

Using Motor Representations for Topological Mapping and Navigation

Raquel Frizzera Vassallo[†], José Santos-Victor[‡], Hans Jörg Schneebeli[†]

[‡]*Instituto de Sistemas e Robótica - Instituto Superior Técnico, Lisboa, Portugal*

[†]*Universidade Federal do Espírito Santo, Vitória, ES, Brasil*
{*raquel,jasv*}@*isr.ist.utl.pt*, {*raquel,hans*}@*ele.ufes.br*

Abstract

We propose the use of motor vocabulary, that express a robot’s specific motor capabilities, for topological map-building and navigation.

First, the motor vocabulary is created automatically through an imitation behaviour where the robot learns about its own motor repertoire, by following a tutor and associating its own motion perception to motor words. The learnt motor representation is then used for building the topological map. The robot is guided through the environment and automatically captures relevant (omnidirectional) images and associates motor words to links between places in the topological map. Finally, the created map is used for navigation, by invoking sequences of motor words that represent the actions for reaching a desired goal. In addition, a reflex-type behaviour based on optical flow extracted from omnidirectional images is used to avoid lateral collisions during navigation.

The relation between motor vocabulary and imitation is stressed by the recent findings in neurophysiology, of visuomotor (mirror) neurons that may represent an internal motor representation related to the animal’s capacity of imitation.

This approach provides a natural adaptation between the robot’s motion capabilities, the environment representations (maps) and navigation processes. Encouraging results are presented and discussed.

1 Introduction

In this paper, we propose the use of motor representations, in the form of a motor vocabulary, for topological map-building and navigation. These motor representations describe the robot’s motor repertoire and are created through an imitation process. Incorporating motor representations into topological maps allows a natural adaptation between the robot’s motion capabilities, the environment representations (maps) and navigation processes.

Topological maps have proved suitable for global navigation tasks, such as “going to a distant place” in an extremely robust manner [8]. In contrast to geometric repre-

sentations, they encode global information of the environment in a qualitative manner, thus relaxing the need for an absolute and precise localization. Topological maps are often represented as graphs where nodes represent visual landmarks and links contain information about actions or directions.

In [15], landmarks are represented by images of a corridor and links are associated to forward motion or, in special cases, to turns. In [2, 10], a reactive approach is used. Both geometric and topological descriptions are created in real-time where active nodes/behaviors and links specify the robot motion, as bearing and compass information.

Recently, omnidirectional images have been used in the context of visual (topological) navigation, as the large field of view significantly improves the robustness of localization and tracking. In [16], the map is composed of omnidirectional images compressed using Principal Component Analysis. Each node in the graph corresponds to a reference position and is associated to a specific set of actions (turn right, forward, etc). In [11], omnidirectional images are used not only for localization but also for determining the steering angle and performing obstacle detection.

Considerably much less attention has been given to the problem of creating these maps. Also, in most cases, maps are associated to motion directions or sensor readings which, to a large extent, are coded *a priori*. Our approach is substantially different from the existing previous work, mainly in the way we use motor representations both for (topological) mapping and navigation.

Using motor representations in the context of mapping and navigation seems a natural process for humans when describing or exploring the environment. When giving directions towards a desired goal point, humans describe the environment’s topology as a sequence of “motor actions”, part of a “motor vocabulary” that represents the human’s motion repertoire. The advantage of associating a motor vocabulary to topological mapping/navigation is two-fold: (i) it offers the ability of describing routes in the environment (mapping) and (ii) the execution of those actions, actually corresponds to following a determined

path in the map (navigation).

Rather than programming these motor representations for a mobile robot, they can be learnt automatically through imitation. While mimicking the teacher’s actions, the robot explores its motion repertoire and associates its own actions and proprioception to *motor words*.

Thus, the created *motor vocabulary* naturally encodes the robot’s specific motion capabilities. Besides defining a human-robot interface as in [1], we propose to use the learnt vocabulary for creating topological representations of the environment. When the user guides the robot throughout the environment, the system collects pictorial information about the workspace and memorizes its own actions in terms of the *motor vocabulary*. Nodes of the topological map are represented by omnidirectional images, while links are associated to *motor words* that have a specific internal representation for each robot.

Finally, after creating the topological map, it can be used for navigation simply by retrieving the visual information and by invoking the sequence of motor commands between the initial position and the final goal.

Besides the topological navigation layer, the robot is endowed with a reflex-type behaviour in order to avoid lateral collisions with walls and objects during motion. For that purpose, the robot’s linear and angular velocities are controlled as a function of optical flow measurements in the omnidirectional images.

Interestingly, the intimate relation between imitation and internal motor representations has been stressed by recent findings in neuroscience. A special group of visuomotor neurons, designated *mirror neurons*, was found in the monkey premotor cortex (area F5) [4]. Those neurons become active both when the monkey acts on an object or when it observes another monkey or experimenter making a similar action. A similar structure seems to exist in the Brocca’s area of the human brain. These neurons probably represent a mechanism that matches observed events to internally generated actions. They may provide a way for coding a *motor vocabulary*. Words correspond to the motor repertoire and could also be the basis of the mechanism of action imitation and understanding [12].

Section 2 describes the basic imitation behaviour and how the robot learns and creates the initial motor vocabulary. Section 3 shows how to use this motor vocabulary for building topological maps and Section 4 explains how optical flow in omnidirectional images is used for defining a reactive level of navigation. Results are included in Section 5 and Section 6 presents conclusions and ideas for future work.

2 Learning a motor vocabulary by imitation

Because our robot’s motion is restricted to the ground plane, the imitation ability was implemented as a *person-*

following behavior. While following a person, the robot learns how to associate a label (*motor word*) to its own perception of the executed motion. The created *motor vocabulary* (set of labels) is defined by the user during the learning phase and may have as many labels as wanted.

2.1 Person following

Usually a person-following behaviour requires skin segmentation and involves a great computational effort [5, 14]. For simplicity, we assumed that the user carries a distinctive blue-colored rectangle. The target is first detected using the hue channel of frontal images captured by a colour camera. Noise in the resulting binary image is filtered through morphological operators and the largest remaining blob is selected. The contour is detected and the rectangle lines are estimated by a robust fitting procedure [6]. Finally, the corners coordinates are determined from the lines intersection, as shown in Figure 1.



Figure 1: Detection of the blue rectangle’s corners for the following behaviour.

A visual servoing strategy [7] was used for implementing the following behaviour. The goal consists in following the blue rectangle at a predefined distance (1m) and oriented toward its center.

The camera optical axis (Z -axis) is parallel to the ground plane (XZ -plane) and aligned with the robot’s forward direction. The motion of each corner of the rectangle in the image (x_i, y_i) can be related to the camera motion by an *image Jacobian* matrix. Since the experimental platform has only two degrees of freedom, T_z and ω_y , the expression for the *Jacobian* can be simplified. Assuming a pinhole camera model, the image motion of each corner is expressed as:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} x_i/Z_i & -1 - x_i^2 \\ y_i/Z_i & -x_i y_i \end{bmatrix} \begin{bmatrix} T_z \\ \omega_y \end{bmatrix}, \quad i = 1..4 \quad (1)$$

where Z_i denotes the depth from the rectangle’s corner to the camera projection center.

Considering the four corners, we stacked together the individual Jacobian matrices associated to each pair of co-

ordinates, yielding the following system:

$$\dot{\mathbf{P}} = \mathbf{J} \begin{bmatrix} T_z \\ \omega_y \end{bmatrix} \quad \text{where} \quad (2)$$

$$\dot{\mathbf{P}} = [x_1 \quad y_1 \quad \dots \quad x_4 \quad y_4]^T$$

$$\mathbf{J} = \begin{bmatrix} x_1/Z_1 & y_1/Z_1 & \dots & x_4/Z_4 & y_4/Z_4 \\ -1 - x_1^2 & -x_1y_1 & \dots & -1 - x_4^2 & -x_4y_4 \end{bmatrix}^T$$

The control error, ε , is defined as the difference between the rectangle's corners coordinates and the reference positions (indicated by the sub-index d):

$$\varepsilon = [x_1 - x_{1d} \quad y_1 - y_{1d} \quad \dots \quad x_4 - x_{4d} \quad y_4 - y_{4d}]^T$$

The feedback control law to determine the robot velocities needed to compensate that error, is obtained through the pseudo inverse, \mathbf{J}^+ , of the total image Jacobian [3]. The motor commands sent to the robot are given by:

$$\begin{bmatrix} T_z \\ \omega_y \end{bmatrix} = -\kappa \mathbf{J}^+ \varepsilon \quad (3)$$

where κ is a proportional gain. The image Jacobian (see Eq. (2)) depends on the Z -component of each corner point that cannot be estimated with just one image. As an alternative, we use $Z = Z_d$, where Z_d is the desired reference distance (1m). Local convergence of the control law is guaranteed in the vicinity of Z_d .

2.2 Learning a motor vocabulary

The following/imitation behaviour described previously allows the robot to learn a *motor vocabulary*. The user can teach the robot as many words as he/she wants (one word at a time). While the user executes a given motion pattern, identified by a specific *motor word*, the robot starts the following behavior and uses its proprioception of linear and angular velocities (v, w) as its internal representation of that movement. The values of (v, w) are retrieved from encoder readings.

For each motion, we use a robust estimate of the mean angular and linear velocities to describe the corresponding *motor word* and to define the separation between the various movements. Once the vocabulary is learnt, the robot can recognize movements and associate them to the corresponding *motor word*. Movement recognition is accomplished with a Bayesian classifier, using the Euclidean distance as the discriminant function.

3 Using a motor vocabulary for topological mapping and navigation

Besides defining an human-robot interface based on actions, we utilize the *motor vocabulary* for topological map building and navigation.

3.1 Mapping

For map building, the user guides the robot inside new halls, corridors and rooms exactly in the same way as for teaching the vocabulary. While following the guide, the robot builds the topological map by a process of capturing omnidirectional images to define nodes, classifying its own movements using the motor vocabulary and associating *motor words* to links in the map.

The decision of whether or not inserting a new image (node) in the topological map is taken based on a comparison with the previously stored reference image. The *sum of squared differences* (SSD) is used as a metric to assess the difference between two images in two ways. On one hand we seek for the rotation between two images that yield the minimum SDD value. On the other hand, the resulting SDD value describes the intrinsic difference between the two images (i.e. after rotation compensation). New reference images (landmarks) are stored whenever the current image differs substantially from the previous reference, i.e. when the SSD-value exceeds a predefined threshold, or the angular difference between the images is greater than 30° .

Figure 2 illustrates the orientation estimates obtained with the correlation method during a pure rotation of the robot. The first image is taken as the reference and used to compare with other captured images. The angular difference reaches 30° for the 20th image. An example of the variation of the SSD-value when the camera translates is shown in Figure 3 to illustrate the rules for updating robot position during map navigation.

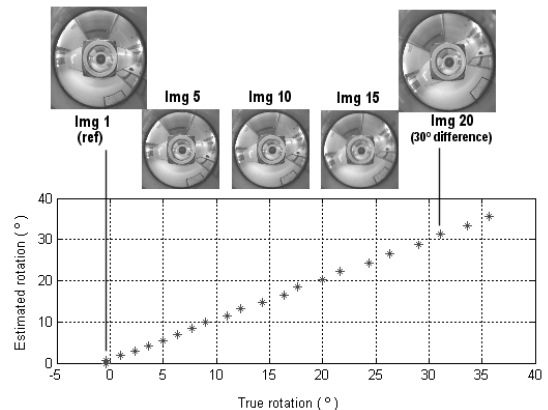


Figure 2: Increasing on the angular difference calculated by correlation for images taken during a rotation of the robot.

While images are captured and compared, the robot proprioception of motion is classified using a *motor word*. Whenever a new node/image is stored, the most frequently selected *motor word*, between the new node and the previous one is attributed to the corresponding link.

To allow for bidirectional motion, we associate to each link not only the learnt motor word, but also its dual: a left turn in one direction must be changed to a right turn when traveling in the opposite direction. Distances between references, obtained from odometry, are also memorized.

After the robot has visited the entire workspace, performing the processes just described, the topological map is completed and can be used for navigation.

3.2 Navigation

In the navigation mode, when the robot is asked to return to a specific place, it starts by localizing itself. This is accomplished by comparing the initial image with the whole map. The best match (lowest SSD) defines the starting position and the robot's orientation with respect to the map.

Once the initial localization is completed, the system determines the shortest path to the goal point. Notice that the robot does not use distance values to navigate but just to evaluate the best path. Having selected the path to follow, the sequence of *motor words* along the determined route is defined and represents the motions required to drive the robot to the goal. Before it starts moving, the robot corrects its orientation to 0° or 180° relative to the initial reference image considering the defined direction of motion.

When executing a motor command, the robot monitors its progress along the route by comparing the captured images against map reference images. Every captured image is correlated with the current node/position ($SSD[k]$) and the subsequent one ($SSD[k+1]$) in the path. The robot position is updated if $SSD[k+1]$ is consistently (6 consecutive times) lower than $SSD[k]$, or if the ratio ($SSD[k+1]/SSD[k]$) is less than 0.8. In both cases, that means the robot is closer to image $I_{[k+1]}$, and that the actual position should be updated. Whenever a new position is reached, the subsequent command is determined by the *motor word* stored in the following link. This behavior continues until the final location is reached. Figure 3 shows the variation of the SSD for images taken while navigating from nodes I_8 to I_7 of a map. The SSD-values for both reference positions are approximately the same halfway between the nodes. When the ratio drops below the chosen threshold of 0.8 (vertical line in Fig. 3), the robot's position estimate is updated.

4 Reflexive behaviour

The previous section described how the robot uses a *motor vocabulary* for topological mapping and navigation. However, this strategy for topological navigation does not allow the robot to avoid collisions with walls and objects. To overcome this problem, we included a reflex-like navigation module that controls the robot's velocity based

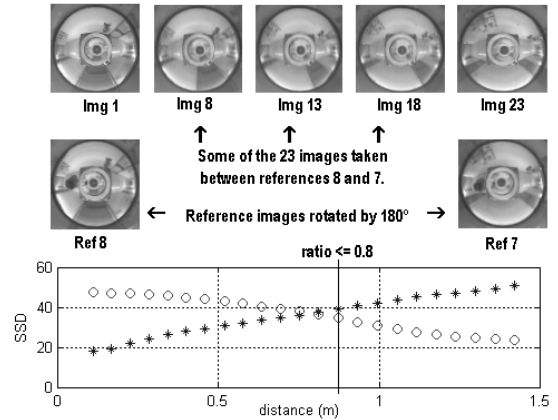


Figure 3: Evolution of the SSD for images taken between nodes 8 and 7. Correlation with images Ref-8 and Ref-7 are represented by asterisks and circles, respectively.

on optical flow measurements. The same omnidirectional images used for localization are used for optical flow estimation. With a hemispherical field of view, the number of features that can be tracked increases and either the focus of expansion (FOE) or the focus of contraction (FOC) will be visible in the image.

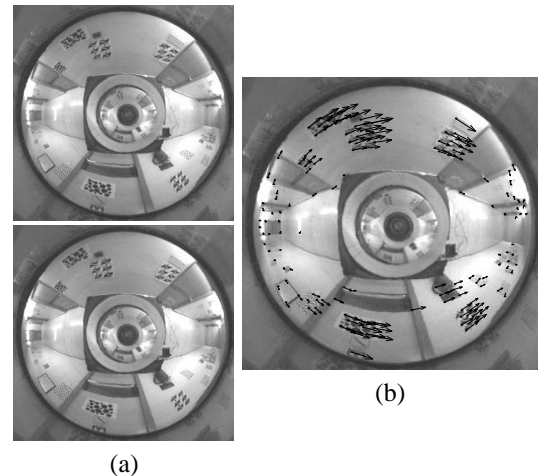


Figure 4: Two consecutive omnidirectional images (a) and the estimated optical flow during map navigation (b).

Image motion is computed using the tracker of Lucas and Kanade [13, 9]. When navigating inside a corridor, the robot compares the estimated flow on the walls in order to align itself with the center of the corridor, by adjusting its linear and angular velocities. Thus, while the robot checks/updates its current position and executes a motor command defined in the map, it keeps a reflex to maintain the robot centered in the corridor avoiding lateral collisions. Figure 4 shows results on optical flow estimation on omnidirectional images inside a corridor.

Currently, this motion-based reflex has been tested in corridor-like environments. We plan to extend its usage to less structured environments and to include this information also during the vocabulary learning phase.

5 Experiments and Results

Experiments were done with a TRC Labmate platform, equipped with a PIIMMX-350MHz on board computer. A color camera is used for the following behaviour and a B&W catadioptric omnidirectional camera is used to capture images of the environment. The omnidirectional system is mounted on the top of the robot with its axis coincident to the platform’s rotation axis (see Figure 5).

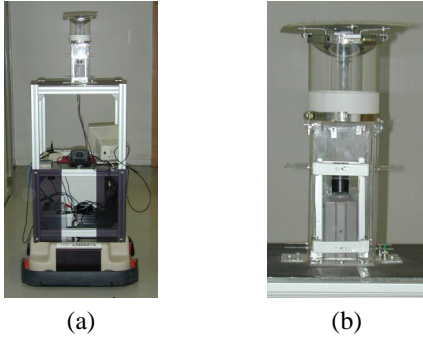


Figure 5: The robot (a) and the catadioptric system (b).

For the following behaviour, the control loop runs approximately at $1.25Hz$ when using 240×320 images for tracking the blue rectangle.

Using the person-following/imitation behaviour, a simple *motor vocabulary* was created by guiding the robot as described in Section 2. Figure 6 shows the clusters found in the (v, w) space, for each taught word/movement. Points in gray represent the selected samples while those in black correspond to outliers. Mean values are represented by asterisks and the lines separating the various clusters are also shown. The created vocabulary is detailed in Table 1.

After learning the *motor vocabulary*, the robot was again guided through the environment to create a topological map. The executed path is shown in Figure 7. The map has three branches that correspond to the floor corridor (nodes 1 to 13), the main entrance (nodes 1 and 14 to 22) and the lab entrance (nodes 1 and 23 to 29). Of course, specific nodes can also be identified by special names like Vision Lab, Room 7.22, Meeting Room, etc.

After that, some navigation experiments were conducted using the automatically built map. Two examples are shown in Figure 8. In the first one (plot-a), the robot was asked to go to node 10. Node 20 was determined as the initial position and a sequence of *motor words* was

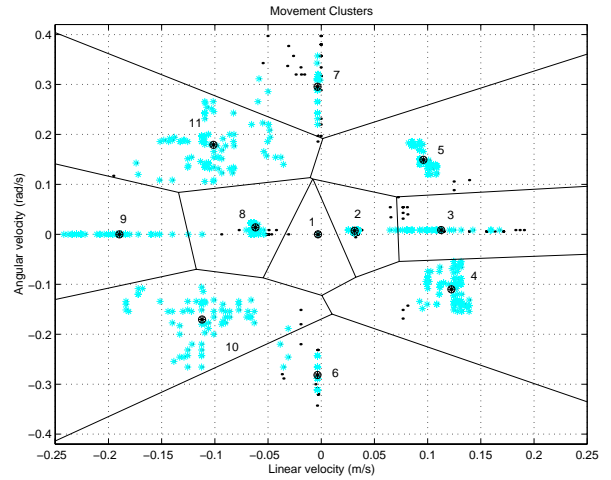


Figure 6: Clusters of the created motor vocabulary. Gray/black points correspond to inliers/outliers. Mean values and separation between the various classes are shown.

N	Motor Word	\bar{v} (m/s)	\bar{w} (rad/s)
W 1	stopped	-0.0030	0
W 2	front slow	0.0314	0.0071
W 3	front fast	0.1127	0.0086
W 4	front right	0.1228	-0.1093
W 5	front left	0.0960	0.1492
W 6	right turn	-0.0035	-0.2816
W 7	left turn	-0.0034	0.2955
W 8	back slow	-0.0619	0.0139
W 9	back fast	-0.1897	0
W10	back left	-0.1093	-0.1709
W11	back right	-0.1013	0.1792

Table 1: Motor vocabulary and mean values for linear and angular velocities of each word.

defined from the map links. The path to the goal was computed as the sequence of nodes “20 - 19 - ... - 14 - 1 - 2 - ... - 10”. Places where the robot updated its position are indicated by asterisks in the graph.

In the other experiment (plot-b), the robot was asked to go to node 27. Position 10 was the initial position. The robot started away from the center of the corridor and corrected its path while updating its position in the map and executing the motor commands. The trajectory control was accomplished by visual servoing on optical flow measurements estimated from omnidirectional images.

For some moments in both experiments, the robot’s direction of motion was opposite to that used during map construction. Thus, the sequence of *motor words* was defined by dual motions. The facility to handle motion in various directions is largely enhanced by the use of omnidirectional images.

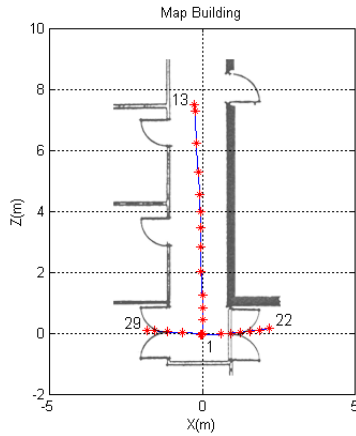


Figure 7: Robot trajectory (odometry plot) during map building. Asterisks correspond to map nodes.

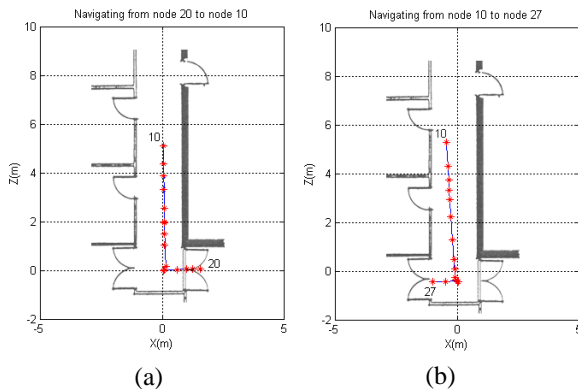


Figure 8: Robot trajectories (odometry plots) when navigating from node 20 to node 10 (a) and from node 10 to node 27 (b). Asterisks indicate position updates. Robot velocity control is made through optical flow measurements.

6 Conclusions

In this work, we used a motor vocabulary for topological map-building and navigation by a mobile robot. The vocabulary is learnt by the robot through an imitation/following behaviour where the motor words are associated to the robot's proprioception.

Associating a motor representation to topological maps seems to be a flexible way of exploring the environment considering the available motor capabilities. It allows a natural adaptation between the robot's motion repertoire, the environment representations and navigation processes.

In addition, a reflex-type behaviour was implemented to avoid lateral collisions of the robot during navigation. It is based on optical flow extracted from omnidirectional images whose large field of view improves the system robustness.

The obtained results are encouraging and, as the next step, we hope to extend the robot's proprioception, including the optical flow estimates to the learning phase for the vocabulary definition and topological mapping.

Acknowledgements

This work is partly supported by CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior via a cooperation between Instituto Superior Técnico, Lisbon - Portugal, and Universidade Federal do Espírito Santo, Vitória - Brazil.

References

- [1] A. Billard. *Imitation: a means to enhance learning of a synthetic proto-language in an autonomous robot*. In K. Dautenhahn & C. L. Nehaniv, editors, *Imitation in Animals and Artifacts*. MIT Press (In Press), 2001.
- [2] G. Dedeoglu, M. J. Mataric, and G. S. Sukhatme. Incremental, on-line topological map building with a mobile robot. *Proc. Mobile Robots XIV - SPIE*, 1999.
- [3] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. Robotics and Automation*, 8(3):313–326, June 1992.
- [4] L. Fadiga, L. Fogassi, V. Gallase, and G. Rizzolatti. Visuomotor neurons: ambiguity of the discharge or 'motor' perception? *Int. J. Psychophysiology*, 35:165–177, 2000.
- [5] S. Feyer and A. Zell. Detection, tracking and pursuit of humans with an autonomous mobile robot. *IROS*, 1999.
- [6] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM*, 24(6):381–395, 1981.
- [7] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Trans. Robotics and Automation*, 12(5):651–670, October 1996.
- [8] B. Kuipers. Modeling spatial knowledge. *Cognitive Science*, 2:129–153, 1978.
- [9] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI'81*, 1981.
- [10] M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Trans. Robotics and Automation*, 8(3):304–312, June 1992.
- [11] Y. Matsumoto, K. Ikeda, M. Inaba, and H. Inoue. Exploration and navigation in corridor environment based on omni-view sequence. In *Proc. IROS 2000*.
- [12] G. Rizzolatti and M. A. Arbib. Language within our grasp. *Trends in Neurosciences*, 21(5):188–194, 1998.
- [13] J. Shi and C. Tomasi. Good features to track. In *Proc. of CVPR94*, 1994.
- [14] H. Sidenbladh, D. Kragic, and H. I. Christensen. A person following behavior for a mobile robot. *IEEE Intl. Conf. on Robotics and Automation - ICRA*, 1999.
- [15] R. Vassallo, H. Schneebeli, and J. Santos-Victor. Visual servoing and appearance for navigation. *Robotics and Autonomous Systems*, 31(1-2):87–97, May 2000.
- [16] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor. Omni-directional vision for robot navigation. *IEEE Workshop on Omnidirectional Vision (held with CVPR)*, June 2000.