34(1) 1–68 (2001)

ISSN 0921–8890

Autonomous Systems

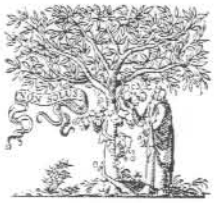
Volume 34 Number 1 31 January 2001

Elsevier

Robotics and Autonomous Systems

Volume 34, Number 1

31 January 2001



ELSEVIER

Amsterdam – London – New York – Oxford – Paris – Shannon – Tokyo

ELSEVIER

Robotics and Autonomous Systems 34 (2001) 23–38

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

A robot self-localization system based on omnidirectional color images

Alessandro Rizzi*, Riccardo Cassinis

Department of Electronics for Automation, University of Brescia, Via Branze 38, I-25123 Brescia, Italy

Received 7 July 1998; received in revised form 15 March 2000; accepted 11 April 2000

Abstract

A self-localization system for autonomous mobile robots is presented. This system estimates the robot position in previously learned environments, using data provided solely by an omnidirectional visual perception subsystem composed of a camera and of a special conical reflecting surface. It performs an optical pre-processing of the environment, allowing a compact representation of the collected data. These data are then fed to a learning subsystem that associates the perceived image to an estimate of the actual robot position. Both neural networks and statistical methods have been tested and compared as learning subsystems. The system has been implemented and tested and results are presented. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Self-localization; Omnidirectional sensor; Visual navigation; Mobile robots

1. Introduction

In the field of visual guidance of autonomous robots, omnidirectional image sensors have been intensively studied [19,20,22]. Having an omnidirectional field of view in a single camera shot is an appealing feature, but an omnidirectional device adds heavy geometric distortions to the perceived scene. In this case, a geometrical model based on navigation method originates complex tasks, thus wasting the acquiring facilities of the omnidirectional device. Exploration and map building using this approach have the same problem as well [9].

A first simplification derives from the use of qualitative methods [17]. Complex visual navigation tasks in real environments can be managed more easily by not trying to recognize objects around the robot, but simply memorizing snapshots from specific places and then correlating them with the currently perceived image [6,12]. The typical problem of this approach is the large amount of memory needed for learning a route in an environment.

An omnidirectional image can be obtained by means of different devices. Besides fish-eye lenses [15], several mirror-based devices such as COPIS [18,19,21], MISS [20] and HOV [22] have been developed and studied taking into account the different geometric viewfield distortions, but little attention has been paid to the reflectance characteristics of the various devices.

The central idea of the present work is to use a conical device, similar to COPIS, but with a different reflecting surface that allows the collection of simplified omnidirectional visual information (Fig. 2).

The proposed system derives from an earlier joint research program started at the Laboratory of Automation, University of Besançon (France) [1,2] and then

* Corresponding author. Tel.: +39-030-3715-469; fax: +39-030-380014.
E-mail addresses: alessandro.rizzi@unibs.it (A. Rizzi), riccardo.cassinis@unibs.it (R. Cassinis).

PII: S0921-8890(00)00103-2

continued at the Department of Electronics for Automation, University of Brescia (Italy) [3–5].

2. Aim and structure of the system

The aim of the system is to support the navigation of an autonomous mobile robot estimating its position in a previously learned working area.

Neither conditioning of the environment around the working area nor accurate distance measurements are required and no explicit and detailed maps of the world surrounding the robot are made.

An interesting feature of the proposed system is its ability to work even if the learned environment changes. This holds true, provided that changes affect only limited portions of the whole vision field [5] (e.g. people walking around, objects that were not present during the training phase or that were removed thereafter, etc.).

The system is designed to work on a car-like autonomous mobile robot and thus its position in a two-dimensional space can be described by three co-ordinates $\langle x, y, \theta \rangle$, representing the position of the center point of the robot and its rotation with respect to a fixed reference system. The system provides only $\langle x, y \rangle$ co-ordinates. In order to make things simpler θ has not been taken into account so far and the heading of the robot has been kept constant by other means. However, due to the system robustness, small undetected rotations (less than 5° in either direction) can be easily tolerated [5].

The whole system can be split into three subsystems: an optical subsystem that collects and modifies the omnidirectional visual information, a pre-processing subsystem that further simplifies the data and a learning subsystem. The overall system structure is shown in Fig. 1.

3. The visual perception subsystem

The visual information reflected by the cone is grabbed with a CCD color camera facing upwards. The cone-shaped mirror is placed at a known distance, coaxially to the camera lens and the camera is focused on the cone surface. The conical mirror collects information from the environment around the robot in each direction orthogonal to the cone–camera axis. The resulting image is an omnidirectional view of a horizontal section of the environment around the robot.

The cone surface reflectance characteristics and the geometric distortions in the perceived image do not allow the vision system to identify any object and the self-localization is obtained from the perceived image without any attempt to detect and recognize objects in the scene. In fact the surface of the cone is not a perfect mirror and the perception system is not designed to obtain a perfect image of the environment. Fig. 2 shows five conical mirrors with different surface finishing (the smoothest at top left) and the way they reflect the same scene.

A large amount of details in the image obtained by a perfect mirror would increase the image elaboration

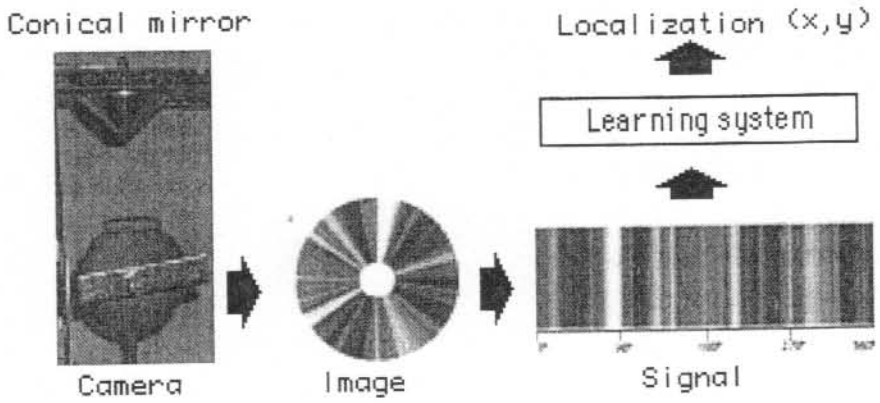


Fig. 1. Structure of the system.

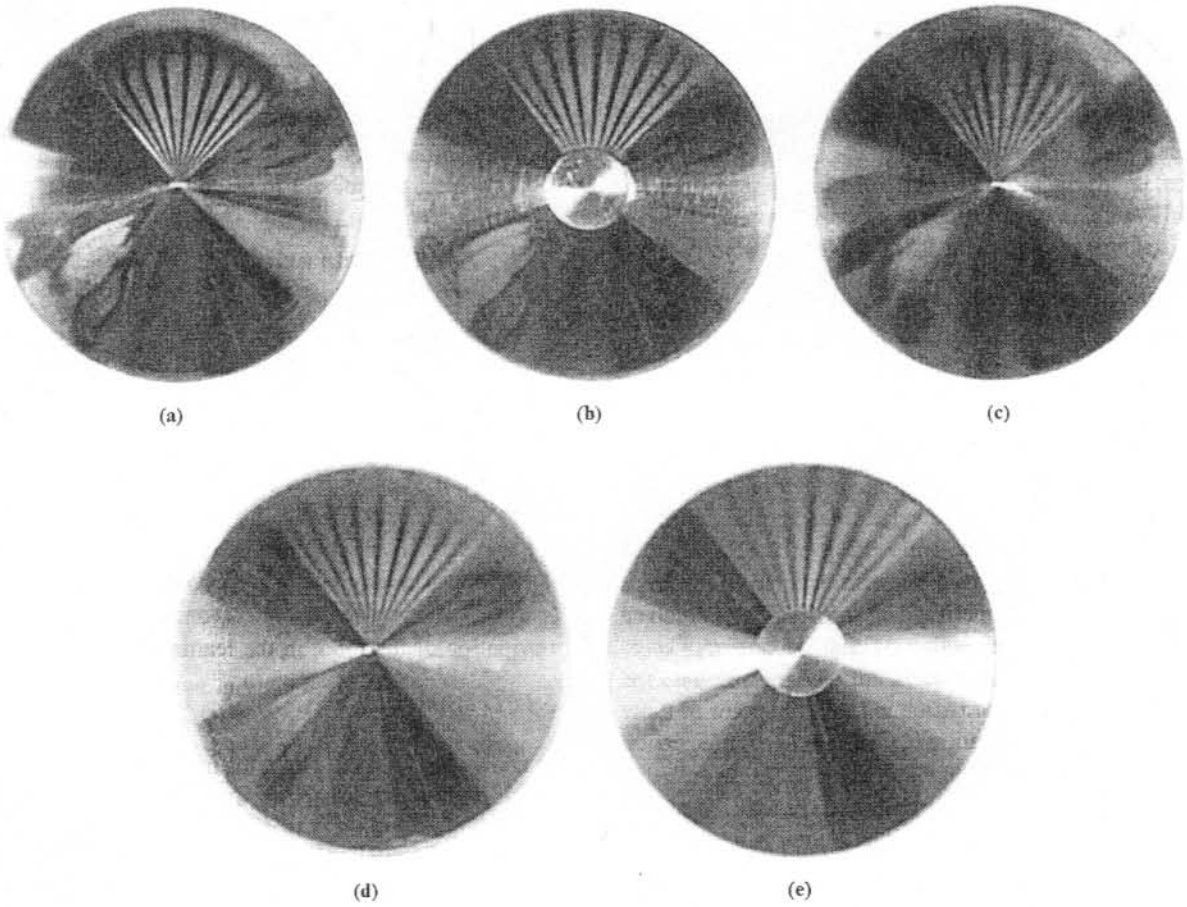


Fig. 2. Images from different cone surfaces.

complexity. As high spatial frequencies are less influent on the performance of the learning system, it is convenient to enhance the image blur. The surface diffusion acts as a low-pass filter [11] and is responsible for the loss of information related to those objects in the environment that are small or far away from the robot. For these reasons, it was decided to use the roughest available surface (bottom right in Fig. 2).

Diffused and constant lighting conditions are initially assumed in order to eliminate the effects of time-varying and space-varying or color-varying lighting. Pollicino has been devised for indoor environments such as offices, where the main source of lighting is artificial and can be considered almost constant. As any vision system, even this one is not completely independent from changes in the environment illumination geometry.

4. The image pre-processing subsystem

The main goal of the image pre-processing is to extract only a small amount of meaningful data from the grabbed omnidirectional image.

Two kinds of pre-processing are performed on the environment data: an optical pre-processing due to the reflectance characteristics of the cone surface, before the image grabbing, and a subsequent extraction of the mean angular chromatic value from the grabbed image. As it can be seen in Fig. 2, as the surface specular-

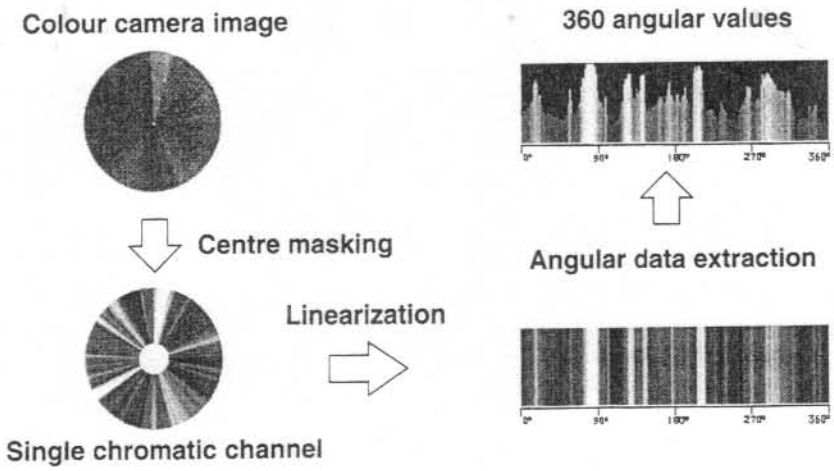


Fig. 3. The image pre-processing.

ity decreases, the image of each angular sector tends to be a vertical superimposition of all the perceived objects in a given direction. With the decrease of specularity fewer single objects are recognizable and the only perceived information is the mean color of these objects. The information obtained with a non-specular cone enhances the color and the brightness reflected by large surfaces in the environment, and large surfaces are more useful than small ones in determining robot localization.

After the image grabbing, the pre-processing subsystem divides the perceived image into the RGB chromatic components. The pixels around the center of the cone do not carry useful information and are discarded (Fig. 3).

A reference system transformation is then applied from rectangular to polar coordinates and the circle of the original image is mapped onto a rectangle (Fig. 3). Each vertical stripe of the rectangle contains pixels representing the chromatic characteristics perceived in the related direction. Then each chromatic channel image is split into 360 sectors of 1°. For each sector and for each channel the mean chromatic lightness value is extracted.

5. System operation

Pollicino uses the data extracted by the omnidirectional vision system in two different ways. In fact the system operation involves two phases: the supervised learning of the working areas, and the autonomous navigation of the robot in the learned areas.

5.1. Learning phase

During the learning phase the mobile robot is guided all around the operating area. In this phase, the system must acquire a number of images at known positions in order to train the learning subsystem.

These data should be collected, either in a regular or irregular way, all around the area to learn. The tests, presented in the following sections, use regular grids to acquire the images to train the learning subsystem. Variations in the shape of these grids do not affect the system performance. Influence of image acquisition densities for the learning set in the output error has been investigated.

If the area to be learned is large, complexity can be kept low by splitting the area into several sections. The learning phase can be realized by teleoperated or externally guided procedures.

5.2. Execution phase

During the execution phase, the mobile robot can use Pollicino to localize itself in the previously learned area. Once an image from the environment has been grabbed, the localization system uses the information organized and stored in the learning subsystem to obtain an estimate of the actual position.

If the robot enters a new subsection of a long path, the corresponding new mapping data, instead of the old ones, will be used for the learning subsystem. The system can easily switch between different learned areas, partially superimposed, when the estimated position falls near the border of the current area.

6. The learning subsystem

The learning subsystem carries out an inverse mapping between the position of the mobile robot in a subsection of the working area and the data extracted from the omnidirectional image corresponding to that position. The performance of the whole localization system is measured by the error between the actual position and the one estimated by the system.

Alternative learning systems based on neural networks and on statistical methods have been tested.

6.1. Neural network approach

The tested neural networks are supervised learning, feed-forward, and back-propagation neural networks. This kind of network is often used to map an arbitrary input space to an output space when output values corresponding to teaching input are available [14]. If the teaching input patterns are not too different, the kind of proposed network provides a useful generalization property that maps similar input patterns into similar output ones. This type of network also exhibits a property of graceful performance degradation that allows a correct operation of the learning system even if the input pattern is partially corrupt [14]. This implies that the system can localize the robot even if the environment changes, provided that the changes affect limited portions of the whole vision field [5]. All these properties enable the neural network to build an approximation of the desired mapping function using a limited number of sample patterns in the learning phase.

6.2. Statistical approach

Two well-known statistical methods have been compared with the neural networks: multiple linear regression (MLR) [13] and principal component regression (PCR) [10]. The same set of images used to train the neural networks has been used to calibrate the models obtained by statistical methods. In order to compare all the different methods, the same performance index set was used to evaluate models. Multiple linear regression is tested here to predict co-ordinate values from a linear combination of the chromatic characteristics perceived, used as independent variables. The PCR method applies the MLR to the same variables in a changed space allowing a reduction of the number of such variables.

6.3. The first learning system

The first learning system, called STDBP and used in early stages of the research, was a feed-forward neural network, trained with a back-propagation algorithm [14]. Its structure is the same as STD3BP, presented in the sequel, but, due to the fact that the first system used a b/w CCD sensor, with an input layer simplified to 360 input units, each one corresponding to the mean lightness value of each 1° angular sector.

6.4. A neural network for color images

A modified version of the first neural network has been developed to deal with color images. It is called STD3BP and, as the previous one, it is a feed-forward neural network trained with a back-propagation algorithm. STD3BP derives directly from STDBP, with an input layer three times larger. It is composed of 360×3 units, one for each 1° wide angular sector and for each chromatic channel. This network, shown in Fig. 4, has two hidden layers and an output layer of two output units for the robot co-ordinates. Each hidden layer is completely connected to the following layer and it is composed of units with a sigmoidal activation function. Each unit in the first hidden layer gets its inputs only from a group of units of the input layer, thus forming a cluster, with input clusters partially superposed. This superimposition makes the neural network more dependent on the angular position of the values, allowing a more precise estimate of the robot position.

6.5. Opponent color model neural network

The last neural network, called BIONET, is inspired by a color perception theory and its input layer is real-

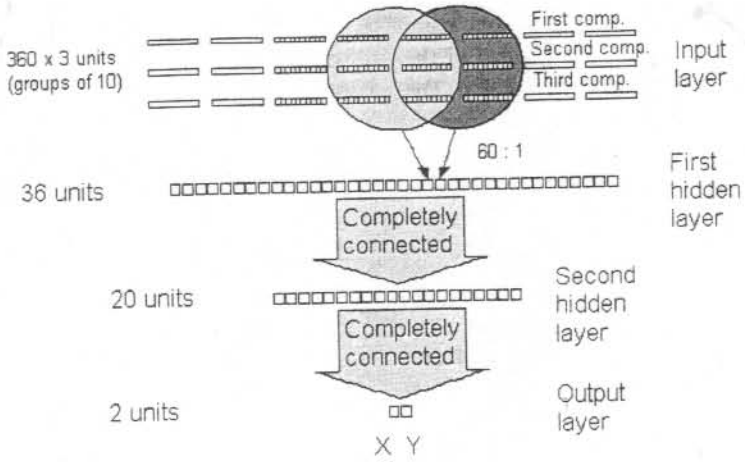


Fig. 4. STD3BP structure.

ized according to the relative neural model of the human visual system (retina and cortex) [8,16], shown in Fig. 5. This model can be divided into two stages. The first stage collects chromatic inputs coming from the three different types of retinal cones, long (L), intermediate (I) and short (S). The tri-stimulus theory associates them approximately with the RGB chromatic channels. The following stage of the model is made of two "opposite chromatic" units (red–green [r–g] and yellow–blue [y–b]) and one "opposite achromatic" unit (black–white [w–bk]). The links between the two stages are realized by excitatory (+) and inhibitory (−) connections. This model is implemented in BIONET like in Fig. 6, with non-adaptive connections between the input layer and the first hidden layer.

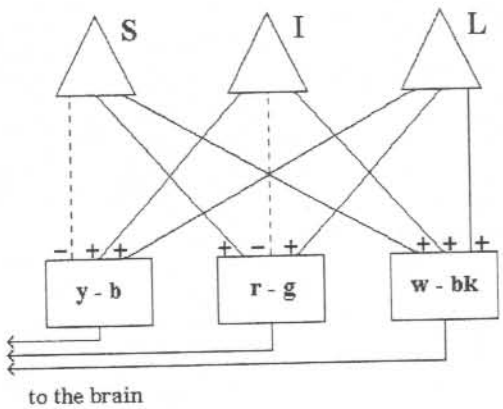


Fig. 5. Opponent color perception neural connections model (BIONET).

6.6. Statistical methods

The same image data sets that were used for neural network tests were also used for the statistical methods. For these methods it was not possible to use 360 sectors as in neural network tests, since the average number of acquired images was not enough for the calibration. For both MLR and PCR methods a choice criteria for the number of sectors to use has been

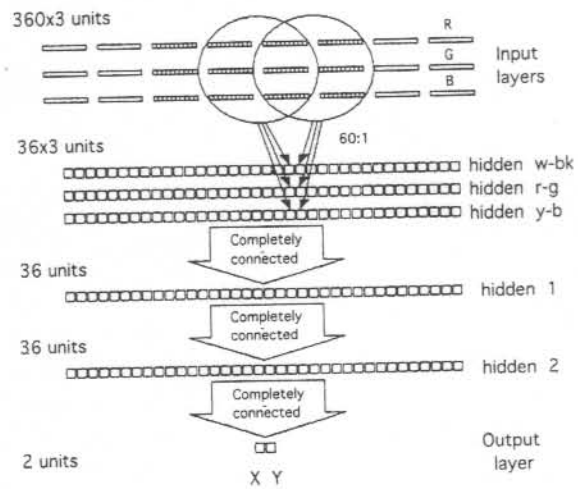


Fig. 6. BIONET structure.

adopted. It is presented in Section 9 and the resulting values are shown in the tables of Section 9. According to the number of sectors, to the density and to the chromatic component, the average pixel values in each sector has been computed and used as angular data.

The MLR method uses the patterns in the training set to find the coefficients of the linear function that estimates robot co-ordinates. This method performs a mapping from the mD-space of the input patterns to the 2D-space of the output co-ordinates. The number of input variables m depends on the number of patterns in the training set that must be greater than m.

Two functions $f_x, f_y : \mathbb{R}^m \rightarrow \mathbb{R}$ are seek, separately one for each of the two co-ordinates x and y so that

$$f_x(v_i) \cong x_i, \qquad f_y(v_i) \cong y_i,$$

where $v_i \in \mathbb{R}^m$ is the image vector and (x_i, y_i) the estimate point. The $2m$ coefficients of the two linear regressions

$$x_i \cong \sum_{j=1}^{m} a_j v_{ij}, \qquad y_i \cong \sum_{j=1}^{m} b_j v_{ij}$$

are determined using the least squares method as follows:

$$a_{1\cdots m} = (XX')^{-1} X\boldsymbol{x}, \qquad b_{1\cdots m} = (XX')^{-1} X\boldsymbol{y},$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the training set image co-ordinates and the matrix $X_{[m\times m]}$ contains all the v_i patterns of the same training set.

The second statistical method, PCR, uses the principal component analysis (PCA) technique to reduce the dimensions of the input space on which MLR is then computed [7].

For the PCR the new components have been computed and the number of principal components to use have been chosen according to the criteria described in Section 9. The MLR has been later applied on the values of the chosen principal components.

6.7. Memory requirements

Each image, pre-processed as it was said in Section 4, requires 1080 bytes, since it consists of 360 24-bits samples. The amount of memory required to store all the image vectors to learn an area depends on the chosen learning density (see Section 7). For a typical navigation task, 20–50 kbytes can be a reasonable estimate. In the tests discussed in Section 7, the neural networks described above have been realized with Stuttgart Neural Network Simulator (SNNS), but can be implemented with a standard programming language. A program that realizes the neural network needs a floating point variable for each node and for each link. The number of bits for each variable affects the precision of the system. STD3BP has 1138 nodes and 2920 links; using a long double (12 bytes using *gcc* on Pentium under Linux), the total amount of memory is less than 50 kbytes. BIONET has the same structure with an added level that performs the opponent color computation, and with a larger second hidden layer. Its nodes are 1262 and its links are 11 736. The memory required in this case is less than 160 kbytes.

7. Test of the system

The system has been tested with the acquisition device mounted on the robot, as visible in Fig. 7, and the computations have been performed off-line and on board as well.

Some indoor environments have been chosen and images all over the test areas have been collected regularly over a grid scheme and in some random points.

A subset of the collected images has been used as a training set for the learning systems, while the others have been used to test the system performances. Pollicino was also tested at random points. In order to fully understand the meaning of the tests, it should be kept in mind that Pollicino is not a complete navigation system by itself, but only a self-localization module that should be integrated with other sensors to provide actual navigation and obstacle avoidance capabilities.

In order to test the system performance, the data corresponding to the pre-processing of each image have been fed to the learning system and the difference between the system output and the expected output has been collected.

All the neural networks and statistical methods chosen have been tested on different sets, in different conditions. The system performances in several tested indoor environment have always shown comparable results. Thus, only two representative test areas (A1 and A2) have been reported.

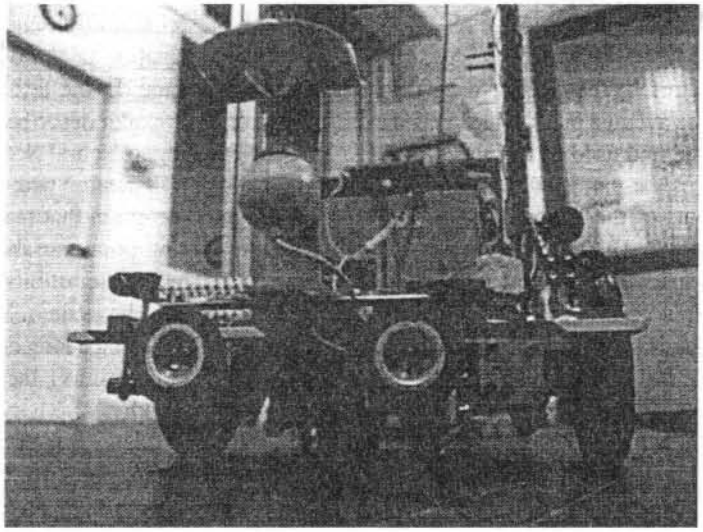


Fig. 7. The mounted acquisition device.

7.1. A1 test area

The A1 test area is a an indoor environment (the robotics lab) with no conditioning. The objects have their normal color and position and their disposition can be seen in Fig. 8. 27 images were taken over a sampling regular grid. From these 27 images three learning sets with different sampling densities have been chosen. The high density set is composed of 15 images, the mid density of 12 images and the low

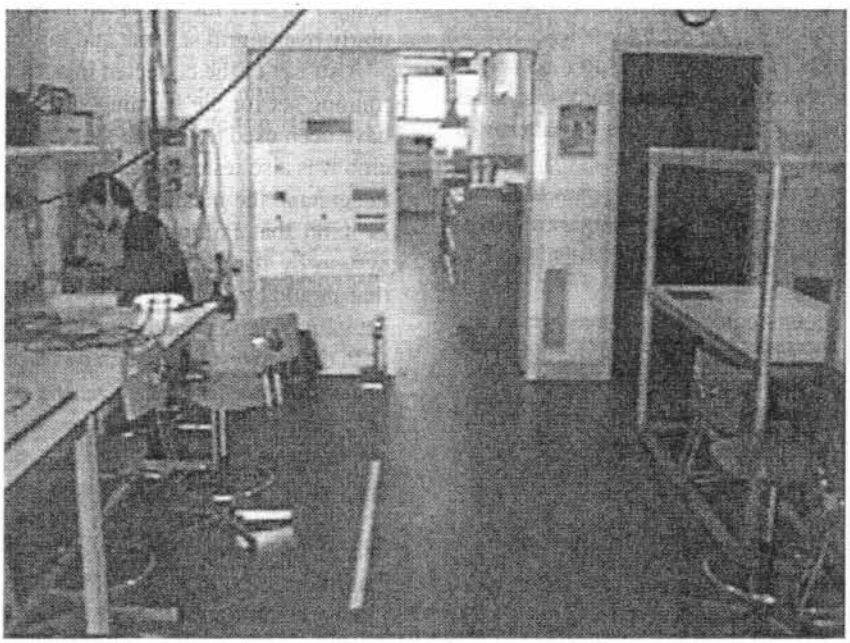


Fig. 8. A1 test area.

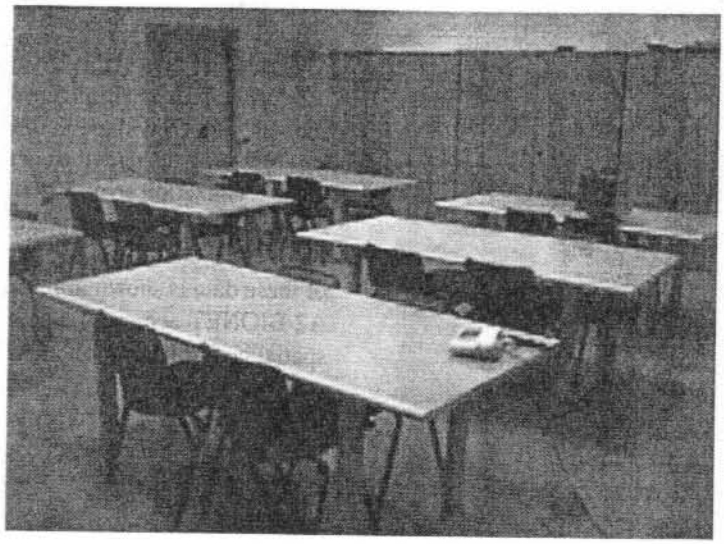


Fig. 9. A2 test area.

density of nine images. The total area dimension is $270 \times 90\,cm^2$.

7.2. A2 test area

Test area A2 (Fig. 9) is settled in a briefing room with tables and chairs. It is smaller than A1 and a more accurate sampling has been done. 96 images were taken in a rectangular test area along a regular grid and at some random points. In the same way as A1 three density sets have been chosen, with 48, 16 and 9 images. The total area size is $110 \times 70\,cm^2$.

8. Neural networks results

In this section the localization results of the system, tested using neural networks in the learning subsystem, are presented.

8.1. A1 results

As can be seen in Fig. 8, the first tested area A1 is an indoor environment in which a robot can easily navigate. In Table 1 the results obtained from STDBP using the single chromatic channels or the Bk/W input, from STD3BP and from BIONET are shown. All these

Table 1
A1 test results[a]

Color components	High density		Mid density		Low density		Learning system
	Err.	Dev.	Err.	Dev.	Err.	Dev.	
R	13.28	4.95	14.63	4.95	15.53	9.23	STDBP
G	15.30	5.63	16.65	6.98	16.20	6.98	STDBP
B	22.95	10.13	20.93	10.35	28.80	11.70	STDBP
Bk/W	18.00	8.10	17.33	7.43	17.10	8.78	STDBP
RGB	11.25	5.18	15.98	4.95	13.05	7.43	STD3BP
HSL	22.50	9.45	27.68	14.85	23.85	12.15	STD3BP
RGB	8.10	3.60	15.08	7.60	18.00	7.65	BIONET

[a]Localization errors in centimeter.

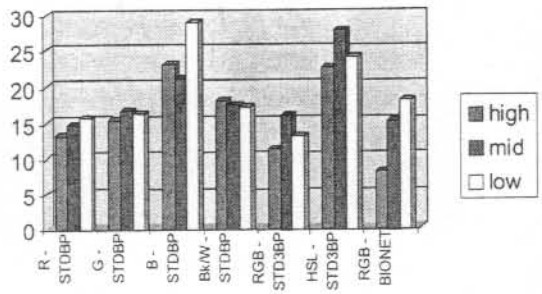


Fig. 10. A1 result comparison (cm).

data have been summarized in the chart of Fig. 10. An example of spatial error distribution in A1 is shown in Fig. 11. The plotted short lines connect the actual positions to their estimate, indicating the localization error. The mean error of almost all the tests seems not to be heavily affected by the density of the learning set. The transformation of the color information in the HSL space highly increases the mean error. This is due to the intrinsic high variance characteristics of the hue channel values. Given these results, HSL has not been used in further tests. The best results are achieved by BIONET.

8.2. A2 results

The tested area A2 is visible in Fig. 9. It is a typical office-like environment with desks, chairs and shelves, lighted with neon tubes. The results are comparable with the previous tested area, with a lower mean localization error. In Table 2 the results obtained from STDBP with the single chromatic channels, from STD3BP and from BIONET are shown. A comparison of these data is shown in Fig. 12. Also, in the test area A2 BIONET achieve the best results. An example of spatial error distribution in A2 can be seen in Fig. 13.

Moreover, some visual data collected in random points have been tested in order to verify how much the system is able to estimate the robot co-ordinates out of the regular grid of the learning set. The results, shown in Table 3 and in Fig. 14, indicate almost the same performances in and out of the sampling grid.

9. Statistical learning systems results

9.1. Test procedure

The neural network results presented in Section 8 have been compared with the results of the statisti-

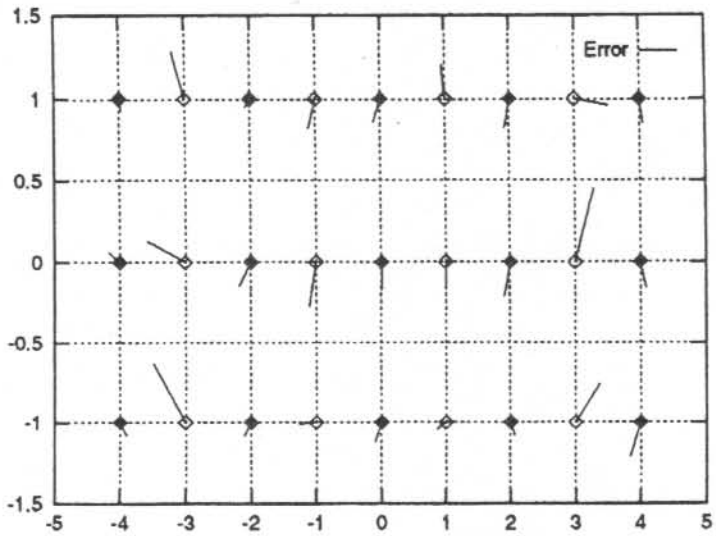


Fig. 11. Errors example in A1 with high density.

Table 2
A2 test results[a]

Color components	High density		Mid density		Low density		Learning system
	Err.	Dev.	Err.	Dev.	Err.	Dev.	
R	5.42	3.70	7.55	5.12	9.59	5.66	STDBP
G	3.84	2.46	8.06	4.47	10.65	5.06	STDBP
B	7.53	4.19	10.04	6.14	13.34	8.65	STDBP
RGB	4.00	2.64	7.77	4.45	13.61	9.98	STD3BP
RGB	2.70	1.55	4.96	2.96	7.59	4.08	BIORETE

[a]Localization errors in centimeter.

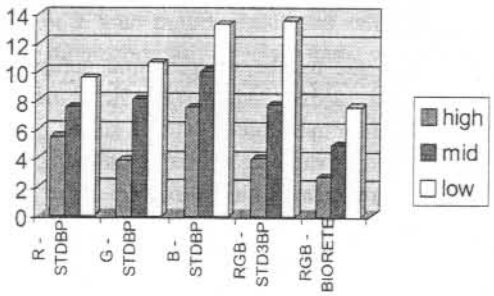


Fig. 12. A2 result comparison (cm).

cal methods described. The pre-processed images are divided into a number of sectors lower than the neural network tests, as they are the independent variables of a model that have to be calibrated by a small number of images. The number of sectors is fixed in about half the number of calibration images. The dependent variables are still the robot co-ordinates. Only the tests performed in the area A2 are presented.

9.2. MLR data sets choice

The mean localization error vs sector number in high density is shown in Fig. 15. The fitting curve of these data is very smooth and for a wide range of sector number the mean localization error remains almost steady. This fact supports the selecting criteria of the number of sectors chosen above. The resulting number of components, according to this criteria, is shown in the following tables.

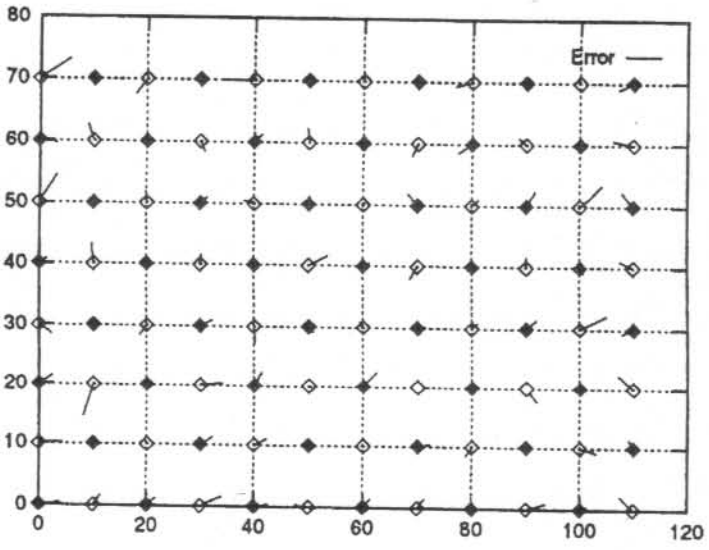


Fig. 13. Errors example in A2 with high density.

Table 3
A2 test results at random points[a]

Color components	High density		Mid density		Low density	
	Err.	Dev.	Err.	Dev.	Err.	Dev.
STD3BP	4.36	0.29	8.58	5.81	13.87	6.38
BIONET	2.65	1.32	4.16	2.42	6.85	1.61

[a]Localization errors in centimeter.

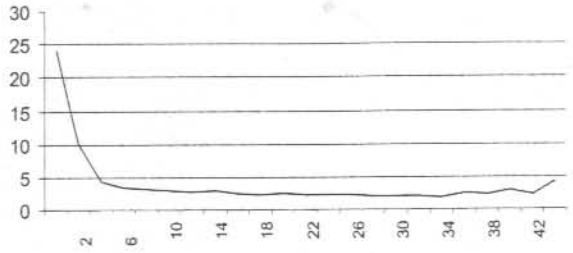


Fig. 15. MLR: mean error (cm) vs number of sectors in high density with RGB data.

9.3. MLR results

Two kinds of tests have been performed with MLR: separately on single chromatic channels and with R, G and B together. In Table 4, the results obtained with different densities are shown (validation image set). An example of spatial error distributions using MLR can be seen in Fig. 16.

The use of all the chromatic channels decreases the estimate error compared to the use of single chromatic data. On average, with respect to STD3BP, MLR on A2 gives a slightly better localization estimate, still comparable with BIONET.

9.4. PCR data sets choice

In the PCR, the number of principal components to use for the linear regression is a basic choice. The adopted criteria for the choice of such a subset is based on the size of the variance of the principal components. The components able to explain $l^* = 0.1$ of the variance are selected [10]. The resulting number of components is shown in the following tables.

As in the linear regression, an analysis of the mean localization error vs the principal component number has been developed on the validation image set. As can be seen in Fig. 17, the mean error remains

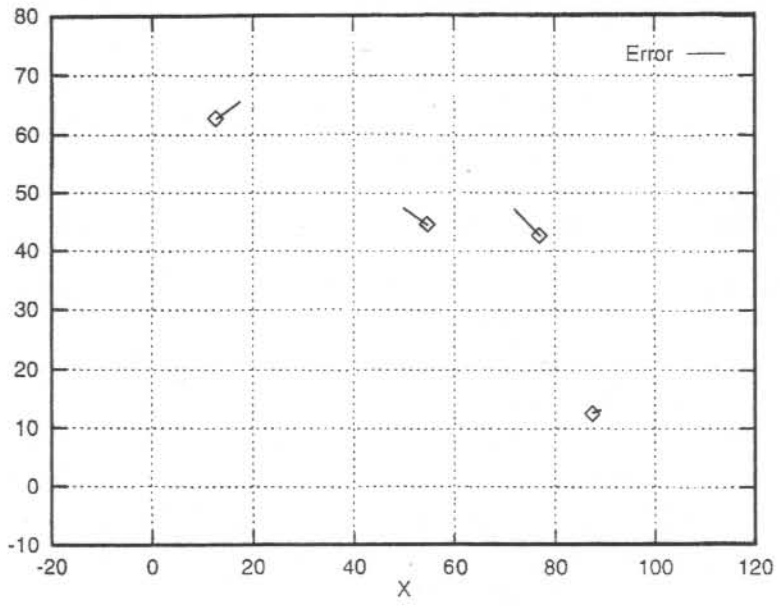


Fig. 14. Errors example at random points, in A2 with high density.

Table 4
A2 test results with MLR[a]

Color components	High density			Mid density			Low density		
	Err.	Dev.	Sect.	Err.	Dev.	Sect.	Err.	Dev.	Sect.
R	3.82	2.19	24	7.89	4.58	9	10.20	8.77	9
G	3.13	2.14	36	4.70	3.15	12	6.68	4.68	8
B	5.56	4.14	12	7.69	4.88	6	9.12	5.29	6
R+G+B	2.58	1.62	20	4.32	2.83	9	6.81	4.56	7

[a]Localization errors in centimeter.

steady in a wide range around the value selected using the suggested criteria. In the same way the images used to train the neural networks and to calibrate the MLR are now used to calibrate the models obtained using PCR and likewise, the validation image set is selected.

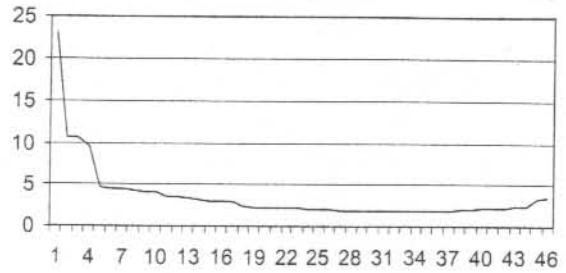


Fig. 17. PCR: mean error (cm) vs number of sectors in high density with RGB data.

9.5. PCR results

Also in this case, two kinds of tests have been performed with PCR: separately on single chromatic channels and with R, G and B together. In Table 5,

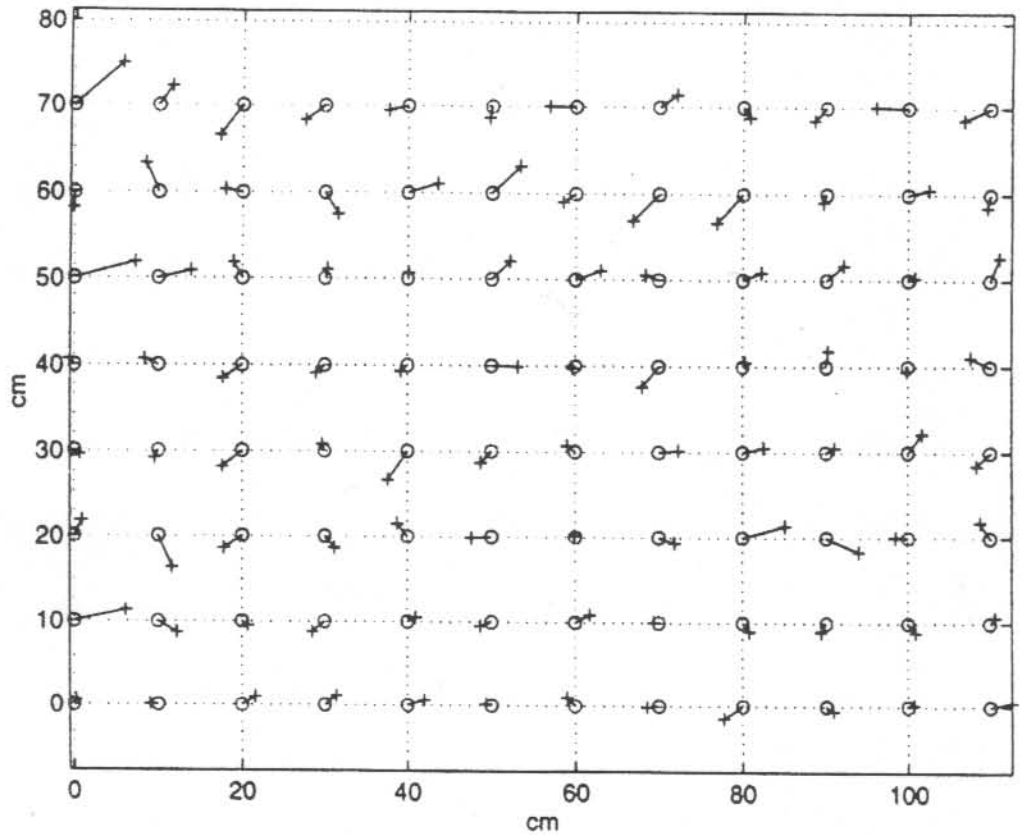


Fig. 16. Errors example in A2 with high density using MLR (cm).

Table 5
A2 test results with PCR[a]

Color components	High density			Mid density			Low density		
	Err.	Dev.	Sect.	Err.	Dev.	Sect.	Err.	Dev.	Sect.
R	3.53	2.01	24	5.66	3.40	8	8.05	4.80	7
G	2.89	2.90	22	4.35	2.84	7	8.22	5.32	4
B	4.46	2.56	23	7.67	5.00	7	11.03	6.41	8
R+G+B	2.42	1.67	29	3.66	2.37	7	6.11	3.75	5

[a]Localization errors in centimeter.

the results obtained with different chromatic input and density are shown. An example of spatial error distributions of the PCR can be seen in Fig. 18. In average PCR shows performances comparable with the other methods, but less dependent from the learning density.

Also using PCR, some visual data collected in random points have been tested in order to verify the system estimate out of the regular grid of the learning set. The results, shown in Table 6, indicate almost the same performances in and out of the sampling grid.

Table 6
A2 test results at random points using PCR[a]

Color components	High density		Mid density		Low density	
	Err.	Dev.	Err.	Dev.	Err.	Dev.
PCR	4.93	3.12	5.11	1.15	13.52	6.38

[a]Localization errors in centimeter.

10. System robustness against rotations

Pollicino does not measure the robot rotation in the fixed reference system defined in the learning phase. While a programmed turn of the robot can be easily compensated with a pattern shift, an unwanted heading change due to skidding or other external unpredictable reasons can result in an extra localization error. Tests were made to investigate the sensitivity of the system to undetected rotations. As it can be seen in Tables 7 and 8, small rotations are well tolerated. As the unwanted rotations gets higher (more than 5°), the system performance sensibly decreases, but such high rotations could be

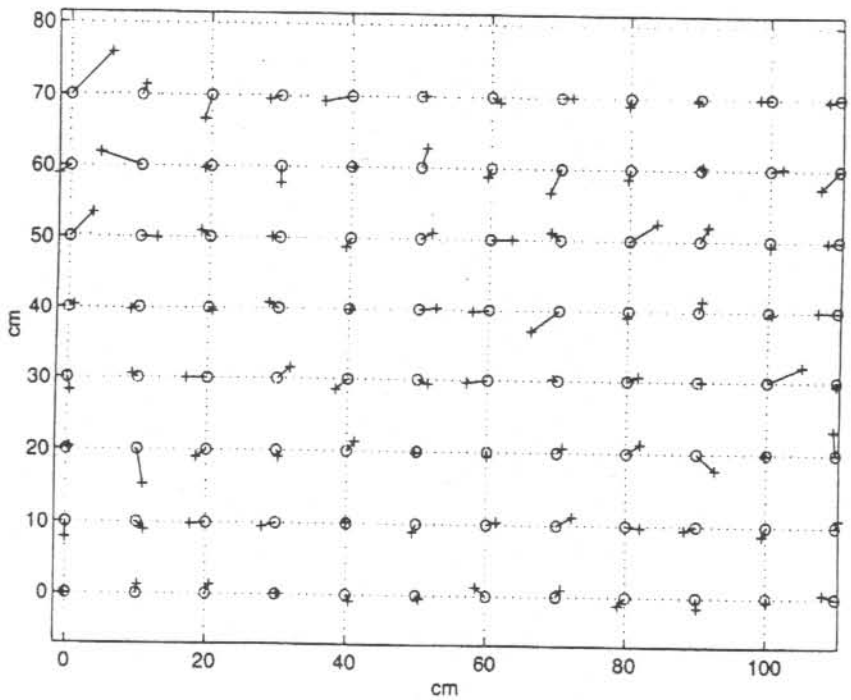


Fig. 18. Errors example in A2 with high density using PCR (cm).

Table 7
Clockwise rotation test results[a]

Clockwise rotations (deg.)	High density		Mid density		Low density	
	Err.	Dev.	Err.	Dev.	Err.	Dev.
2	4.56	1.76	8.46	4.06	12.43	1.86
5	6.73	0.50	10.84	3.76	17.08	8.81
10	14.28	4.51	20.21	1.17	24.89	5.13
15	15.24	3.30	23.90	7.47	30.67	1.31
20	17.02	5.10	29.37	1.00	27.44	1.04
45	40.32	4.48	30.45	4.17	35.23	1.99

[a]Localization errors in centimeter.

Table 8
Counterclockwise rotation test results[a]

Counter-clockwise rotations (deg.)	High density		Mid density		Low density	
	Err.	Dev.	Err.	Dev.	Err.	Dev.
2	4.30	1.78	9.11	4.27	9.92	2.29
5	6.26	0.65	11.72	3.94	15.50	12.71
10	13.05	4.26	22.03	1.09	24.87	7.19
15	13.92	3.17	26.09	8.02	31.80	1.46
20	15.52	4.79	32.11	0.90	27.93	1.06
45	36.49	4.23	33.29	4.39	37.28	2.48

[a]Localization errors in centimeter.

managed if the robot keeps track of its heading in the reference system by other means (e.g. magnetic or gyrocompass).

11. Conclusions and perspectives

A self-localization system for autonomous robot has been presented. The main features of this system are: it is a completely passive system, it does not require any environment conditioning and it works even if the learned environment changes moderately (e.g. people walking, moved objects, etc.). The precision of the localization estimate is acceptable and, especially with high density learning sets, the localization error is low, compared to the needs of a navigating mobile robot.

Pollicino has been implemented with low cost components and it does not require powerful computational resources. In fact the presented system has been implemented on low cost PC (e.g. 486) with response time almost instantaneous in relation with the sampling time of the low cost camera (about eight frames per second).

Regarding the learning subsystem choice, a comparative analysis of the results indicates that all the tested methods have comparable performances. Moreover, the randomness of the error direction suggest that an iterative estimate could further increase the overall system performance.

The use of the system in environments with heavy lighting changes, that might affect the system performance, should be investigated, and automatic learning procedure should be developed in order to realize an autonomous guidance system able to drive the robot back from unknown places visited only once.

References

[1] E. Bideaux, P. Baptiste, C. Day, D. Harwood, Mapping with a backpropagation neural network, in: Proceedings of the IMACS-IEEE/SMC Symposium, Lille, France, 1994.

[2] R. Cassinis, D. Grana, A. Rizzi, A perception system for mobile robot localization, in: Proceedings of the WAAP'95 Workshop congiunto su Apprendimento Automatico e Percezione, Ancona, Italy, 1995.

[3] R. Cassinis, D. Grana, A. Rizzi, Self localization using an omnidirectional image sensor, in: Proceedings of the SIRS'96, Fourth International Symposium on Intelligent Robotic Systems, Lisbon, Portugal, 1996.

[4] R. Cassinis, D. Grana, A. Rizzi, Using colour information in an omnidirectional perception system for autonomous robot localization, in: Proceedings of the EUROBOT'96, First Euromicro Workshop on Advanced Mobile Robots, Kaiserslautern, Germany, 1996.

[5] R. Cassinis, D. Grana, A. Rizzi, V. Rosati, Robustness characteristics of Pollicino system for autonomous robot self-localization, in: Proceedings of the EUROBOT'97, Second Euromicro International Workshop on Advanced Mobile Robots, Brescia, Italy, 1997.

[6] B. Crespi, C. Furlanello, L. Stringa, Memory based navigation, in: Proceedings of the IJCAI Workshop on Robotics and Vision, Chambéry, France, 1993, p. 1654.

[7] W.R. Dillon, M. Goldstein, Multivariate Analysis: Methods and applications, Wiley, New York, 1984.

[8] D.S. Falk, D.R. Brill, D.G. Stark, Seeing the Light, Harper and Row, New York, 1986.

[9] M.O. Franz, B. Scholkopf, H.A. Mallot, H.H. Bulthoff, Learning view graphs for robot navigation, Autonomous Robots 5 (1) (1998) 111–125.

[10] I.T. Jollife, Principal Component Analysis, Springer, New York, 1986.

[11] S.K. Nayar, K. Ikeuchi, T. Kanade, Surface reflection: physical and geometrical perspectives, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (1991).

[12] T. Ohno, A. Ohya, S. Yuta, Autonomous navigation for mobile robots referring pre-recorded image sequence, in: Proceedings of the IEEE/RSJ IROS'96, 1996.
[13] P.G. Hoel, Introduction to Mathematical Statistics, Wiley, New York, 1947.
[14] D.E. Rumelhart, J.L. McClelland, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, Cambridge, MA, 1986.
[15] S. Shah, J.K. Aggarwal, Mobile robot navigation and scene modeling using stereo fish-eye lens system, Machine Vision Applications 10 (4) (1997) 159–173.
[16] G. Wyszecki, W.S. Stiles, Color Science, Wiley, New York, 1982.
[17] Y. Yagi, S. Fujimura, M. Yashida, Route representation for mobile robot navigation by omnidirectional route panorama Fourier transformation, in: Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, 1998.
[18] Y. Yagi, Y. Nishizawa, M. Yashida, Map based navigation of the mobile robot using omnidirectional image sensor copis, in: Proceedings of the IEEE International Conference on Robotics and Automation, Nice, France, 1992.
[19] Y. Yagi, Y. Nishizawa, M. Yashida, Map based navigation for a mobile robot with omnidirectional image sensor copis, IEEE Transactions on Robotics and Automation 11 (5) (1995) 634–648.
[20] Y. Yagi, H. Okumura, M. Yashida, Multiple visual sensing system for mobile robot, in: Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, CA, Vol. 2, 1994, pp. 1679– 1684.
[21] Y. Yagi, M. Yachida, Real-time generation of environmental map and obstacle avoidance using omnidirectional image sensor with conic mirror, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1991.
[22] K. Yamazawa, Y. Yagi, M. Yachida, Omnidirectional imaging with hyperboloidal projection, in: Proceedings of the IEEE/RSJ IROS'93, Vol. 2, 1993, pp. 1029–1034.

Alessandro Rizzi graduated in Information Science from the University of Milan in 1992 and got his Ph.D. in Information Engineering from the University of Brescia (Italy). From 1992 to 1994 he worked as a contract researcher at the University of Milan in the Image Laboratory, studying the problem of color constancy and its application in computer graphics. From 1994 to 1996 he worked as a contract professor of Information Systems at the University of Brescia. Since 1994 he has been working with Professor Riccardo Cassinis at the Advanced Robotics Laboratory studying problems of self-localization and navigation using color information, omnidirectional devices and biologically inspired models. In 1999 he worked as a contract professor of Computer Graphics at Politecnico of Milano. He is presently researching on color appearance models applied to robotic visual navigation and image correction.

Riccardo Cassinis received his degree in Electronic Engineering in 1977 from the Polytechnic University of Milan. In 1987 he was appointed as an Associate Professor of Robotics and of Numerical Systems Design at the University of Udine. Since 1991 he is working as an Associate Professor of Computer Science and of Robotics at the University of Brescia. He has been working as a Director of the Robotics Laboratory of the Department of Electronics in Milan, of the Robotics Laboratory of the University of Udine, and is now the Director of the Advanced Robotics Laboratory of the University of Brescia. Since 1975 he has been working on several topics related to industrial robots, and since 1985 he is involved in navigation and sensing problems for advanced mobile robots. His current interests include mobile robots for exploring unknown environment, with particular attention to problems related to humanitarian de-mining.

Robotics and Autonomous Systems
Vol. 34, No. 1, 31 January 2001

Abstracted/indexed in: Cambridge Scientific Abstracts, CompuMath Citation Index, Compuscience, Computer Abstracts, EIC Intelligence, Engineering Index/Compendex, INSPEC Information Services, ISI Alerting Services, Science Citation Index – Expanded

Contents

0921-8890(20010131)34:1;1-5