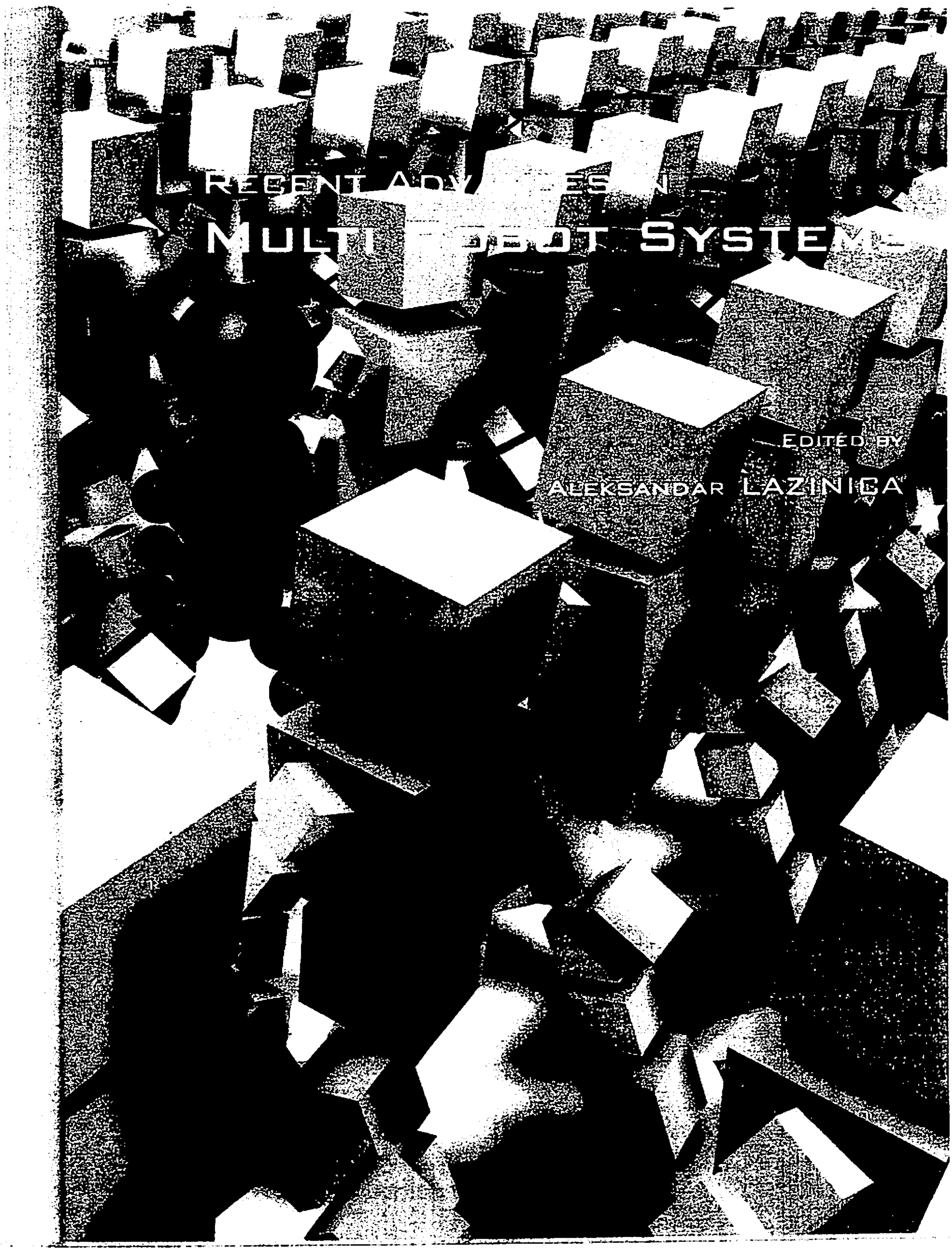


Recent Advances in Multi Robot Systems

Edited by

Aleksandar Lazinica

I-TECH

Published by I-Tech Education and Publishing

I-Tech Education and Publishing
Vienna
Austria

www.i-techonline.com
Additional copies can be obtained from:
publication@ars-journal.com

First published May 2008
Printed in Croatia

A catalogue record for this book is available from the Austrian Library.
Multi Robot Systems, Recent Advances, Edited by Aleksandar Lazinica
p. cm.
ISBN 978-3-902613-24-0
1. Multi Robot Systems. 2. Recent Advances. I. Aleksandar Lazinica

Contents

9

Appearance-Based Processes in Multi-Robot Navigation

Luis Payá, Oscar Reinoso, José L. Aznar, Arturo Gil and José M. Marín
Miguel Hernández University
Spain

1. Introduction

When a mobile robot has to carry out an assigned task in a large environment, it has to take decisions about its localization and about the trajectory to follow to move from its current position to the destination point. To solve this problem a map or an internal representation of the environment is needed. This map should contain enough information to allow the robot to compute its current position and the necessary control action to lead it to its destination following, maybe, a chosen trajectory. In the last years, intensive research on this field, using SLAM techniques (Simultaneous Localization And Mapping) has been developed. This approach tries to build a global map of the environment while simultaneously determining the location of the robot. Usually, these approaches rely on the extraction of several landmarks or characteristic points of the environment either natural or artificial, as (Thrun, 2001) does.

Recent research in robotics has extended these concepts to applications that require the use of a team of robots. These applications require the coordination between the members of the team while they move within the environment. In some applications, the use of collaborative robotics clearly improves the performance, comparing to a single robot carrying out the same task. As an example, (Thrun, 2003) presents a probabilistic EKF algorithm where a team of robots builds a map online, while simultaneously they localize themselves. (Fenwick et al., 2002) take into account the problem of the concurrent mapping and localization with extra positional information available when multiple vehicles operate simultaneously. In (Ho & Newman, 2005), a map is built using visual appearance. From sequences of images, acquired by a team of robots, subsequences of visually similar images are detected and finally, the local maps are joined into a single map.

A typical problem in collaborative robotics implies a path following e.g. to perform a surveillance task in an office environment or an assembly or delivery task in an industrial environment. Also, the problem of formations, where a team of robots must navigate keeping a relative position in a structure of robots, can be seen as a problem of path following, where one or several robots must follow the path the leader is recording with an offset either in space or in time.

This problem of route following can be solved without necessity of creating complex maps of the environment. It is just needed a teaching step, where the route to follow is learned, and a navigation step, where the second robot follows the route just comparing its current sensory information with the data stored in the database. Classical approaches in this field are model-based approaches. In these approaches, the extraction of several landmarks or feature points along the images permits computing the image Jacobian, which relates the change of the coordinates in the image with the changes in motion in the ground plane. Then, using the principles of visual servoing, the second robot can follow the route, as in (Burschka & Hager, 2001). Also, in the behaviour-based control (Balch & Arkin, 1998), some features of the images are extracted to carry out the localization and navigation of the members of a team in a formation problem.

However, other approaches suggest that these processes could be achieved just comparing the general visual information of the images, without necessity of extracting any feature. These appearance-based approaches are specially useful for complicated scenes in unstructured environments where appropriate models for recognition are difficult to create. As an example, (Matsumoto et al., 1996) and (Matsumoto et al., 1999) address a method consisting on the direct comparison of low-resolution images. This method may lead to errors when the size of the route is quite long so, other features must be added to make the method more robust, such as histogram, texture and density of edges, (Zhou et al., 2003). However, these features do not contain any geometric information so they are useful just for localization but not for navigation. To solve it, in some occasions, a feature selection is accomplished to determine the relationship between the images (Boorj et al., 2006).

When working with the whole images, the complexity of the problem can be reduced by means of the PCA (Principal Components Analysis) subspace as in (Kröse et al., 2004) or (Maeda et al., 1997), where through PCA techniques a database is created using a set of views with a probabilistic approach for the localization. Other works have shown how this subspace reduction can be applied to other tasks in robotics, such in loop closure detection for SLAM tasks (Newman et al., 2006). In classical PCA approaches, all the views along the route must be available before the database can be created so the navigation of the second robot cannot begin until the leader has finished learning the route. Actually, a new model must be built from the scratch when we want to include information about new locations in the map. These problems can be overcome using an incremental PCA method, as shown in (Payá et al., 2007), or other kind of techniques to describe the environment, like working in the Fourier domain (Menegatti et al., 2004).

In this paper, we present an appearance-based method for route following where incremental PCA and Fourier Transform have been used to build the database, and a probabilistic Markov process has been implemented for robot localization during the navigation. In the next section, the problem to solve is presented. Then, in section 3, the representation of the environment along the route is detailed. After this, in section 4, the implementation for the navigation is exposed. The results of the experiments carried out with a team of mobile robots are presented in section 5. To finish, the conclusions of the work are exposed.

2. Description of the problem

2.1 Phases to accomplish

The main goal of the work is to create an architecture to carry out multi-robot route following using just visual information and with an appearance-based approach. In this task, a leader robot goes through the desired route while a team of robots follow it. To accomplish this task, two phases must be implemented: the learning phase and the navigation phase.

In the first phase, the leader robot is teleoperated through the route to follow and it stores some general visual information along this route. In a general way, a new image is acquired when the correlation of the current one goes down a threshold respect the previously stored so, images are stored more frequently when the information changes quicker (e.g. turnings). In this step, it is important to define correctly the representation of the environment to allow that any robot can follow the route of the leader one with an offset either in space or/and in time in an efficient way.

The main disadvantage of the appearance-based techniques is that they require huge amounts of memory to store the necessary information of the environment and they suppose high computational cost to make the necessary comparisons during the autonomous navigation so, the key points of these approaches reside on the type and quantity of information to store and in how to make the comparison between the current view and the stored information to reduce the computation time.

Previous works solved the problem by means of low-resolution images (Payá et al., 2005). The main drawback arises when the scenes used to define the route are highly unstructured and varying. In this case, it is necessary to increase the resolution to get an acceptable correlation in navigation. However, it would suppose an increase on the computational cost, so the average speed of the robot would be limited. In section 3, different alternative methods are proposed which are able to deal with high resolution images without slowing down the performance of the classifier.

On the other hand, once the database is created or while it is being built, during the navigation phase, the second robot is situated in a point near the learned route. Then, it has to recognize which of the stored positions is the nearest to the current one and drive to tend to the route, following it till the end. It must be carried out just comparing its current visual information with the information stored in the database. Two processes must run successively: auto-location and control. During the auto-location, the current visual information is compared with the information in the database, and the most similar point is taken as the current position of the robot. This decision must be made taking into account the aperture problem in office environments. This way, the new position of the robot must be in a neighbourhood of the previous one, depending on the speed of the robot.

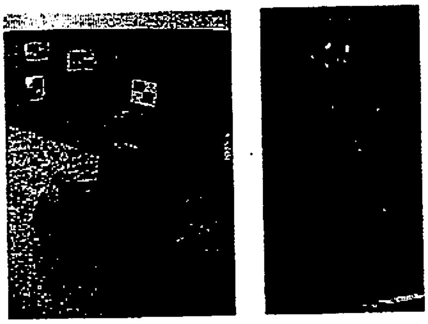


Figure 1. Pionner P3-AT robot and Catadioptric system that provides omnidirectional images

Once we have a new matching, in the control phase, the current visual information must be compared with the matched information of the database, and from this comparison, a control law must be deducted to lead the robot to the route.

In this application, we work with a team of Pioneer P3-AT robots that are equipped with a catadioptric system (fig. 1) consisting on a forward-looking camera and a parabolic mirror that provides omnidirectional images of the environment (fig. 2). To work with this information in an efficient way, the omnidirectional images are transformed to panoramic images, as shown on fig. 3. The size of these panoramic images is 256x64.

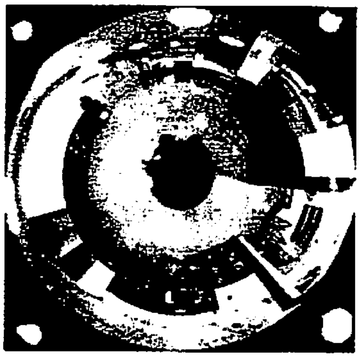


Figure 2. Omnidirectional view of the environment

Figure 3. Panoramic view of the environment

3. Representation of the environment

An appearance-based approach for robot navigation implies using the information of the whole images, without extracting any kind of landmark. If the auto-location process described in the previous section was carried out with high resolution images, the computational cost of the classifier would be unbearable. However, it is possible to compress the information of the image with a feature extraction method.

In visual object recognition systems, different techniques have been used to reduce the dimensionality of the data, allowing the classifier to work on a smaller number of attributes. The goal is to discard the irrelevant or redundant data and to keep as much information as possible. The most popular technique for dimensionality reduction purposes is Principal Component Analysis (PCA) (Oja, 1983); (Kirby, 2000). This technique has been used both in general object recognition applied to robotic tasks (Murase & Nayar, 1995); (Nayar et al., 1996) and in face recognition applications (Kirby & Sirovich, 1990); (Turk & Pentland, 1991). Another widely used technique is Linear Discriminant Analysis (LDA) which takes into account class memberships in computing the subspace (Belhumeur et al., 1997); (Swets & Weng, 1996). Dimensionality reduction is in fact a feature extraction process, as new features are extracted from the original attributes.

In our approach, the original images defining the route are compressed using three different techniques: PCA, Incremental PCA and Fourier Transform. This compressed information is used both in the localization and in the autonomous navigation steps.

3.1 PCA

The philosophy of the appearance-based methods consists in working with the general visual information of the images, without extracting any interesting point. Thus, this family of methods presents the disadvantages of the size of the database necessary to retain all the information of the environment and the computational cost of the comparisons between the whole images.

When working with 64x256 images, the data vectors fall in a 16384 dimensional space. However, all these data are generated from a process with just three degrees of freedom (position and orientation of the robot). This way, before storing the images, a reduction of the dimensionality of the data can be performed with the goal of retaining the most relevant information of each scene. Since pixels tend to be very correlated data, a natural reduction step consists on performing Principal Components Analysis (PCA), as in (Kirby, 2000).

Each image $\vec{x}_j \in \Re^{M\times 1}; j = 1 \ldots N$, being M the number of pixels and N the number of images, can be transformed in a feature vector (also named projection of the image) $\vec{p}_j \in \Re^{K\times 1}; j = 1 \ldots N$, being K the PCA features containing the most relevant information of the image, $K \leq N$. In traditional PCA, first of all, the data matrix is built using the images of the environment. The PCA transformation is computed from the covariance of the data matrix using SVD and the Turk and Pentland's method (Turk & Pentland, 1991). After the process, a new data matrix with the most relevant information is obtained.

The input data to the PCA transformation is the matrix X:

$$\begin{aligned} \mathbf{X} &= \left[\tilde{x}^1 \ldots | \tilde{x}^j | \ldots | \tilde{x}^N\right] \\ \tilde{x}^j &= \vec{x}^j - \bar{m} \\ \bar{m} &= \frac{1}{N}\sum_{j=1}^{N} \vec{x}^j \end{aligned} \tag{1}$$

where $\vec{x}^j$ is each representative view of the route (arranged in 1-column vector) and $\bar{m}$ is the mean of the set of N views.

The PCA transformation is computed from the covariance of the data matrix using equation 1, e.g. using SVD and the Turk and Pentland's method (Turk & Pentland, 1991). If the size of the data matrix is M x N, where N is the number of images along the route and M is the number of pixels of each image, and also $M >> N$, a total amount of N eigenvectors are obtained. The dimensionality reduction is done by:

$$P = V^T \cdot X \tag{2}$$

where V is the matrix containing the K principal eigenvectors and P is the reduced data of size K x N, where $K \leq N$. After this process, we have reduced each of the original images with M pixels to one vector with K components. The database will consist on these N vectors. Fig. 4 shows the structure of the database for a sample route, taking K=3. In this case, each image could be reduced to a point in the 3D space. The three coordinates of each point contain the most relevant information of the training images.

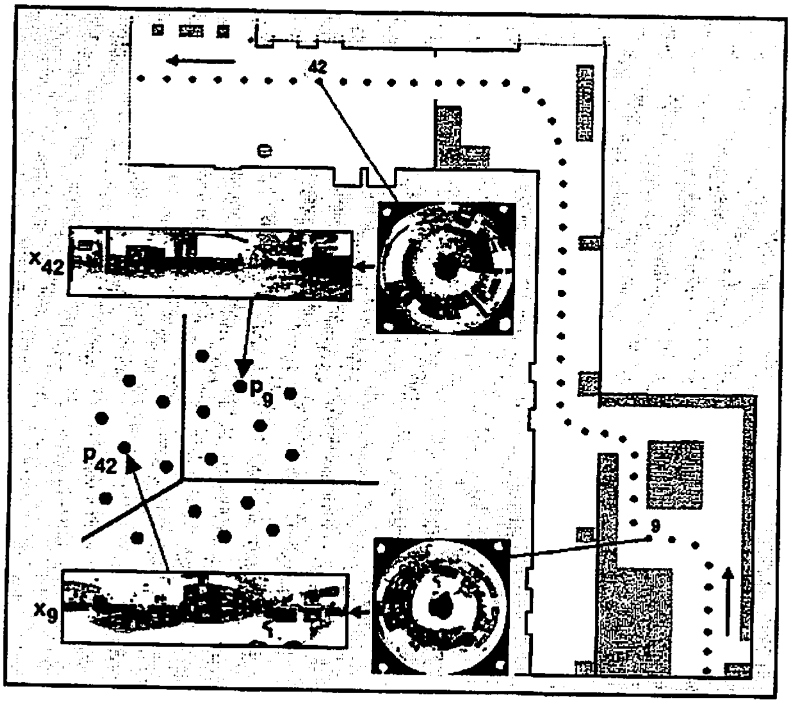


Figure 4. The route images are dimensionally reduced by means of PCA

The projection matrix P has part of the original image information but with lower size vectors, depending on the number of eigenvectors used in the transformation. If $K = N$, the information in P is the same as in X (the reconstruction error is 0 (Kirby, 2000)), but the dimension of the projected vectors is smaller: each projection, p_j, has just K components. Fig. 4 illustrates this approach.

3.2 Incremental PCA

In classical PCA approaches, all the images along the environment must be available before carrying out the compression. This way, the robots that follow the route should wait the leader one to run till the end. However, in collaborative tasks, usually it is necessary that some robots follow the first one while it is still recording the information.

With the PCA approach, the robot that is building the database should do it from the scratch when a new image along the route is captured, what is computationally very expensive. To overcome this disadvantage, a progressive construction of the database can be implemented. Several algorithms have been proposed to perform PCA in an incremental way (Artac et al., 2002).

As can be proved, when having a set of eigenvectors from a set of views, when a new image $\vec{x}_{N+1}$ is added to the database, these eigenvectors and the projections of the stored images can be updated following the next algorithm (Artac et al., 2002):

First, the mean must be updated:

$$\vec{m}' = \frac{1}{N+1}(N \cdot \vec{m} + \vec{x}_{N+1}) \tag{3}$$

Now, the set of eigenvectors must be updated so that they include the information of the new image $\vec{x}_{N+1}$. To do it, we compute the residual vector, that is the difference between the reconstruction and the original *N+1* image $\vec{h}_{N+1} = (V \cdot \vec{p}_{N+1} + \vec{m}) - \vec{x}_{N+1}$, that is orthogonal to the eigenvectors, and normalize it, obtaining $\hat{\vec{h}}_{N+1}$.

The new matrix of eigenvectors can be obtained by appending $\hat{\vec{h}}_{N+1}$ to *V* and rotating it:

$$V' = \left[V \cdot \hat{\vec{h}}_{n+1} \right] \cdot R \tag{4}$$

Being *R* the solution to the eigenproblem $D \cdot R = R \cdot \Lambda'$ and *D*:

$$D = \frac{N}{N+1} \cdot \begin{bmatrix} \Lambda & \vec{0} \\ \vec{0}^T & 0 \end{bmatrix} + \frac{N}{(N+1)^2} \cdot \begin{bmatrix} \vec{p} \cdot \vec{p}^T & \delta \cdot \vec{p} \\ \delta \cdot \vec{p}^T & \delta^2 \end{bmatrix} \tag{5}$$

where $\delta = \vec{h}_{N+1} \cdot (\vec{x}_{N+1} - \vec{m})$, $\vec{p} = V^T \cdot (\vec{x}_{N+1} - \vec{m})$ and Λ is a diagonal matrix containing the original eigenvalues. This way, if $V \in \Re^{MxK}$, then $V' \in \Re^{Mx(K+1)}$.

The image representations can be updated with the next expression:

$$\vec{p}_{i(N+1)} = (R')^T \cdot \begin{bmatrix} \vec{p}_{i(N)} \\ 0 \end{bmatrix} + \left[V \cdot \vec{h}_{N+1} \right]^T \cdot (\vec{m} - \vec{m}') \tag{6}$$

Fig. 5(b) shows the input data and the results of this approach in each iteration, comparing to the classical PCA (Fig. 5(a)).

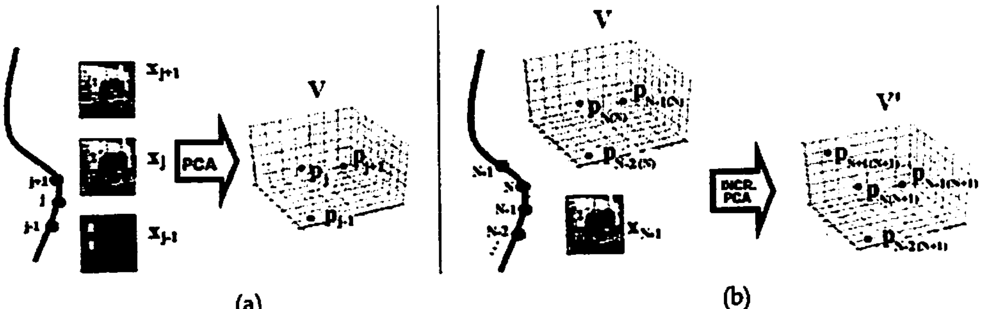


Figure 5. (a) Using PCA, the images of the route are dimensionally reduced. (b) Input and output data of the Incremental PCA process in each iteration. When a new image arrives. the projections in the database and the matrix *V* are updated

3.3 Fourier Transform

Another compact representation that we can use is a representation in the Fourier domain (Menegatti et al., 2004). To do it, each omnidirectional image is transformed into the panoramic one. Then, this panoramic image is expanded row by row into its Fourier Transform.

The sequence of complex numbers $\{a_n\}=\{a_0,a_1,\ldots,a_{N-1}\}$ can be transformed into the sequence of complex numbers $\{A_n\}=\{A_0,A_1,\ldots,A_{N-1}\}$ using the Discrete Fourier Transform with the following expression:

$$A_k=\sum_{n=0}^{N-1} a_n \cdot e^{-j\frac{2\pi nk}{N}}; \qquad k=0,\ldots,N-1 \tag{7}$$

The Fourier Transform computed in this way, row to row, presents two interesting properties for robot mapping and localization when omnidirectional images are used. First, the most relevant information in the Fourier domain concentrates in the low frequency components. These are the components whose magnitude is highest, as shown on fig. 7. This way, the information of the image is compressed on these components. Furthermore, removing high frequency information can lead to an improvement in localization because these components are more affected by noise.

The second interesting property is the rotational invariance. If two images are acquired at the same point of the environment but with different headings for the robot, then, the power spectrum of each row (the module of the Fourier Transform) is the same in both cases. This is because these two rotated omnidirectional images lead to two panoramic images THAT are the same but shifted along the horizontal axis (fig. 6). Each row of the first image can be represented with the sequence $\{a_n\}$ and each row of the second image will be the sequence $\{a_{n-q}\}$, being q the amount of shift, that is proportional to the relative rotation between images.

The rotational invariance is deducted from the shift theorem, which can be expressed as:

$$\Im[\{x_{n-q}\}]=X_k e^{-j\frac{2\pi qk}{N}}; \qquad k=0,\ldots,N-1 \tag{8}$$

where $\Im[\{x_{n-q}\}]$ is the Fourier Transform of the shifted sequence, and X_k are the components of the Fourier Transform of the non-shifted sequence. According to this expression, the Fourier Transform of a shifted sequence of numbers is equal to the Fourier Transform of the original signal multiplied by a complex number whose magnitude is the unit. This means that the amplitude of the Fourier Transform of the shifted image is the same as the original transform and there is only a phase change, proportional to the amount of shift q.

This way, when a new omnidirectional image is captured, first it is transformed into the panoramic image and then, the Fourier transform of each row is computed. This way, each 64x256 image is compressed to a sequence of 64xF numbers that are the magnitudes of the Fourier Transform components.

Each image $[x]_j \in \Re^{RxC}; j=1\ldots N$, being R the number of rows and C the number of columns, is transformed into a new matrix $[t]_j \in \Re^{RxF}; j=1\ldots N$, being F the first Fourier components (corresponding to the lower frequencies).

It must be taken into account that the Fourier Transform is intrinsically, an incremental procedure, due to the fact that to compress each scene it is not necessary any kind of information from the rest of scenes. This way, it also permits multi-robot navigation as Incremental PCA did and with the important property of rotational invariance.

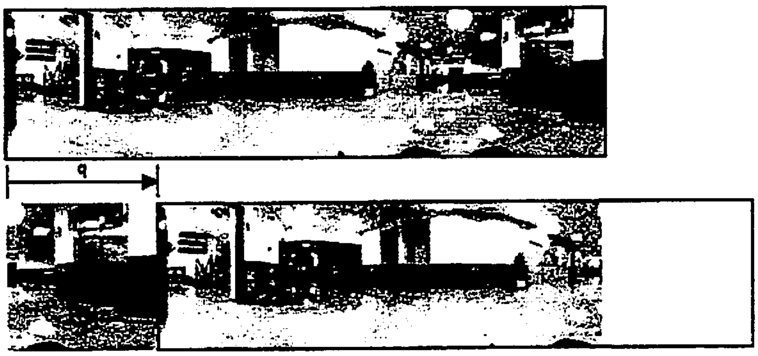


Figure 6. Two panoramic images taken in the same point but with different heading for the robot

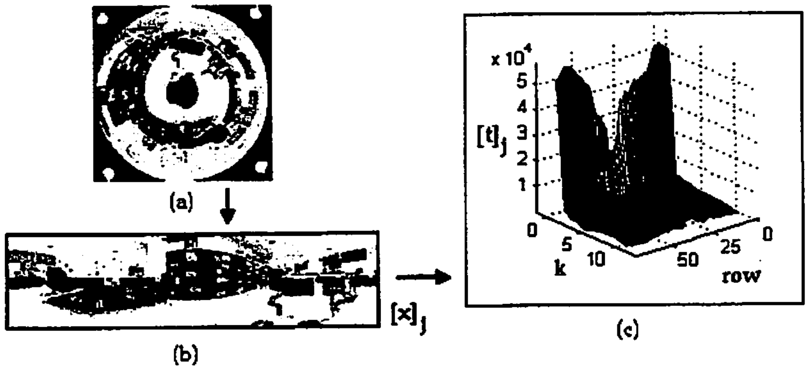


Figure 7. For each point on the route, the omnidirectional view (a) is transformed into the panoramic view (b). Then, the Fourier transform is computed row to row (c). The main information is in the module of the components of lower frequency so these components are retained in a matrix that identifies that point of the route

4. Robot navigation

After the Learning Step, the leader robot has created a database consisting on some feature vectors that are labelled as positions 1,..., *N*. For each position, two feature vectors represent the visual information taken at that position: the PCA projection and the Fourier Transform. Now, a different robot can follow the same route carrying out two processes: Auto-location and ControL

4.1 Auto-Location

When the follower robot captures a new image $[x]_i$, using this information it must know which of the stored points is the closest one.

Using the incremental PCA approach, first, the projection $\vec{p}_i$ of the current image on the current eigenspace calculated by the leader is computed. Then, this vector has to be compared with those stored in the database $\vec{p}_j$. The one that offers the minimum distance is the matching one, which corresponds to the current position of the robot. The criterion used here is the Euclidean distance.

$$d_{ij} = \sqrt{\sum_{l=0}^{k} (p_{il} - p_{jl})^2}; \quad j = 1 \ldots N \tag{9}$$

On the other hand, if we use the Fourier Transform approach, once the robot captures a new image, its Discrete Fourier Transform is computed row to row. This results in a new matrix $[t]_i$ that contains the magnitude of the low-frequency components of the transformed sequence. Then, this matrix is compared with the Fourier matrices in the database $[t]_j$ with the expression:

$$d_{ij} = \sum_{m=0}^{R-1} \sqrt{\sum_{l=0}^{F-1} \left(t_i^{ml} - t_j^{ml}\right)^2}; \quad j = 1 \ldots N \tag{10}$$

To normalize the values of the distances, a degree of similarity between projections is computed with eq. (11). The image of the database that offers the maximal value of γ_{ij} is the matching image, and then, the current position of the robot (the nearest one) is j.

$$\gamma_{ij} = \frac{\left(d_i^{\max}/d_{ij}\right) - 1}{\left(d_i^{\max}/d_i^{\min}\right) - 1}; \quad \gamma_{ij} \in [0,1]; \quad j = 1 \ldots N \tag{11}$$

where $d_i^{\max} = \max_{j=1}^{N}(d_{ij})$ and $d_i^{\min} = \min_{j=1}^{N}(d_{ij})$.

In office environments, that present a repetitive visual appearance, this simple localization method tends to fail often as a result of the aperture problem. This means that the visual information captured at two different locations that are far away can be very similar. To avoid these problems, a probabilistic approach, based on a Markov process, has been used. The current position of the robot can be estimated using the Bayes rule, with the next expression:

$$p(y|x;\theta) \propto p(x|y;\theta) \cdot p(y) \tag{12}$$

where $p(y)$ denotes the probability that the robot is on the position y before observing the image x. This value is estimated using the previous information and the motion model. $p(x\,|\,y)$ is the probability of observing x if the position of the robot is y. This way, a method to estimate the observation model must be deducted. In this work, the distribution $p(x\,|\,y)$ is modelled through a sum of Gaussian kernels, centred on the n most similar points of the route:

$$p(x_i \,|\, y_j) = \frac{1}{n} \cdot \sum_{j=1}^{n} \left(\gamma_{ij} \cdot e^{-\left(\frac{y - y_j}{\sigma}\right)^2} \right); \quad j = 1 \ldots N \tag{13}$$

Each kernel is weighted by the value of confidence γ_{ij}. Then, these kernels are convolved with a Gaussian function that models the motion of the robot, and that depends on the previous position and velocity of the robot. At last, the new position is considered at the point with highest contribution probability $\max_{j=1}^{N}(p(x_i \mid y_j))$.

Fig. 8 shows this process for *n=10* kernels. The previous position is the 28th. First, the ten most similar positions are selected. Then, a kernel function is assigned to each position. After that, the motion model is applied and at the end, the contribution of each kernel to each position is computed, selecting the point with the maximum contribution. In this case, despite the most similar location is the 10th, after the localization process the position selected would be the 28th.

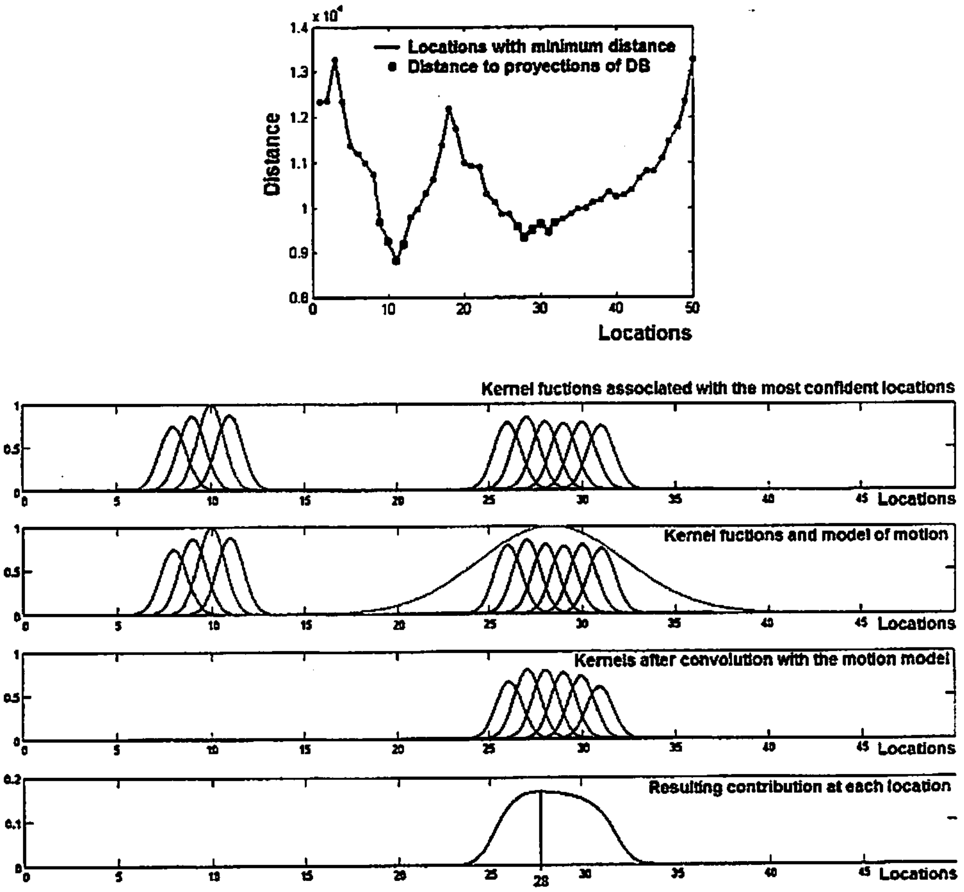


Figure 8. Improvement of the localization phase through a probabilistic-based approach. The new location depends on the previous one and the image acquired

After several experiments, the Fourier Transform approach has proved to be a more robust way to compute the current position of the robot. During a typical navigation problem, the time needed to compute the feature vector, compare with all of the vectors in the database and extract the matching point with the probabilistic approach is about 25% lower, comparing to the incremental PCA approach.

Also, the rotational invariance of the Fourier Transform is an important property when the robot starts the navigation. If at first, the follower robot is near the route but its heading is very different to the heading of the leader, the PCA approach would not be able to retrieve the correct position, because it does not present rotational invariance. Opposite to it, with the Fourier approach, the position is computed correctly so the robot is able to correct its heading and tend to the route. In the final implementation, only the Fourier Transform has been used for Localization.

4.2 Control

Once the robot knows its position, it has to compute its heading, comparing to the original heading that the leader had when it captured the image at that position. Then, the steering speed of the robot should be proportional to this relative heading.

When working in the Fourier domain, eq. 8 allows computing the shift q and so, the orientation of the robot. However, after some localization experiments, this expression has showed a very unstable behaviour, and it has not been able to deal with changes in illumination and noise. It only works well under very controlled conditions and if the follower robot is exactly on the same point where the leader captured the image.

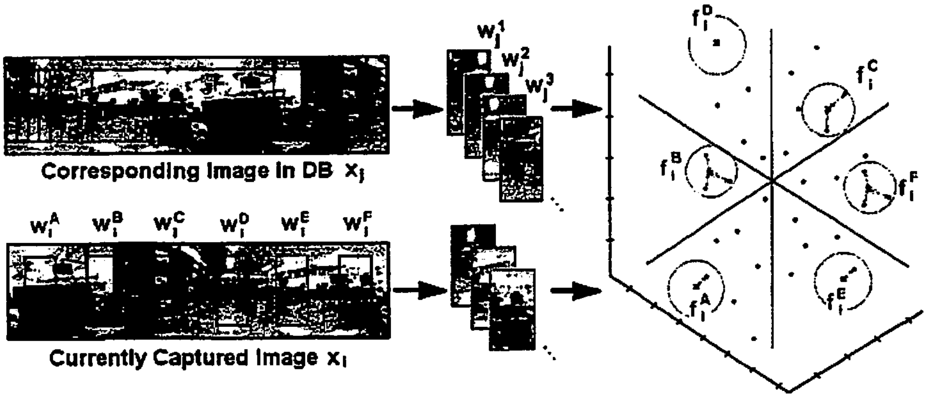
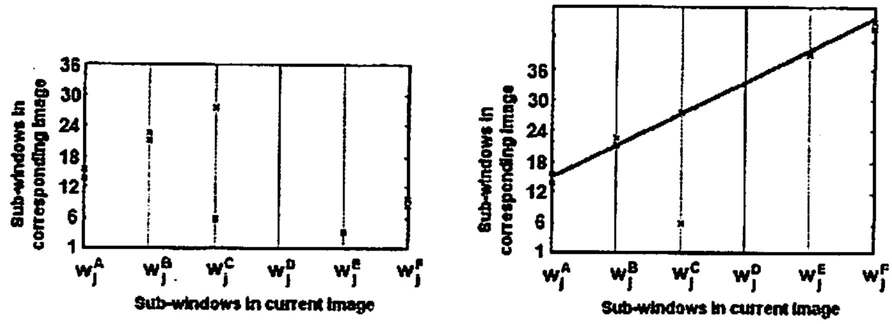

Corresponding Image in DB x_j
w_j^1
w_j^2
w_j^3
w_i^A w_i^B w_i^C w_i^D w_i^E w_i^F
Currently Captured Image x_i
f_i^D
f_i^C
f_i^B
f_i^F
f_i^A
f_i^E

Figure 9. Calculating the linear and steering speeds of the follower robot. When it captures a new image, six sub-windows are extracted and tracked over the corresponding image

The procedure used here to retrieve the orientation is the following. From each image stored in the database, j, a set of N' sub-windows is obtained from the whole image where $\bar{w}_j^i \in \Re^{M' x1}$, is each sub-window. The sub-windows are obtained scanning the original scene with a step in the horizontal axis (fig. 9). Carrying out a process of PCA compression, the PCA components $\bar{f}_j^i \in \Re^{K' x1}$, of each sub-image are calculated, where $K' \leq N'$. Fig. 9 shows these projections as red dots in the case $K'=3$. During the autonomous navigation, six sub-windows $(\bar{w}_j^A, \bar{w}_j^B, \bar{w}_j^C, \bar{w}_j^D, \bar{w}_j^E, \bar{w}_j^F)$ are taken on the currently captured view (fig. 9) and tracked over the central band of the matching image. To do this, once the robot knows its location, the PCA components of these six sub-windows are calculated. This operation returns six K'-components vectors $\left(\vec{f}_j^A, \vec{f}_j^B, \vec{f}_j^C, \vec{f}_j^D, \vec{f}_j^E, \vec{f}_j^F\right)$ that are shown as blue crosses on fig. 9. Then, the most similar projections to each of them are extracted. These most similar projections are those that fall in six spheres whose centres are $\left(\vec{f}_j^A, \vec{f}_j^B, \vec{f}_j^C, \vec{f}_j^D, \vec{f}_j^E, \vec{f}_j^F\right)$. The radius of these spheres is chosen so that a number of corresponding windows is extracted. In this work, a total number of ten sub-windows are extracted. The linear and steering velocities are inferred using a controller, whose inputs are the most similar projections to each of the six sub-windows. Analyzing these data and solving possible inconsistencies, the controller infers the linear and steering velocities of the robot to tend to the recorded route. To do it, once the most similar sub-windows are recognized, the controller tries to arrange them and look for a correlation that shows clearly if the robot has to turn left or to turn right to tend to the pre-recorded route.



(a) (b)

Figure 10. Computing the steering speed of the robot with the sub-windows approach. This speed is proportional to the ordinate in the origin of the fitted line

Fig. 10 shows an example of how the controller works. In fig 10(a), the blue crosses are the most similar sub-windows. On this figure, the most similar windows to $\vec{w}_j^A$ are the windows 15 and 16 $\left(\vec{w}_j^{15}, \vec{w}_j^{16}\right)$, the most similar to $\vec{w}_j^B$ are the windows 22 and 23, $\left(\vec{w}_j^{22}, \vec{w}_j^{23}\right)$, the most similar to $\vec{w}_j^C$ are 6 and 28 $\left(\vec{w}_j^{6}, \vec{w}_j^{28}\right)$, $\vec{w}_j^D$ has no correspondences, the most similar to $\vec{w}_j^E$ is the 3rd window $\left(\vec{w}_j^{3}\right)$ and at last, the most similar to $\vec{w}_j^F$ are the windows 9 and 10, $\left(\vec{w}_j^{9}, \vec{w}_j^{10}\right)$. This distribution of correspondences shows that the robot has to turn left so that the sub-windows fit with those of the corresponding image. After doing a least-square fitting with these points, the change in the heading of the robot should be proportional to the ordinate in the origin of the adjusted line, so that the robot tends to the position which the image was originally captured with. However, some considerations must be taken into account.

1. Usually, the line presents a discontinuity, because of the periodicity of the panoramic images. Then, it is necessary to correct some of the correspondences so that the fitting is correct, as shown on fig. 10(b).
2. Some correspondences may be outliers due to partial occlusions or little changes in the environment (position of the objects or illumination). These points should not be taken into account while computing the fitting.

3. As the six sub-windows are taken with a constant offset, the slope of the fitted line has to be near to N'/6. It must be taken into account while fitting the line.
4. The ordinate at the origin of the chosen linear regression shows how the steering of the robot should be to tend to the route correctly.

It must be taken into account that the linear speed must depend on the steering. If the steering is high, it means that the robot heading is very different to the heading the leader had, so the linear speed has to be low to permit the correction in the heading.

5. Results

Several experiments have been carried out to validate the approach. Fig. 11 shows a typical route of around 50 meters, recorded in an office environment and the route of the follower when it starts from two different points around it. Typically, the follower robot tends to the route and follows it, showing a great performance on the straight lines and a relatively bigger error in the turnings. However, with this approach, the robot is able to find the route and tend to it, showing a very stable behaviour till the end. Fig. 12 shows the evolution of the localization during the navigation of the follower robot and the probability calculated, what can be a measure of the precision.

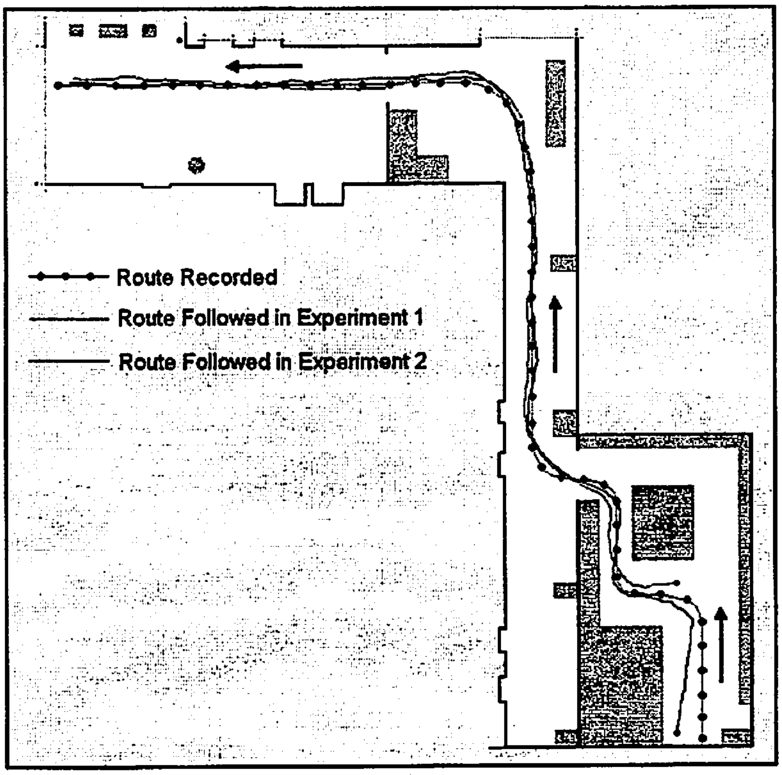

Route Recorded
Route Followed in Experiment 1
Route Followed in Experiment 2

Figure 11. Route recorded by the leader robot and routes followed by a follower robot in two experiments with different initial point

To carry out these experiments, two Pioneer P3-AT robots have been used with two processors onboard that communicate using a CORBA-based architecture where they interchange the necessary information. It is important to design an application where the different robots can share the necessary information in an easy and quick way due to the fact that the follower robot has to use continuously the database that the leader one is computing. An additional processor has been added to the architecture to carry out some calculations to reduce the computational cost of the processes in the robots.

On fig. 12, the localization shows a correct evolution, despite the aperture effect in such office environments, and the robot recovers correctly of some errors in localization (such as those in images 100 and 170). Also, the probability begins with quite low value (the robot is far from the route). Then, it tends to increase when the robot approaches to the straight line and decreases again in the turnings.

In the second experiment, the robot was initially close to the route, but it has a 180° orientation respect the leader. Despite this fact, the robot is able to localize itself using the Fourier approach, correct its heading, tend to the route and follow it.

Also, the sub-windows approach has showed to be robust when partial occlusions appear or when the original environment suffers small changes.

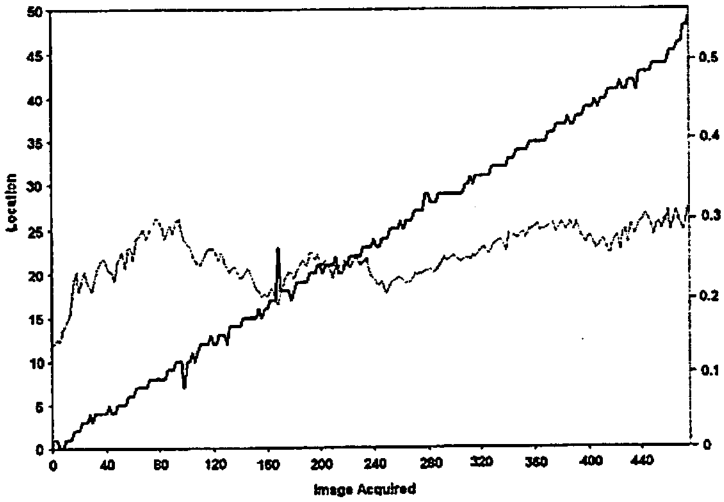


Figure 12. Evolution of the localization and the final probability $p(x \mid y)$ in the route followed during the experiment 1

6. Conclusions

In this paper, an appearance-based multi-robot following-route scheme is presented. The proposed solution uses low resolution panoramic images obtained through a catadioptric system, and techniques for dimensionality reduction to extract the most relevant information along the environment. To allow a new robot can follow the route that another robot is recording at the same time, an incremental PCA and a Fourier Domain algorithm are presented.

The objective of the work is that other robots can follow this route from a distance (as in space or in time). To do it, a probabilistic algorithm has been implemented to calculate their current position among those that the leader has stored, and a controller has been implemented, also based on the appearance of the scenes, to calculate the linear and turning speeds of the robot.

Some experiments have been carried out with two Pioneer 3-AT robots using a CORBA-based architecture for communication. These experiments show how the process employed is useful to carry out route following in an accurate and robust way.

We are now working in other control methods to reduce the error during the navigation, studying the effects of illumination changes and occlusions more accurately. Also, other techniques to compress the information are being analysed to achieve a higher speed of the follower robots.

At last, more sophisticated ways of building a map are being evaluated so that the robot can find the route and follow it although its initial position is far from this route. These more complex maps are expected to be useful in a number of applications in multi-robot navigation.

7. Acknowledgements

This work has been supported by MEC through project DPI2007-61197 "Sistemas de percepción visual móvil y cooperativo como soporte para la realización de tareas con redes de robots".

8. References

Artac, M.; Jogan, M. & Leonardis, A. (2002). Mobile Robot Localization Using an Incremental Eigenspace Model, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1205-1030, ISBN 0-7803-8232-3, Washington, USA, May 2002, IEEE

Balch, T. & Arkin, R. (1998). Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6, (Nov 1998) 926-938, ISSN 1042-296X

Belhumeur, P.N.; Hespanha, J.P. & Kriegman, D.J. (1997). Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, (Jul 1997), 711-720, ISSN 0018-9340

Boorj, D.; Zivkovik, Z. & Kröse, B. (2006). Sparse appearance based modeling for robot localization, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 1510-1515, ISBN 1-4244-0259-X, Beijing, China, IEEE

Burschka, D. & Hager, G. (2001). Vision-Based Control of Mobile Robots, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1707-1713, ISBN 0-7803-6578-X, Seoul, Korea, May 2001, IEEE

Fenwick, J.; Newman, P. & Leonard, J. (2002). Cooperative Concurrent Mapping and Localization, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1810-1817, ISBN 0-7803-8232-3, Washington, USA, May 2002, IEEE

Ho, K. & Newman, P. (2005). Multiple Map Intersection Detection using Visual Appearance, *Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems*, ISSN 0219-6131, Singapore, Dec 2005

Kirby, M. & Sirovich, L. (1990). Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No.1, (Jan 1990), 103-108, ISSN 0018-9340

Kirby, M. (2000). *Geometric data analysis. An empirical approach to dimensionality reduction and the study of patterns*, Wiley Interscience, ISBN 0-4712-3929-1

Kröse, B.; Bunschoten, R.; Hagen, S.; Terwijn, B. & Vlassis, N. (2004). Household robots: Look and learn. *IEEE Robotics & Automation magazine*, Vol. 11, No. 4, (Dec 2004) 45-52, ISSN 1070-9932

Maeda, S.; Kuno, Y. & Shirai, Y. (1997). Active navigation vision based on eigenspace analysis, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 1018-1023, ISBN 0-7823-0123-4, Grenoble, Francia, Sep 1997, IEEE

Matsumoto, Y.; Inaba, M. & Inoue, H. (1996). Visual navigation using view-sequenced route representation, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 83-88, ISBN , Minneapolis, USA, IEEE

Matsumoto, Y.; Ikeda, K.; Inaba, M. & Inoue, H. (1999). Visual Navigation Using Omnidirectional View Sequence, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 317-322, ISBN 0-7823-5184-3, Korea, Oct 1999, IEEE

Menegatti, E.; Maeda, T. & Ishiguro, H. (2004). Image-based memory for robot navigation using properties of omnidirectional images. Robotics and Autonomous Systems. Vol. 47, No. 4, (Jul 2004), 251-276, ISSN 0921-8890

Murase, H. & Nayar, S.K. (1995). Visual learning and recognition of 3D objects from appearance. *International Journal on Computer Vision*, Vol. 14, No. 1, (Jan 1995), 5-24, ISSN 0920-5691

Nayar, S.K.; Nene, N.A. & Murase, H. (1996). Subspace methods for robot vision, *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, (Oct 1996) 750-758 ISSN 0882-4967

Newman, P.; Cole, D. & Ho, K. (2006). Outdoor SLAM using Visual appearance and laser ranging, *Proceedings of the International Conference on Robotics and Automation*, pp. 1180-1187, ISBN 0-7803-9506-9, Orlando, USA, May 2006, IEEE

Oja, E. (1983). *Subspace methods of pattern recognition*, Research Studies Press and J. Wiley, ISBN 0-8638-0010-6.

Payá, L.; Vicente, M.A.; Navarro, L.; Reinoso, O.; Fernández, C. & Gil, A. (2005). Continuous navigation of a mobile robot with an appearance-based method, *Proceedings of the second International Conference on Informatics in Control, Automation and Robotics*, pp. 443-446, ISBN 972-8865-30-9, Barcelona, Spain, Sep 2005

Payá, L.; Reinoso, O.; Gil, A.; Pedrero, J. & Ballesta, M. (2007). Appearance-Based Multi-Robot Following Routes Using Incremental PCA, *Proceedings of Knowledge-Based Intelligent Information and Engineering Systems*, pp. 1170-1178, ISBN 3-540-74828-8, Vietri sul Mare, Italy, Sep 2007, Springer

Swets, D.L. & Weng, J.J. (1996). Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, (Aug 1996), 831-836, ISSN 0018-9340

Thrun, S. (2001). A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, Vol. 20, No. 5, (2001), 335-363, ISSN 0278-3649

Thrun, S. (2003). Robotic Mapping: A Survey, In: *Exploring Artificial Intelligence in the New Milenium*, 1-35, Morgan Kaufmann Publishers, ISBN 1-55860-811-7, San Francisco, USA

Turk, M. & Pentland, A. (1991). Eigenfaces for recognition. *Journal on Cognitive Neuroscience*, Vol. 3, No. 1, (Jan 1991) 71-86, ISSN 0898-929X

Zhou, C.; Wei, T. & Tan, T. (2003). Mobile robot self-localization based on global visual appearance features, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1271-1276, ISBN 0-7803-7737-0, Taipei, Taiwan, Sep 2003, IEEE