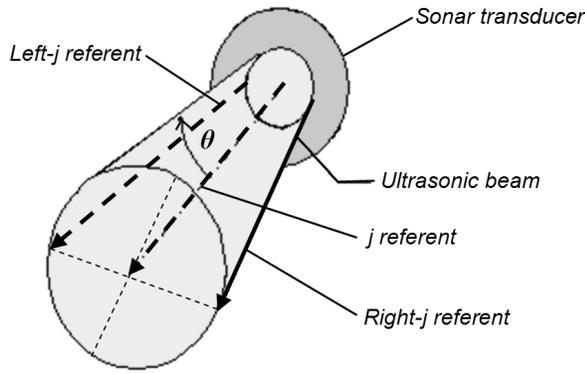


Fig. 7. Vectors calculated to build the J-Index of a single sonar transducer percept.

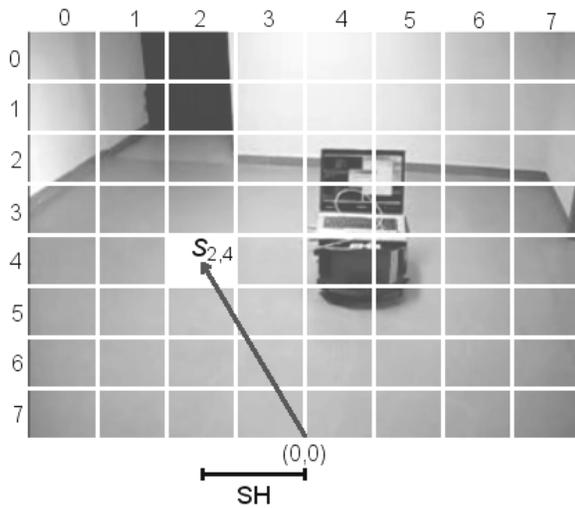


Fig. 8. J referent vector of a segment of visual sensory data.

location of the geometrical center of that particular segment within the visual field. As the orientation of the camera is fixed, it is straightforward to estimate the relative  $X$  coordinate (left / right position relative to the robot) of the corresponding percept, being  $SH$  the distance from the optical vertical axis of the camera (center of the field of view) to the center of the segment. Fig. 8 depicts an example of segmented visual input in which the visual field has been divided into 64 smaller regions and the referent vector for segment  $S_{24}$  is calculated. Estimating the distance to the visual percept is a different matter. Usually a stereo vision system is used. Having just one camera, a pinhole model could be applied. Nonetheless, in this work distance to objects is provided exclusively by sonar percepts.

Fig. 9 shows an example where  $j$  referent vectors are calculated only for those segments in which any saliency has been detected. In this case, as the goal is to follow a red counterpart robot, two segments where the color histogram presents a salient frequency of red have been selected. At this point, when the sensor preprocessor is building single percepts from visual input, a foveal region is to be selected, and the rest of the image is discarded and not taking part

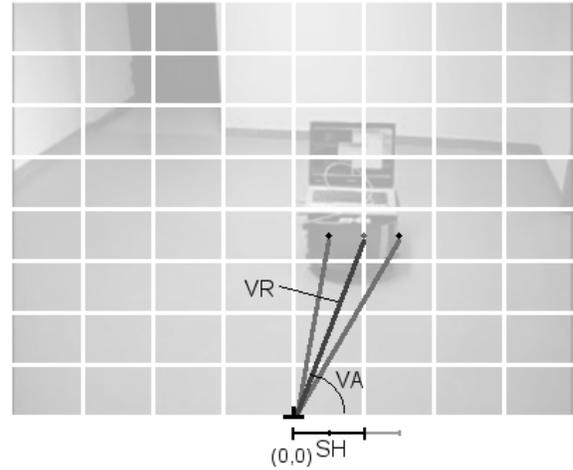


Fig. 9. J-Index of a single visual percept.

in any further processing. This means that no single percepts are built with visual information outside the simulated fovea. The foveal region has to be kept small, therefore, a maximum of four contiguous segments are considered to form a single visual percept. The J-Index of a single percept that has been formed as a combination of contiguous segments is calculated by adding a new  $j$  referent vector pointing to the geometrical center of the set of segments (see Fig. 9). Concretely, (3) is used to calculate the main  $j$  referent vector of the visual single percept, where  $CH$  is the relative height at which the camera is located,  $VR$  is the distance from the optical axis origin to the center of the percept, and  $VA$  is the angle relative to the optical horizontal axis ( $SH = VR * \sin(VA)$ ).

$$j = (X, Y, Z) = \begin{pmatrix} VR * \sin(VA) \\ CH \\ ? \end{pmatrix} \quad (3)$$

Note that the calculation of all context criteria parameters is rather quick, and no complex processing is carried out at this level of the architecture. One of the advantages of having an attention mechanism is the processing power saving, and this principle is preserved by keeping simple context criteria parameters. When more processing is required in order to build complex percepts and apply inference rules, this is uniquely done using a reduced subset of the sensory space, which has been already selected by the application of a given context.

### C. Actions Context Composition

As both Single Percepts and Simple Actions include the basic time and location contextualization parameters (timestamps and J-Indexes) it is straightforward to calculate similarity distances between them. Other specific sensory parameters, like color, are used exclusively with single percepts. Therefore, contexts can be generally defined based on the dimensions of relative time and relative location, and also specifically defined for some sensory modalities using other specific parameters. Each sensory context is used to build a representation structure called complex percept (see Fig. 10a). Complex percepts enclose the required information to represent the meaning of the

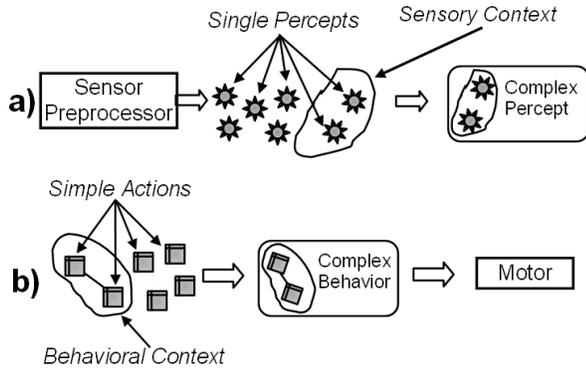


Fig. 10. Formation of complex percepts and complex behaviors.

associated sensory context as required by the subsystems of the autonomous control system. As behavioral contexts are formed they may trigger the generation of the corresponding complex behaviors, which are representations that enclose sequences of actions specified by the behavioral context (see Fig. 10b). In the present work, the behavioral context formation has been oversimplified in order to generate uncomplicated behaviors for the available actuator: the P3DX differential drive. Two basic operations have been defined for the control of the differential drive:

- 1) **RotateInPlace**: this operation takes an angle in degrees as input parameter (positive values mean counterclockwise rotation) and triggers the robot rotation in position until it completes the consigned angle.
- 2) **MoveStraight**: this operation takes a speed in meters per second as input parameter (positive values mean move forward) and triggers the robot movement towards the current heading (or backwards for negative speed values).

Attending to the relative direction specified by the attention mechanism (a composition of J-Indexes representations), an angle parameter is calculated for the RotateInPlace operation in order to set the robot heading towards the object that “called robot’s attention”. Also, a speed parameter is calculated as a function of the distance to the object. This means that the typical minimum behavioral context is formed by a sequence of simple actions like a RotateInPlace operation followed by a MoveStraight operation.

#### IV. MULTIMODAL INTEGRATION

Combining multiple monomodal sensory data sources is a typical problem in mobile robotics, also known as multisensory integration or sensor data fusion [16]. Actually, in the present work we are also approaching the problem of fusing proprioceptive and exteroceptive sensor data. Neuroscientists refer to the binding problem [17], as the analogous problem of how to form a unified perception out of the activity of specialized sets of neurons dealing with particular aspects of perception. From the perspective of autonomous robot control we argue that the binding problem can be functionally resolved by applying the proposed contextualization mechanism.

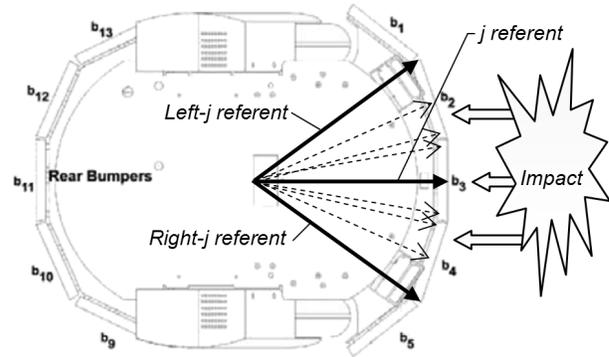


Fig. 11. Vectors calculated to build the J-Index of a complex bumper contact percept.

#### A. Monomodal Context Formation

Taking the bump panel percepts as example, we can illustrate how a sensory context gives place to a monomodal complex percept. Using the aforementioned common criteria, time and location, if the bumper handler of our robot reports contact in bump panels b2, b3, and b4 simultaneously (see Fig. 11), a context is automatically created if these three independent notifications have close enough timestamps. Therefore, the three single percepts are associated by a temporal context. Additionally, as b2, b3, and b4 are located side by side, the corresponding contact percepts J-Indexes will indicate proximity, thus forming an additional spatial context that again associates these three single percepts. The newly created complex percept, which is a composition of three single percepts, also holds a representation of a J-Index. This complex percept J-Index is calculated as a function of the reference vectors of the former single percepts (note that Fig. 11 depicts with solid lines the J-Index referent vectors of the formed complex percept, and dashed lines represent the referent vector of the old single percepts).

The way the J-Index of a complex percept is calculated depends on the nature (shape, dimensions, etc.) of the single percepts that take part in the context that gave place to it. The composition of J-Indexes is trivial when all the single percepts belong to the same modality (as illustrated in Fig. 11). However, the composition can be complex when several different modalities are involved.

#### B. Multimodal Context Formation

Focusing on the mentioned fundamental criteria for contextualization (time and location), all percepts, independently of their modality, can be compared with each other, thus allowing a simple mechanism to create perceptual contexts. The contexts formed following this method can have significant meaning. For instance, “all objects within the reach of the robot” (context formed applying the criterion of location and estimating that the relative location is below a given threshold, like the robotic arm reach distance in this case), or “all events that took place between five and ten minutes ago” (context formed applying the criterion of time and estimating that the relative timestamp of the events fall within the given

interval). Similarly, more specific criteria can be used in order to build more specific contexts which might not involve all the available sensory modalities. This is the case of the motion and color criteria used in this work.

### C. Contextualization Hierarchy

The proposed contextualization mechanism supports hierarchical composition; hence complex percepts can be built by either combining:

- A number of single percepts.
- A number of complex percepts.
- Both single and complex percepts.

In order to assemble coherent percepts, a priority policy has been established in relation to complex percept formation. The first and top priority contextualization step is to build complex percepts that come from the application of contextualization criteria over the same modality single percepts. The outcome of this first step is a set of monomodal complex percepts. As illustrated above, these monomodal complex percepts can come from simultaneous and contiguous bumper contacts or from simultaneous and contiguous salient visual segments. Once the first contextualization step is completed, both the newer monomodal complex percepts and existing single percepts enter the CERA Workspace where multimodal complex percepts are built (see Fig. 13 for an example).

### D. Managing contradictory percepts

A common application of multimodal sensory information fusion is the disambiguation or refutation of contradictory sensor data. In the case under study in this paper, contradictory information happens to be processed when the sonar transducers fail to detect a sharp solid corner (the ultrasonic beams are diverted, and do not come back to the transducer, failing to provide a realistic range measurement). In such a scenario, the last resort are the bumpers. When the robot base is too close to the sharp corner, bumpers will contact the obstacle and notify single percepts, which in turn will become complex percepts. However, during the process of complex percepts formation, potential contradictory information has to be handled. The time criteria for context formation will associate the roughly simultaneous readings from both sonar and bumpers. But, in the case of a bad sonar reading the single percepts available are not consistent. Therefore, a policy has to be established in order to build a significant complex percept out of conflicting single percepts. A single but effective approach is to apply a level of confidence to each sensor modality depending on the situation. In the case described here, we have just assigned more confidence to bumper contact notifications than sonar measurements.

## V. PAYING ATTENTION TO COUNTERPART ROBOTS

Following a counterpart robot across an unknown office-like environment has been selected as a preliminary problem domain for the testing of the proposed attention mechanism. It provides a valid real world scenario where the sensors and actuators described above can be used to achieve the mission

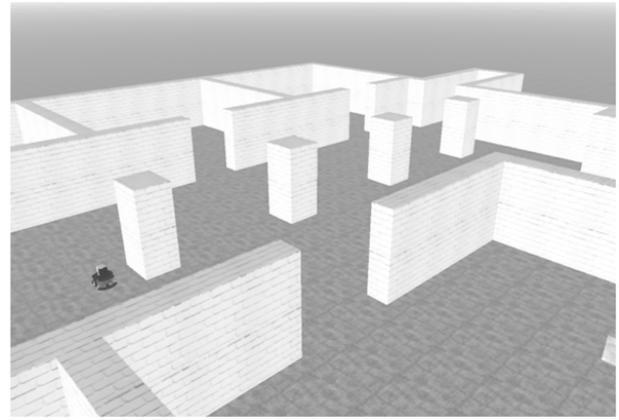


Fig. 12. Simulated indoor environment.

goal: to find the *P3DX-Target* robot and then keep a constant distance to it. Both simulated and real environment setups have been prepared. All localization estimation problems have been neglected for the time being. Figure 12 shows a screen capture of the simulated environment we have used for initial testing.

One of the objectives of the proposed attention mechanism is to offer an effective policy for selecting the next action (or complex behavior) as part of the robot's main control loop. In the case of counterpart chasing, spatial contexts are defined in order to estimate the best heading that the robot should take. A specific CERA Instantiation Layer has been coded with the aim of representing the particular complex percepts that are required for the chasing task.

Robot mission is structured in two sequential tasks. Firstly, during the searching task, *P3DX-Chaser* has to find out if the *P3DX-Target* robot is in the surroundings. Secondly, if the target has been detected (target engaged), the chaser has to follow it keeping a constant separation distance. If for some reason the target is lost, *P3DX-Chaser* will come back to the first task. During the initial searching task, the chaser wanders randomly performing surrounding inspections ( $360^\circ$  turns) periodically. Two attentional contexts are applied during the searching phase in order to detect the presence of the target: "a red and moving object". Detecting this sort of objects involves paying attention to complex percepts that are formed as a result of the conjoined application of color and motion context criteria. Therefore, such a context should be activated during the searching phase. However, as motion criteria is difficult to assess when the own referential system is also moving, CERA Core Layer initially activates only the red color context while the robot is wandering or performing a  $360^\circ$  scan. When a salient complex percept is obtained due to red color saliency (like in Fig. 9), robot comes to a full stop and activates a new attentional context with two criteria: red color and motion. Then, if more complex percepts are obtained from a given location, the target is recognized and engaged, and the second task (following) is activated. During the second task, again a single criterion context for red color is activated and the robot heading is adjusted to the direction indicated by the target complex percepts (*SH* value and sign). Basically, a *j* referent vector is calculated based on the target

complex percept J-Index, and simple actions are generated that will cause the chaser to head towards the target.

Keeping a constant distance to the target is facilitated by the ranging data obtained from sonar. As single percepts from vision and single percepts from sonar share location parameters (j referent vectors), distance to target can be estimated by multimodal contextualization. Actually, the complex percepts that represent the target are composed of both visual and sonar single percepts. These single percepts were associated in the same complex percept because of their affinity in relative location. This means that target complex percepts include sonar ranging data in addition to the visual pose estimation.

Fig. 13 shows an example of both visual and sonar data as ingested in the CERA Physical Layer, where associated time and location parameters are calculated. Then, sensor preprocessors build single percepts including timestamps and J-Indexes. All generated single percepts enter the CERA Physical workspace where the complex percepts are built based on current active contexts. Active contexts are established by the higher control system running in the CERA Core Layer. The scenario depicted in figure 13 corresponds to the chasing task; hence a context for red objects is active (in addition to location and time, which are always used as contextualization criteria). Right side of the figure shows an example of the complex percepts that are formed due to the application of the mentioned contextualization mechanism. Single percepts corresponding to visual segments  $S_{5,5}$  and  $S_{6,5}$  are selected because the present saliency in terms of the red color contextualization criterion. Given that their j referent vectors happen to be contiguous, a new monomodal (visual) complex percept is built as a composition of them. As shown in the picture, the J-Index of the newer monomodal complex percept points to the geometrical center of the visual segment formed as a combination of the two former single percept segments. It can be noticed that the J-Index of this visual complex percept does not spot the actual center of the target, but the approximation is good enough for the realtime chasing task. Once monomodal complex percepts have been built, time and location contextualization is applied amongst different modalities.

Right bottom representation in the picture (Fig. 13) corresponds to sonar j referent vectors, including a highlighted single percept (the one obtained from the reading of the sonar transducer oriented at  $+10^\circ$ ). The projection outlined top-down from the visual complex percept to this sonar single percept indicates that both percepts are to be associated and will form a multimodal complex percept. Time association is obvious; however, the location contextualization between visual and sonar percepts require some additional parametric alignment as these different modality sensors present particular orientations and wide span. Furthermore, as explained above, only the X coordinate is considered for visual percepts (visual horizontal axis). While we have used a  $+90^\circ$  field of view camera, the Pioneer 3DX robot frontal sonar ring covers a total field of  $+195^\circ$  (including blind angles between transducer cones). Therefore, only percepts originated from the central  $90^\circ$  of sonar coverage are taken into account for visual to sonar contextualization. Black dashed lines on the right hand side of

the figure represent the alignment between visual horizontal axis and the central  $-45^\circ$  to  $+45^\circ$  angular span of frontal sonar. In this case, the value of  $SH$  in the visual complex percept corresponds to the sonar percept originated from the sonar transducer at  $+10^\circ$ . The measurement represented in this particular sonar percept (2493 millimeters) is directly the distance estimate assigned to the multimodal complex percept as visual percept itself does not provide any distance estimate.

Preliminary results obtained applying the proposed attention mechanism to the human-controlled target chasing task are promising; however more complex environments have to be tested in order to appreciate the real potential of the cognitive approach. In addition to the manual control of *P3DX-Target*, which produces very variable results, three autonomous simple behaviors have been implemented with the aim to test the capability of the attention mechanism when confronted to different movement patterns (scenarios a, b and c depicted in Fig. 14). Fig. 14 shows the typical trajectories of the autonomous control schemes implemented in *P3DX-Target*. Initial time to target engaged state varies and is basically dependent on start position of both *P3DX-Chaser* and *P3DX-Target* robots. Therefore, in the present case, the performance of the attention mechanism is measured in terms of the overall duration of target engaged state (percentage of total navigation time when the target is engaged). The performance when chasing targets in open space (wide corridors) is 100% in scenario (a), and close to 100% in scenarios (b) and (c). However, when the target (a, b, or c) performs obstacle avoidance maneuvers close to walls, performance usually fall to 50-70%. In these situations the chaser also has to avoid obstacles, eventually causing the loss of target.

## VI. CONCLUSION AND FUTURE WORK

A novel attention mechanism for autonomous robots has been proposed and preliminary testing has been done in the domain of simple mobile object recognition and chasing. The integration of the attention cognitive function into a layered control architecture has been demonstrated. Additionally, the problem of multimodal sensory information fusion has been addressed in the proposed approach using a generic context formation mechanism. Preliminary results obtained with the simulator show that this account is applicable to classical mobile robotics problems. Nevertheless, moving to a real world environment and facing more demanding missions including localization requirements would imply dealing with the problem of imperfect odometry [18]. In such a scenario our proposed attention mechanism had to be integrated into a SLAM (Simultaneous Localization and Mapping) system.

The attention mechanism proposed in this work is designed to be highly dynamic and configurable. Following the same principles described above, more contexts can be created as more contextualization criteria are defined in the system. The concrete definition of criteria and context is to be selected based on the specific problem domain.

The system described in this paper is work in progress. Counterpart recognition is currently based on color and movement detection, however we are working on adding other

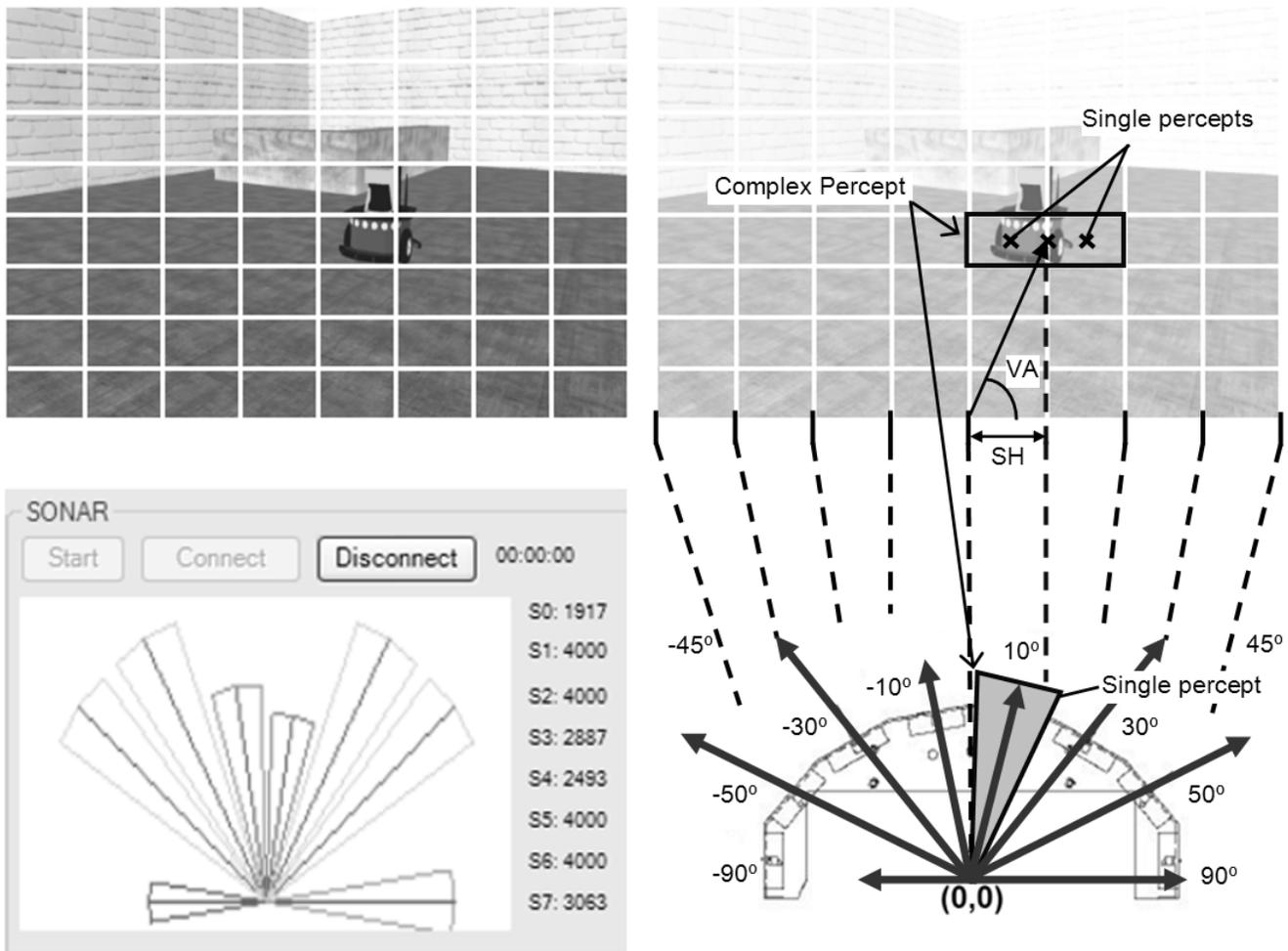


Fig. 13. Upper left image is a representation of the segmented visual sensory information acquired by the simulated onboard camera. Lower left graph is a capture from the CERA graphical user interface displaying real time sonar transducer measurements. Sonar ranging representation capture corresponds to the particular instant when the camera acquired the image depicted above. Right hand side of the picture shows a representation of the multimodal complex percept being built with this sensory information

visual and higher cognitive properties recognition in order to build a most robust mechanism. Concretely, visual texture recognition, vertical symmetry detection, and movement pattern identification are expected to greatly improve robustness in real world environments. Furthermore, a mechanism for the detection of structurally coherent visual information could be implemented as part of the proposed attention mechanism. Complex percepts formed representing unique objects could be evaluated in terms of their structural coherence, as human brain seems to do [19]. More complex attentional contexts (and therefore more contextual criteria) have to be defined in order to face other problem domains. Perception is well covered for sonar range finder and bumpers. However, additional development is required in the CERA Physical Layer in order to add more functionality to visual perception, e.g. visual distance estimation. The definition of behavioral contexts and complex behaviors should also be enhanced to cope with more complex actuators and to generate more efficient behaviors. At the level of the CERA Core Layer, learning mechanisms could be applied in order to improve the attention selection technique.

Moreover, the attention mechanism is to be integrated with other Core Layer modules, like memory and self-coordination modules in order to use the required related information for the activation of appropriate contexts in the Instantiation and Physical layers.

Given the need to define complex spatiotemporal relations in the process of attentional contexts formation, the application of fuzzy temporal rules will be considered as they have been proved to be an effective method in landmark detection (like doors) for mobile robots [20].

#### ACKNOWLEDGMENT

This research work has been supported by the Spanish Ministry of Education and Science CICYT under grant TRA2007-67374-C02-02.

#### REFERENCES

- [1] P. J. Burt, "Attention mechanisms for vision in a dynamic world," pp. 977-987 vol.2, 1988.

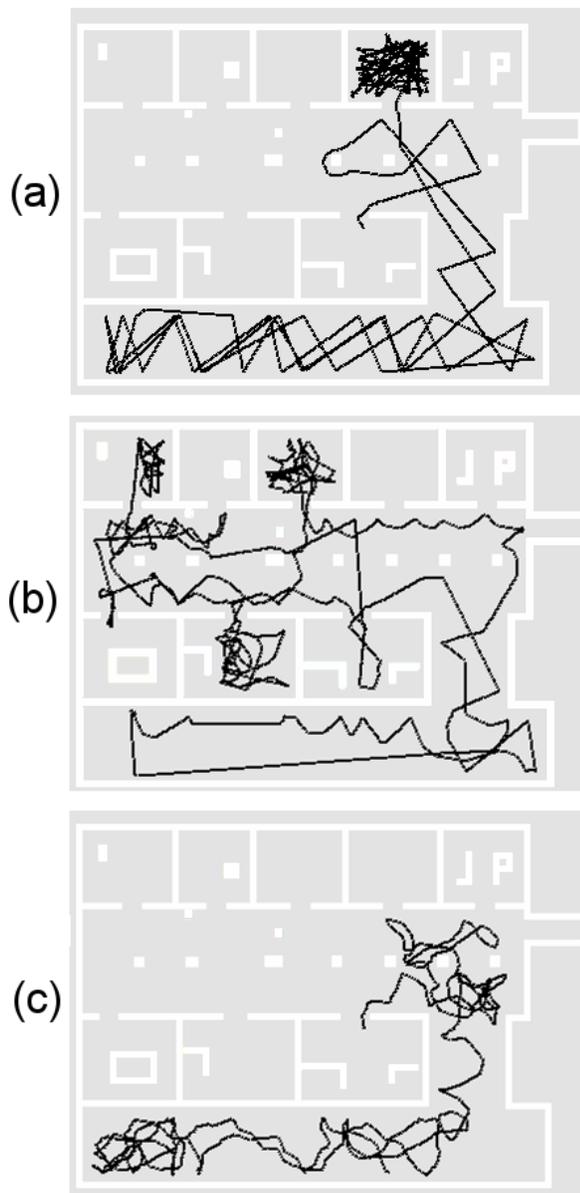


Fig. 14. Behavior (a) is the outcome of the simplest control algorithm which is based on performing random turns when the target is too close to an obstacle. Behavior (b) is obtained by adding an attentional bias to unvisited areas. Finally, behavior (c) adds random turns to the former control strategies. These three simple autonomous control strategies have been used to implement a moving target in the Robotics Developer Studio simulation environment.

- [2] R. Arrabales and A. Sanchis, "Applying machine consciousness models in autonomous situated agents," *Pattern Recognition Letters*, vol. 29, no. 8, pp. 1033–1038, 6/1 2008.
- [3] J. L. C. Mariño, "Una arquitectura de atención distribuida para agentes con sensorización multimodal", Doctoral Thesis. Universidade de Coruña, 2007.
- [4] S. Frintrop, A. Nchter, and H. Surmann, "Visual attention for object recognition in spatial 3d data," in *2nd International Workshop on Attention and Performance in Computational Vision*, ser. Lecture Notes in Computer Science, vol. 3368. Springer, 2004, pp. 168–182.
- [5] R. Arrabales, A. Ledezma, and A. Sanchis, "Modeling consciousness for autonomous robot exploration," in *IWINAC 2007*, ser. Lecture Notes in Computer Science, vol. 4527–4528, 2007.
- [6] B. J. Baars, *A Cognitive Theory of Consciousness*. New York: Cambridge University Press, 1993.
- [7] B. A. Wandell, *Foundations of Vision*. Sinauer Associates, 1995.
- [8] S. Gould, J. Arfvidsson, A. Kaehler, B. Sapp, M. Messner, G. Bradski, P. Baunstarck, S. Chung, and A. Y. Ng, "Peripheral-foveal vision for real-time object recognition and tracking in video", in *Proceeding of the International Joint Conference on Artificial Intelligence*, 2007, pp. 2115–2121.
- [9] M. H. Giard and F. Peronnet, "Auditory-visual integration during multimodal object recognition in humans: A behavioral and electrophysiological study", *The Journal of Cognitive Neuroscience*, vol. 11, no. 5, pp. 473–490, September 1 1999.
- [10] D. Senkowski, D. Talsma, M. Grigutsch, C. S. Herrmann, and M. G. Woldorff, "Good times for multisensory integration: Effects of the precision of temporal synchrony as revealed by gamma-band oscillations", *Neuropsychologia*, vol. 45, no. 3, pp. 561–571, 2007.
- [11] C. Spence and S. Squire, "Multisensory integration: Maintaining the perception of synchrony", *Current Biology*, vol. 13, no. 13, pp. R519–R521, 7/1 2003.
- [12] L. Fogassi, V. Gallese, G. Pellegrino, L. Fadiga, M. Gentilucci, G. Luppino, M. Matelli, A. Pedotti, and G. Rizzolatti, "Space coding by premotor cortex", *Experimental Brain Research*, vol. 89, no. 3, pp. 686–690, 06/01 1992.
- [13] M. Avillac, S. Deneve, E. Olivier, A. Pouget, and J.-R. Duhamel, "Reference frames for representing visual and tactile locations in parietal cortex", *Nature neuroscience*, vol. 8, no. 7, pp. 941–949, 2005.
- [14] S. Zeki, J. Watson, C. Lueck, K. Friston, C. Kennard, and R. Frackowiak, "A direct demonstration of functional specialization in human visual cortex", *Journal of Neuroscience*, vol. 11, no. 3, pp. 641–649, March 1 1991.
- [15] I. Aleksander and B. Dunmall, "Axioms and tests for the presence of minimal consciousness in agents", *Journal of Consciousness Studies*, vol. 10, no. 4–5, 2003.
- [16] R. C. Luo, "Multisensor integration and fusion in intelligent systems", pp. 901–931, 1989.
- [17] A. Revonsuo and J. Newman, "Binding and consciousness", *Consciousness and Cognition*, vol. 8, no. 2, pp. 123–127, 6 1999.
- [18] S. Thrun, "Probabilistic algorithms in robotics", *AI Magazine*, vol. 21, no. 4, pp. 93–109, 2000.
- [19] D. L. Schacter, E. Reiman, A. Uecker, M. R. Roister, L. S. Yun, and L. A. Cooper, "Brain regions associated with retrieval of structurally coherent visual information," *Nature*, vol. 376, pp. 587–590, aug 1995.
- [20] P. Carinena, C. V. Regueiro, A. Otero, A. J. Bugarin, and S. Barro, "Landmark detection in mobile robotics using fuzzy temporal rules", pp. 423–435, 2004.