

Self-positioning and localization of a mobile robot using vision-based behaviors

François Tièche, Narinder Singh Ghai and Heinz Hügli

Institute of Microtechnology, University of Neuchâtel
Rue A.-L. Breguet 2, CH-2000 Neuchâtel, Switzerland
Phone: +41 38 23 34 58, Fax: +41 38 23 34 01
E-mail: tieche@imt.unine.ch

This paper presents a method to localize a mobile robot in a topological map. This work enters a navigation method developed for mobile robots, which basic idea is to represent the robot spatial knowledge by a topological map and to use the behavioral approach to control the robot moves between fixed positions of the environment known as self-positioning sites. A feature of this method is that each physically grounded site is represented by a node in the topological map, and the localization problem is to find the node-site correspondences. In the paper we develop and analyze a localization method based on vision that compares images taken by a robot-mounted camera with references images. Special considerations is given to noise rejection by making the method robust against environmental changes. The localization method is implemented on a Nomad 200 mobile robot. Presented localization results illustrate its performance and degree of robustness.

I. INTRODUCTION

The behavioral approach to design autonomous systems like mobile robots is based on the existence of individual behaviors and on the strong robot-environment interactions they provide. It is inspired to some extent by the animal world, where a behavior may be described as an independent stereotyped action that is maintained by a specific stimulus [5]. For example, the behaviors perform, continuously and concurrently, simple tasks such as: "follow a wall", "go towards", "follow a corridor". The success of the behavioral approach comes from the strong interaction of the robot with the world provided by the set of behaviors. It allows the robot to move around surely in a complex and dynamic environment. More complex reactive tasks can also be achieved easily by combining behaviors, as was shown for example by a task that tidies up chairs in a room [13].

Navigation tasks cannot be achieved only with behaviors, a representation of the environment becomes necessary. A solution to this problem is provided by combining a world representation with a special kind of behavior known as a self-positioning behavior. These behaviors control the robot by servoing its moves to low-level primitives, which may be visual primitives such as points and segments extracted from image sequences that relate to invariant features and structures of the environment such as corners, angles, ceiling structures [4]. Self-positioning sites are created at discrete locations of the environment.

A proposition for the body of a world representation is the *cognitive map* [2, 7], which may be analyzed in two ways. From the topological point of view (or *reasoning level*), the cognitive map is centered on a symbolic representation. From the robot resources point

of view (or *control level*), the cognitive map is grounded in the interactions of robot sensors and actuators with the environment.

At the reasoning level, areas where the self-positioning behaviors are stimulated are symbolized by nodes, while all reactive behaviors, that move the robot between sites (i.e. follow-wall behavior, go-towards behavior, follow-way behavior), are represented by edges. Nodes and edges form a network that describe the spatial knowledge of the robot about its environment. This topological map can be used to plan robot actions and distinguish ambiguous sites. It can also be learnt by exploration of the environment.

Given physically grounded self-positioning sites and a topological map with nodes, the localization problem consists in finding the valid site-node correspondences. The problem occurs each time the robot is lost or its estimated position error is too large.

Generally, localization and positioning occur when a robot navigates in a mapped environment. Localization consists in finding a environment-map correspondence while positioning consists in adjusting the position and orientation of the robot with respect to the map. Positioning is used in geometrical maps, and is generally performed with extended Kalman filtering [3] or with correlation techniques [11]. Localization is related to topological maps. Mataric uses a sonar and a compass to represent landmarks (doors, wall, etc.) and localizes the robot while it is moving by recognizing a sequence of landmarks [9]. Kurz represents nodes by similar sonar scans, and localizes the robot by finding the node whose sonar scan is closer to the current scan [8]. Kortenkamp defines gateways with a sonar sensor signature and uses a visual information (vertical edges) to identify gateways. This localization is done without moving the robot [6]. For hybrid maps combining geometrical and topological representation, Basri proposes a method for both localization and positioning [1]. Localization is performed first by matching 2D images, and the positioning is done by finding a geometrical transformation between the current image and reference images.

The work presented here differs from previous work by the fact that instead of a positioning, a self-positioning is performed. It adjust the position and orientation of the robot with respect to the environment in a quite accurate way. The localization method can take advantage of this.

The localization method we develop and analyze in this paper is based on vision. It compares images taken by a camera mounted on top of the robot with references images. Each node is characterized by a reference image and an unknown site is localized by comparing with the

set of reference images the image taken while the robot is self-positioned. Special considerations is given to noise rejection by making the method robust against environmental changes.

In the following sections, we first describe the robot architectural principles, the experimental development environment MANO and give a short description of the Nomad 200 mobile robot hardware. A self-positioning behavior named "homing on corner" is explained in section V. Section VI presents the principles of the localization which implementation and test is discussed in section VII. Section VIII concludes this paper.

II. ARCHITECTURE

The robot architecture follows the principles of the behavioral approach. It is composed of three layers : sensor-motor, behavioral, and sequencing (Fig. 1). The layers operate synchronously with respect to each other. The lowest one, called sensori-motor layer, is based on control theory and on signal processing. It is responsible for the elementary movements of the robot and processes data acquired by the sensors. The second is the behavioral layer. It is composed of a set of behaviors that on one hand control the robot with respect to environmental characteristics , and on the other hand extract measures of the world in order to feed the robot internal world representation: the topological map. On top, the sequencing layer implements tasks which are described as sequences of behaviors. Its heart is animated by a state automaton which receives as its input the status of the behaviors and information from the map, and which activates elementary behaviors.

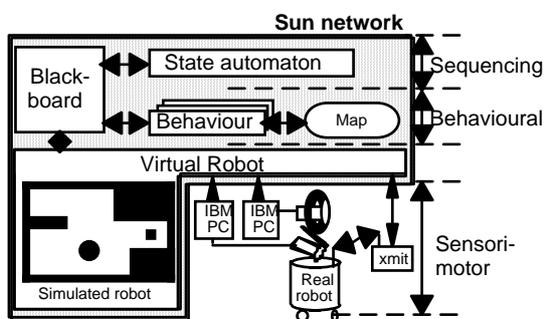


Figure 1: Development environment MANO.

III. DEVELOPMENT ENVIRONMENT MANO

MANO is an experimental mobile robot development environment [5, 13] (Fig. 1). It implements above described architecture. The core of this environment is a virtual robot unit and of a blackboard which handles the communication between the different layers. The three layers of the architecture are connected to these central elements. The sensori-motor layer is implemented on dedicated hardware located in the robot itself and on additional external units. The two other layers together with the blackboard and the virtual unit are distributed over a network of SUN workstations.

IV. ROBOT HARDWARE

The virtual robot unit links to the blackboard. It offers an interface with equivalent access to either the real or the simulated robot. The transition from real robot to simulated robot is possible at any time by a simple switch. In addition to the simulator, the virtual robot interface provides extended capabilities to monitor the robot, sensor data, commands, position etc.

The blackboard is the communication channel between the virtual robot, the behavioral layer and the sequencing layer. It acts as a server, using a TCP/IP connection protocol. Clients can connect from any point of the network.

The robot Nomad 200 — from Nomadic Technologies [12] — is a one-meter-tall cylindrical robot moved by a three wheel synchronous drive motion system; its upper body - the turret - can be rotated independently around its vertical axis. The three-dimensional robot control space is described by $(v, \omega, \varpi)^T$, where v is the translational velocity in the heading direction, ω the angular velocity of the robot frame and ϖ the angular velocity of the turret. At constant velocities, the robot moves on a circle which radius is equal to v/ω .

The Nomad200 provides sensors of different types: 16 sonars, 16 infrared range-sensors and 20 tactile sensors, and the Sensus500 sensor. Of special interest here is the latter which is a structured vision system that determines range by triangulation. It combines a laser diode used as a light source which produces a horizontal 'plane of light' with a CCD camera for image generation. Any object in front of the robot which intersects the plane of light, forms a light stripe on the camera image, which position determines its range. This device is central to the self-positioning used.

Among several other sensors mounted on the robot, the localization uses a gray scale camera linked to a Matrox image processing system.

The communication between the mobile robot and the fixed computers is established via a serial radio link.

The sensori-motor layer is implemented on a number of PC-boards: the servo loops controlling the robot are on board while some vision processing is currently performed remotely

V. SELF-POSITIONING BEHAVIOR: HOMING ON CORNERS

The localization uses a self-positioning behavior called homing on corners that serves the robot to a fixed position defined with respect to a geometrical configuration of the environment: a corner. The self-positioning site - or homing site- is defined by a position on the bisectrice of the corner, at a fixed distance from it. The behavior receives range profiles from the Sensus500 and controls the robot moves to the homing site (Fig. 2) shows the robot performing a homing on a corner. The trace of the Sensus500 laser is visible on the walls.



Figure 2: Self-positioning the robot by homing on a corner

Range data from the Sensus500 sensor are first segmented into straight segments which are submitted to a grouping process that labels joining segments as corners. In presence of a corner, the behaviour is said stimulated, and the control begins. The figure 3 indicates the circular path of the robot reaching the homing site. Radius R of the circular path and homing site position are determined. The translation velocity v is chosen to decrease with the remaining path length. Since R and v are known, the angular velocity of the robot frame is obtained by $\omega=v/R$.

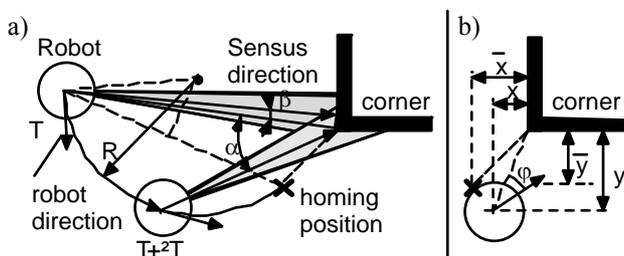


Figure 3: a) Robot path during the self-positioning behaviour b) Positioning error

In addition to moving the robot, the problem is also to control the turret angular velocity so that it always looks at the corner (and keeps the corner in the active range of the sensor). As shown in figure 3, at time T the turret is not aligned on the corner. An angular correction of β must be done. Furthermore the robot moves affect the view angle that must be corrected by an angle α . In all the robot position is evaluated at periodic time intervals ΔT , α and β are computed and the turret angular velocity is obtained by:

$$\omega = k \text{ Error!}$$

As soon as the robot reaches the homing site, the behavior is said satisfied.

Unless other positioning methods, this self-positioning behavior "homing on corners" is characterized by a good accuracy. Experiments

performed with Nomadic 200 result in mean positioning errors in position and angle as defined in figure 2 of

$$\sigma = (2.4 \text{ cm}, 2.2 \text{ cm})^T$$

$$\sigma_\varphi = 1.3^\circ$$

VI. LOCALIZATION

The proposed localization method compares images taken by a camera mounted on top of the robot with reference images. Each node is characterized by a reference image and an unknown site is localized by comparing the image, taken while the robot is self-positioned, with the set of reference images. The richness of the information of a gray scale image allows, in most of the cases, to characterize a self-positioning site in a unique way.

The localization is composed of a learning phase and a recognition phase. The learning consists in characterizing each node i by a model m_i , composed of a node identifier h_i and a reference image r_i .

During recognition, image I taken from an unknown site, is compared with the set of images r_i . The result of the localisation is the identifier h^* of the node whose comparison with I produces the highest score.

From the measures of similarity $c(I, r_i)$ issues from the comparisons, score s_i are defined as follows:

$$s_i = \begin{cases} c(I, r_i) & \text{if } c(I, r_i) > T, \\ \text{NIL} & \text{otherwise} \end{cases}$$

The score is high when the image I is close to a reference image and low otherwise. In addition, in order to reject images with similarities below a minimum value the respective scores are set to NIL.

In practice T is chosen equal to 60%.

The highest score wins, and there is a possible rejection in case all scores are NIL:

$$h^* =$$

Error!

As a measure of similarity between two images, the correlation coefficient is used. It is preferred to the classical correlation because of its invariance to changes in offset and scale of the intensity image. It is thus more robust to changes of the lightning conditions.

Two localisation methods, differing by the way they compute the scores, are considered. The first one (named F) compares entire image from test and model whereas the second one (named F) compares multiple subimages of them.

The accuracy of the self-positioning behavior is an interesting property for the localization because images taken from a same site will be very close. Image shifts of the recorder image are very limited, a property which implies that image shifts in horizontal and vertical directions $D_x \times D_y$ in the computation of correlation can be kept low.

A) COMPARING FULL IMAGES (F)

The model m_i is composed of a node identifier h_i , and a reference image which size is $N \times M$. The correlation coefficient is simply computed for each reference image r_i .

$$c(I, r_i) = \rho_{I, r_i}$$

B) COMPARING SUB-IMAGES (S)

With this second localization method, several sub-images are selected in the full reference image. For each reference, the comparison is done between the set of sub-images and the corresponding parts of the unknown image.

In this case the model m_i is composed of a node identifier h_i , n_i sub-images t_{ij} , called templates, as well as their positions (x_{ij}, y_{ij}) in the reference image. The templates, that size is $K \times K$, are chosen manually.

The correlation between an image I and a template t_{ij} gives:

$$\rho_{I,t_{ij}} = c(I, t_{ij})$$

The correlation delivers for each template a correlation coefficient $\rho_{I,t_{ij}}$ and the position $(\tilde{x}_{ij}, \tilde{y}_{ij})$ of the best matching. Again, a score is defined for the comparison of template j of image i with the corresponding sub-image of I by

$$s_{ij} =$$

Error!

which, as above, sets the score equal to similarity when it is high. Also, in order to reject images with similarities below a minimum value T , the respective scores are set to NIL. There is now as an additional rejection based on the position of the correlation peak. It must be very close to the learned template position, otherwise, the score is NIL.

From the individual scores s_{ij} , the global score is now computed as the mean of n_i non-NIL scores from s_{ij} , $j=1..n_i$. Formally

$$s_i = \text{Error!}$$

ROBUSTNESS TO CHANGES

In the second localization method, several templates are selected as templates, instead of the full image. It is expected to offer a better tolerance to environment modifications because only the correlation coefficients of the sub-images concerned by the modification will be affected, while they will be unchanged in other.

COSTS OF LOCALIZATION METHODS

Decreasing the size of the templates will increase the processing speed. Let us consider the computational cost of the correlation operation. In the following, the distance thresholds D_x and D_y are supposed equal and are named D . The cost of method F which correlates two full images of size $N \times M$, is.

$$\text{Cost}(F) \approx NMD^2$$

The proposed method for localization is yet implemented with a correlation operation moving the template over all the image I . The recognition cost for one reference image composed of n templates which size is K^2 becomes:

$$\text{Cost}(S) \approx nMNK^2$$

The size of D is about the same as K , so as soon as n increases, this method is worse than method F . Another possibility is to correlate the template just around its nominal position. This will suppress the distant templates threshold filtering. The cost is reduced to:

$$\text{Cost}(S') \approx nK^2D^2$$

The gain with respect to the first method (S) is about MN/nK^2 , the ratio of the full image surface by the n templates surface. It . This gain of this method with respect to the implemented method is NM/D^2 .

VII. EXPERIMENTAL RESULTS

The experiments with a real robot compare the robustness and the performance of the two localization methods described above. In particular we focus on robustness when the lightning environment is changed and when several self-positioning behaviors are repeated.

The experimental environment is a rectangular 10 by 6 meter long room where four corners reported in the figure 4 are used as self-positioning sites. A gray scale video camera, equipped with a fish-eye lens is mounted horizontally on the top of the robot. In order to avoid the perturbation resulting from human activities, the camera is placed as high as possible. It is placed 2 meter high and points to a direction opposite to the homing corner. The images are processed by a system based on the Matrox Image SeriesTM IM-640 (real-time processor RTP) [10]. The size of the image is 320x240 pixels by 8 bits, and the templates size is 64x64 pixels by 8 bits.

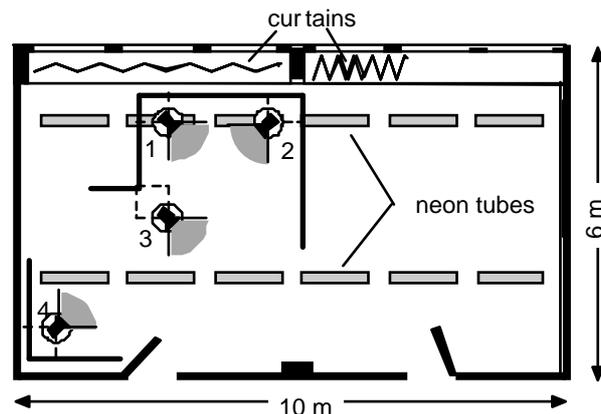


Figure 4: Experimental environment with self-positioning sites

We test the robustness of the localization by modifying the lightning of the room. Five configuration are tested, the first being the test configuration used for learning. The 4 latter are configurations with successively increasing degradations of the original lightning conditions:

- C1: all neon tubes ON and curtains closed
- C2: all neon tubes ON and curtains opened
- C3: only eastern neon tubes ON and curtains closed
- C4: only western neon tubes ON and curtains closed
- C5: all tubes OFF and curtains opened

LEARNING

The room is configured as C1, and the robot is homed once at the four self-positioning sites. Figure 5 shows the reference image corresponding to each site, the rectangles indicate the $n_i=3$ templates selected for method S. These templates are chosen manually in the upper part of the image where there is few modifications.

The templates are essentially taken on neon tubes because of the better contrast.

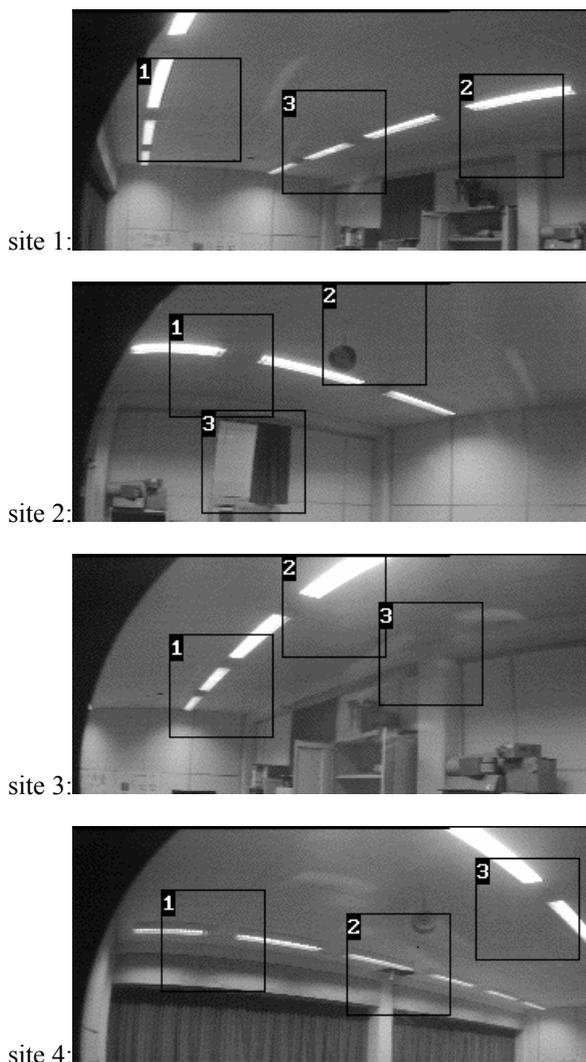


Figure 5: Images taken from each of the four self-positioning sites and the templates chosen as learning set

RECOGNITION

During the recognition test, the self-positioning behavior is repeated four times per site, and for every home the lightning is modified according to the five configurations. Methods F and S are tested. Method F comparing full images is noted F in table I. Method S is noted accordingly. For both method, the distance threshold $D=D_x=D_y$ between the learned and the best matching template is set to 32 pixels. The table I shows the results of the identification. The evaluation encompasses the test of five lightning configurations.

		h1*		h2*		h3*		h4*		h0	
		S	F	S	F	S	F	S	F	S	F
C1	h1	4	1							-	-
	h2			4	1					-	-
	h3					4	1			-	-
	h4							4	1	-	-
C2	h1	4	1							-	-

C3	h2			4	1					-	-
	h3					4	1			-	-
	h4							4	1	-	-
	h1	4	-							-	1
C4	h2			4	1					-	-
	h3					4	-			-	1
	h4							4	1	-	-
	h1	4	1							-	-
C5	h2			-	-					4	1
	h3					4	1			-	-
	h4							-	-	4	1
	h1	-	-							4	1

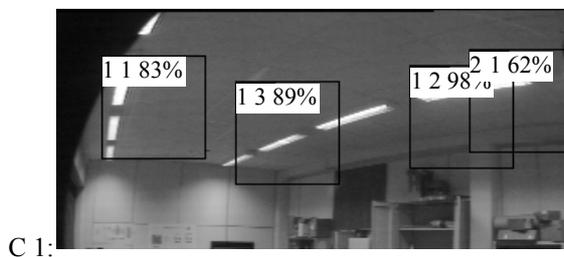
TABLE I : Recognized and rejected sites for the 4 tests and five configurations

The localization is very robust when the self-positioning behaviors is performed several times.

The performance of the localization decreases clearly when the lightening degradation increases. The localization is perfect for C1 which is the learning configuration. For C2 that enhances the illumination, results are also perfect. C3 gives good results for (S) but (F) rejects two sites. For C4 and C5, at a level of high degradation, the recognition has several rejections and (S) and (F) performances are similar.

The figure 6 shows the results of the correlation between the homing site 1 and all templates, with all configurations. The information found on the images is: rectangles show the best matching template position; numbers indicate successively the model number, the template number and the correlation coefficient. We observe that despite important lightning modifications in the images, the site is well identified in 4 cases. When all the neon are OFF (fig 6 C5), it is no wonder that an image, which templates are taken on the neon (fig 5 home 1), is not recognised.

Let us remark that the informations on figure 6 are displayed before removing distant templates. The model m21 appearing in images C1 and C3, with a coefficient of 62%, will be set to NIL by the distance filter because of $|x_1 - x_{21}| = (195,5)$ for C1 and $(196,5)$ for C3.



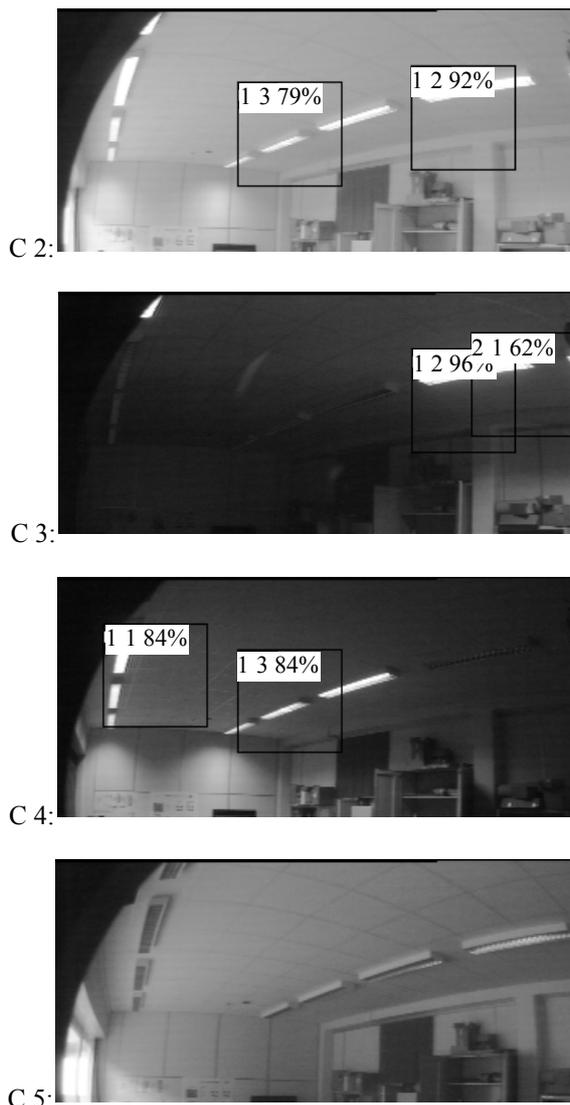


Figure 6: Results for site 1

The method using templates (S) is more robust than the method comparing full images (F).

Computational costs of current implementation refer to case 2 of section VI. Identification time for one site among four using three templates per site is 2.8 seconds. It is composed of a constant time of 0.4 second and of a correlation time per template of $t=0.2$ second. This value is acceptable in that case, but for a real map composed of a large number of sites it is too long. For example, the localisation using 20 sites with 3 templates per site needs 16 seconds. The proposed improvement refers to case 3 should decrease the computational costs by a factor $NM/D2 \approx 75$. So the localisation becomes efficient and better than method F.

VIII. CONCLUSIONS

We proposed and developed a method to localize a mobile robot in its topological map. It finds a node-site correspondence between a physically grounded self-positioning site and the nodes of the topological map. The localization method is based on the comparison of

images taken while the robot is self-positioned on its site and is adapted to be insensitive to changes of the environment. The localization uses the correlation coefficient for measuring similarity between gray scale images. We propose two localization methods, the first one comparing full images, the second one comparing multiple sub-images - or templates - extracted from the reference image.

We implemented both localization method and a self-positioning "homing on corner" behavior in the development environment MANO, that encompasses a mobile robot Nomad200 moving in an room environment, a set of different sensors, dedicated vision hardware, a collection of sensory-based behaviors as well as a versatile control unit. The self-positioning "homing on corner" behavior was performed on a real robot, showing its good accuracy. The localization method comparing templates shows better robustness and performances than the method comparing full images. This localization method gives excellent results for reasonable lightning changes of the environment.

REFERENCES

- [1] R. Basri and E. Rivlin, "Homing Using Combinations of Model Views", *Int. Joint Conference on Artificial Intelligence*, September, 1993
- [2] J.H Connel and S. Mahadevan, In: *Robot Learning*. Kluwer Academic Publishers, Boston, 1993.
- [3] J.L. Crowley, "Navigation of an intelligent mobile robot", *IEEE Journal of Robotics and Automation*, 1(1), 1985, pages 31-41
- [4] C. Facchinetti, F. Tièche and H. Hügli, "Three Vision-Based Behaviors For Self-Positioning A Mobile Robot", *Proc. of the Fourth Int. Conf. on Intelligent Autonomous Systems (IAS-4)*, Karlsruhe, Germany, March 1995, pp. 64-71.
- [5] H. Hügli, J.-P. Müller, Y. Gat, M. Rodriguez, C. Facchinetti and F. Tièche, "Architecture of an autonomous system: application to mobile robot navigation", *Proc. of the Second Symposium on Artificial Intelligence and Robotics*, EPFL-Ecublens, Switzerland, September 1994, pp. 97-110.
- [6] D. Kortenkamp and T. Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing", *Proc. of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, 1994
- [7] B.J. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on semantic hierarchy of spatial representations", *Journal of Robotics and Autonomous Systems*, 1991
- [8] A. Kurz, "Building maps on a learned classification of ultrasonic range data", *Proc. of the 1st IFAC Int. Workshop on Intelligent Autonomous System*, pages 193-198, 1993
- [9] M.J. Mataric, "Environment learning using a distributed representation", *Proc Int. Conf. on Robotics and Automation*, Cincinnati, 1990.
- [10] "The Matrox Imaging Library for the Image-Series", Revision 2.1, Matrox Electronics Systems Ltd., 1993.

- [11] H.P. Moravec and A. Elfes, "High resolution maps from wide angle sonar", *IEEE Int. Conference on Robotics and Automation*, pages 116-121, 1985
- [12] "Nomad 200 Users's Guide", Nomadic Technologies, Mountain View CA, 1992
- [13] F. Tièche, C. Facchinetti & H. Hügli, "A Behavior-Based Architecture to Control an Autonomous Mobile Robot", *Proc. of the Fourth Int. Conf. on Intelligent Autonomous Systems (IAS-4)*, Karlsruhe, March 1995