A STRATEGY FOR BUILDING TOPOLOGICAL MAPS THROUGH SCENE OBSERVATION

Roger Freitas, Mário Sarcinelli-Filho and Teodiano Bastos-Filho
Departamento de Engenharia Elétrica, Universidade Federal do Espírito Santo
Vitória - E.S., Brazil

José Santos-Victor
Instituto de Sistemas e Robótica, Instituto Superior Técnico
Lisbon, Portugal

Keywords: Learning, Topological Navigation, Incremental PCA, Affordances.

Abstract: Mobile robots remain idle during significant amounts of time in many applications, while a new task is not assigned to it. In this paper, we propose a framework to use such periods of inactivity to observe the surrounding environment and *learn* information that can be used later on during navigation. Events like someone entering or leaving a room, someone approaching a printer to pick a document up, etc., convey important information about the observed space and the role played by the objects therein. Information implicitly present in the motion patterns people describe in a certain workspace is then explored, to allow the robot to infer a "meaningful" spatial description. Such spatial representation is not driven by abstract geometrical considerations but, rather, by the role or function associated to locations or objects (affordances) and learnt by observing people's behaviour. Map building is thus bottom-up driven by the observation of human activity, and not simply a top-down oriented geometric construction.

1 INTRODUCTION

In many applications, mobile robots remain idle for significant amounts of time, while a new task is not assigned to it. Similarly, in many research labs mobile robots remain inactive during extended periods of time, while new sensorial information processing or navigation algorithms are being tested.

The motivation of this work is to use those periods of inactivity to observe the surrounding environment and *learn* information that can be used later on during navigation. For example, events like someone entering or leaving a room or approaching a printer to pick a document up, convey important information about the observed space and the role played by the objects therein.

The development of algorithms to extract useful information from the observation of such events could bring significant savings in programming, while affording the robot with an extended degree of flexibility and adaptability. In this work, we explore the information implicitly present in the motion patterns people describe in a certain workspace, to allow the robot to infer a "meaningful" spatial description. Interestingly, such spatial representation is not driven by abstract geometrical considerations but, rather, by the role or function associated to locations or objects and learnt by observing people's behaviour.

The mobile robot we use in this work combines peripheral and foveal vision. The peripheral vision is implemented by an omnidirectional camera that captures the attention stimuli to drive a standard, narrow field of view pan-tilt (perspective) camera (foveal vision).

Other research groups have used information associated to people's trajectories to help robot navigation. In (Bennewitz et al., 2002) mobile robots equipped with laser sensors are used to extract trajectories of people moving in houses and offices. The trajectories are estimated using the Expectation-Maximization (EM) algorithm and the models are used to predict human trajectories in order to improve people following. In (Bennewitz et al., 2003) the same authors propose a method for adapting the behavior of a mobile robot according to the activities of the people in its surrounding. In (Kruse and Wahl, 1998) an off-board camera-based monitoring system is proposed to help mobile robot guidance. In (Appenzeller et al., 1997) it is developed a system that builds topological maps by looking at people. Their approach is based on cooperation between *Intelligent Spaces* (Fukui et al., 2003) and robots. *Intelligent Spaces* are environments

Freitas R., Sarcinelli-Filho M., Bastos-Filho T. and Santos-Victor J. (2005).
A STRATEGY FOR BUILDING TOPOLOGICAL MAPS THROUGH SCENE OBSERVATION.
In *Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics - Robotics and Automation*, pages 168-174
DOI: 10.5220/0001171101680174

endowed with sensors like video cameras, acoustic sensors, pressure sensors, monitors and speakers that send information about the environment to a central processing system. Usually, the beings present in the environment are human beings and, in some cases, robots. From the analysis of the sensorial data, the *Intelligent Space* can supply the "users" with necessary information to accomplish some task. For example, this kind of environment is able to build maps and send them to the robots, allowing them to navigate safely. However, this approach is characterized by low scalability, i.e., if the robot is supposed to navigate in a different environment, such environment should be structured *a priori*.

Our approach to this problem is to extract the motion patterns of people from the robot's viewpoint directly, using an on-board vision system. The advantage of such approach is that the robot can *learn* from environments that are not structured for this purpose, thus giving to the learning process more flexibility and scalability. However, the robot cannot observe the entire environment at once, which is a limitation that can be overcome by using an incremental learning strategy. Such a strategy allows the robot to observe the environment from an initial position and to create a partial model representing the observed region. Then, starting from this initial model, the robot may change its position in the environment, and to keep observing it from the new position. From the new observations, the initial model could be validated, changed or enlarged.

The implementation of the incremental learning process is based on an incremental algorithm of Principal Component Analysis (PCA). An incremental algorithm that is based on (Murakami and Kumar, 1982; Hall et al., 1998; Artač et al., 2002) is here adopted. The omnidirectional images that are captured by the robot during the learning process will represent the nodes of a topological map of the environment. The incremental PCA (IPCA) algorithm allows the integration of new images (new nodes) in an online way. This incremental approach, in conjunction with the strategy of observing people's movements, will give the robot a high level of autonomy on building maps, while extracting information that allows the perception of some functionalities associated to specific regions of the environment.

Such topics are hereinafter addressed in the following way: Section 2 describes the overall learning system, and preliminary results are shown in Section 3. Section 4 describes the approach to enlarge the partial map created through observation, and in Section 5 some conclusions and discussions about possible developments are presented.

2 OVERALL LEARNING SYSTEM

Fig. 1 shows a scheme of our overall approach. The most important subsystems, which embed increasing level of cognition, are the vision, measurement and modeling subsystems.

The *Vision System* comprises peripheral and foveal visual capabilities. Peripheral vision is accomplished by an omnidirectional camera and is responsible for detecting movement. Foveal vision is accomplished by a perspective camera that is able to execute pan and tilt rotations, and is responsible for tracking moving objects.

The *Measurement System* is responsible for transforming visual information into features the robot is trying to learn, e.g., transforming 2D image information into trajectory points on the floor, referred to a common coordinate frame.

The *Modeling System* is responsible for building models that explain data from the measurement system. This system operates in two different levels of cognition, labelled geometric level and temporal level. The geometric level modeling system outputs strictly geometric models. The temporal level modeling system outputs models that incorporate concepts like temporal analysis and *appearance*. Depending on the kind of model the robot is trying to build, this system could also drive the way the vision system operates (e.g. controlling the gaze direction).

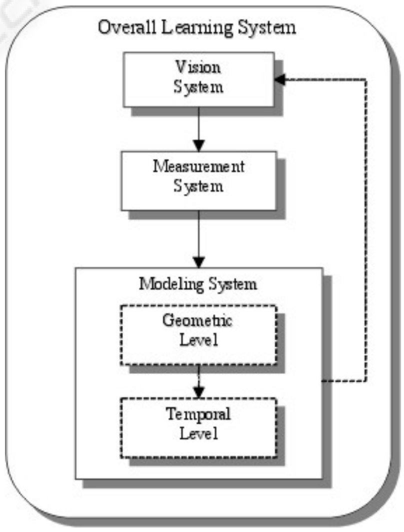


Figure 1: The Overall Learning System

In this paper, we use the scheme shown in Fig. 1 to learn possible trajectories and interesting places in the environment surrounding the robot. In this case, the Measurement System is responsible for transforming 2D image information into trajectory points on the floor. The Modeling System is responsible for building models of possible trajectories and/or finding interesting places in the environment that should be investigated in more detail (low level modeling).

We assume that the robot has no *prior* knowledge about the structure of the working environment. From any position inside it, the robot should extract useful information to navigate. In order to do that, it should be able to detect moving objects, track these objects and transform this information into possible trajectories (a set of positions in an external coordinate system) to be followed. In the following subsections, we describe in detail each one of these subsystems.

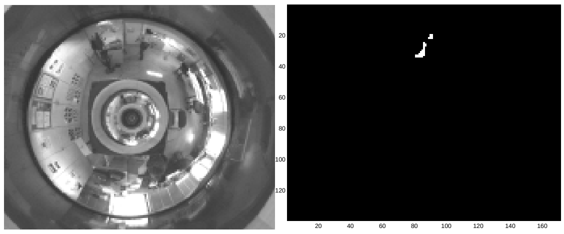
2.1 Vision System

The vision system deals with two types of visual information: peripheral and foveal (see Fig. 2). The peripheral vision uses an omnidirectional camera to detect interesting image events and to drive the attention of the foveal camera. The foveal vision system is then used to track the objects of interest, using a perspective camera with a pan-tilt platform.

2.1.1 Attention System

The attention system operates on the omnidirectional images and detects motion of objects or people in the robot vicinity. Other visual cues could be considered, but in the current implementation we deal exclusively with motion information. Motion detection can be easily performed by using background subtraction. Moving objects are detected by subtracting the current image from the background image (previously obtained). In this work, the background is modeled using the method proposed in (Gutchess et al., 2001), which uses a sequence of images taken from the same place and outputs a statistical background model describing the static parts of the scene.

Fig. 3 shows an omnidirectional image taken in the laboratory and the result of movement detection. Once the movement is detected, a command is sent



(a) (b)

Figure 3: Omnidirectional image captured (a), movement detection (b).

to the pan and tilt camera to drive its gaze direction towards the region of interest and to start tracking the moving object. To direct the camera gaze towards the detected target, we would need to determine the required camera pan and tilt angles. The camera pan angle must be set to the angular position of the target in the omnidirectional image. To determine the tilt angle, we would need to determine the distance to the detected target. Instead, for simplification we always use a reference tilt position that roughly points the camera towards the observed region.

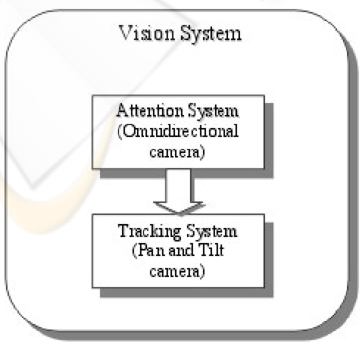


Figure 2: The Vision System

2.1.2 Tracking System

Whenever the Tracking System is activated, the Attention System is deactivated. We are currently using a simple tracking algorithm to illustrate the idea of learning about the environment from observing human actions. The next step is to improve its performance and robustness.

The current tracking routine takes two consecutive images as the input and extracts the pixels displaying some change. The result is that different regions (moving objects) in the two images are highlighted. Then, we calculate a bounding box around the detected area. The point to be tracked is the middle point of the bottom edge of the bounding box (theoretically a point on the floor).

While operating, the system is continuously detecting regions of interest in the peripheral field of view. The foveal vision system then tracks these objects, while they remain visible. If the target is not visible anymore, the Attention System is made active again. The measurement system described in the sequence will integrate the information of different tracked objects into a common coordinate system, from where more global information can be interpreted.

2.2 Measurement System

In order to estimate trajectories relative to the robot, it is necessary to estimate the distance from the robot to the moving object in each image acquired. Usually, this problem is solved using two or more cameras set in different places and applying stereo vision techniques.

As the robot is stationary while observing the environment, consecutive images of a given moving object differ only by camera rotations (pan and tilt). Thus, stereo can not be used to reconstruct the 3D trajectory of the target. The alternative used to solve this problem is to estimate the homography $\mathbf{H}$ between the floor and the image plane, i.e., to find an *a priori* plane projective transformation that transforms an image point *(u,v)* into a point on the floor *(X,Y,1)*, or

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{H} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}, \qquad (1)$$

where $\mathbf{H}$ is the 3×3 homography matrix. Initially, the homography is estimated using a set of ground plane points, whose 3D positions are known with respect to some reference frame. Then, when the foveal camera moves, the homography is updated as a function of the performed motion. So, as the camera is tracking the object, its pose is changing, and the same happens to the homography between the image plane and the floor. For this reason, we use the pan and tilt angles to update the homography (see Fig. 4).

We assume that the intrinsic parameters of the pan-tilt camera are known *a priori*, after an initial calibration step. The intrinsic parameters are used to decompose the homography matrix into a rotation matrix and a displacement vector (camera pose) relating the camera frame to a world frame. Pan and tilt angles generate canonical rotation matrices that multiply the original rotation matrix, thus updating the homography.

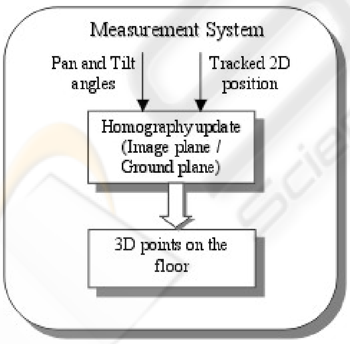


Figure 4: The Measurement System

In order to recover camera pose, we apply the methodology presented in (Gracias and Santos-Victor, 2000), which we briefly describe next. The homography, $\mathbf{H}$, can be written as

$$\mathbf{H} = \lambda \mathbf{K} \mathbf{L} \qquad (2)$$

where λ is an unknown scale factor, $\mathbf{K}$ is the camera intrinsic parameter matrix and $\mathbf{L}$ is a matrix composed from the full (3 x 3) rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$. Hence,

$$\mathbf{L} = [\ \overline{\mathbf{R}} \quad \mathbf{t}\ ], \qquad (3)$$

where $\overline{\mathbf{R}}$ is a 3x2 submatrix comprising the first two columns of matrix $\mathbf{R}$. Due to noise in the estimation process, homography $\mathbf{H}$ will not follow exactly the structure of (2). Alternatively, using the Frobenius norm to measure the distance between matrices, the problem can be formulated as

$$\lambda, \mathbf{L} = \arg \min_{\lambda, \mathbf{L}} \|\lambda \mathbf{L} - \mathbf{K}^{-1}\mathbf{H}\|^2_{frob} \qquad (4)$$

subject to $\overline{\mathbf{L}}^T \overline{\mathbf{L}} = \mathbf{I}_2$, where $\overline{\mathbf{L}}$ is a 3x2 submatrix comprising the first two columns of $\mathbf{L}$. The solution of (4) can be found through Singular Value Decomposition (SVD). Let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be the SVD of $\mathbf{K}^{-1}\mathbf{H}$. Then, $\overline{\mathbf{L}}$ is given by

$$\overline{\mathbf{L}} = \mathbf{U}\mathbf{V}^T, \qquad (5)$$

and

$$\lambda = \frac{tr(\mathbf{\Sigma})}{2}. \qquad (6)$$

The last column of $\mathbf{L}$, namely $\mathbf{t}$, can be found as

$$\mathbf{t} = \mathbf{K}^{-1}\mathbf{H} \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\lambda} \end{bmatrix}, \qquad (7)$$

thus resulting

$$\mathbf{L} = [\ \overline{\mathbf{L}} \quad \mathbf{t}\ ]. \qquad (8)$$

The last column of rotation matrix $\mathbf{R}$ can be found by computing the cross product of the the first two columns. The updated rotation matrix is given by

$$\mathbf{NewR} = \mathbf{R} \cdot \mathbf{R}_{PAN} \cdot \mathbf{R}_{TILT}. \qquad (9)$$

Finally, the updated homography is then

$$\mathbf{NewH} = \lambda \cdot \mathbf{K} \cdot \mathbf{NewL}, \qquad (10)$$

where

$$\begin{aligned} \mathbf{NewL} &= [\ \overline{\mathbf{NewR}} \quad \mathbf{Newt}\ ] \\ \mathbf{Newt} &= \mathbf{NewR} \cdot \mathbf{t}. \end{aligned}$$

We have now a way to project all tracked trajectories onto a common coordinate system associated to the ground plane. In this global coordinate system, the different trajectories described by moving objects can be further analyzed and modeled, as described in the next subsection.

2.3 Modeling System

The modeling system is responsible for building models explaining data emerging from the measurement system. Depending on the nature of the models the robot is building, this system can drive the way vision system operates. This system can operate in two different levels of cognition:

- *geometric level* - the geometric level modeling system outputs strictly geometric models, e.g., metric trajectories that could be followed by the robot;
- *temporal level* - the temporal level modeling system outputs models that incorporate a temporal analysis as well as concepts like *appearance*, e.g., images representing regions of the environment can be associated to a spacial description the robot can use to navigate (topological maps);

2.3.1 *Geometric Level*

In this work, the modeling system operates on geometric level, once it aims to interpret the observed (global) trajectories onto representations that can be used for navigation. Currently, we consider three main uses of such data:

- the observed trajectories correspond to free (obstacle free) pathways that the robot may use to move around in the environment;
- regions where trajectories start or end might correspond to some important functionality (e.g. doors, tables, tools, etc) and should be represented in a map;
- if many trajectories meet in a certain area, it means that that region must correspond to some important functionality as well.

Hence, from observation the robot can learn the location of interesting places in the scene and the most frequent ways to go from one point to another. Moving further, the robot also might be able to distinguish uncommon behaviours, what could be used in surveillance and monitoring tasks.

2.3.2 *Temporal Level*

The next step in the modeling process would be the addition of a temporal analysis of the events that occur while the robot observes the scene. Concepts like appearance are incorporated in the model as well. Appearance is often used to solve the problem of mobile robot localization based on video images (Gaspar et al., 2000). Rather than characterizing from strictly known geometric features, the approach is to rely on appearance-based methods and a temporal analysis to enrich the model of the environment. The temporal analysis will allow the characterization of pathways, as well as regions where people usually stop and stay for periods of time while engaged in some activity.

3 EXPERIMENTAL RESULTS

We performed preliminary experiments in the laboratory to verify the performance of the Vision, Measurement and Modeling systems. The robot stayed observing the laboratory while people walked by, along different trajectories. Each trajectory was performed and recorded separately. The positions on the floor, measured by the system, are shown in Fig. 5.

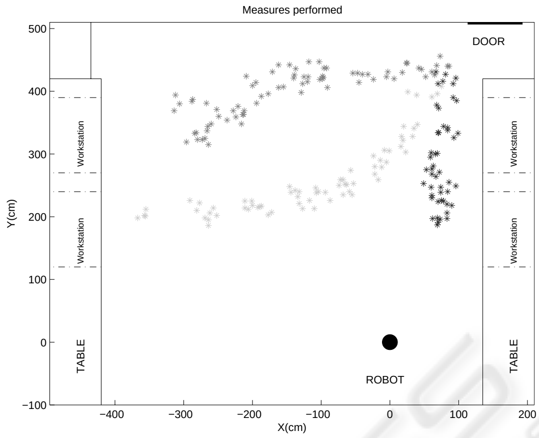


Figure 5: Real data measured from observing people's movements.

The data generated by the Measurement System is then interpreted by the Modeling System. When analyzing the data shown in Fig. 5, the most interesting point is the kind of information that can be extracted from such data. One could try, for example, to extract models of observed trajectories. In this case, the model could be obtained statistically (Bennewitz et al., 2002) or deterministically. In the deterministic case, local (e.g. splines) or global (e.g. polynomial) models could be used.

To illustrate the idea, the trajectories shown in Fig. 6 were modeled using a linear polynomial model. Places of interest can be detected as well (see Fig. 6). In this case, we applied a threshold on the data shown in Fig. 5 based on the number of times a position was visited. This is done in order to filter the data, thus discarding positions that are not frequently visited. Then, we use a k-means algorithm to cluster the remaining data.

By identifying these places, a strategy for modeling and identification can be derived, thus providing an autonomous way of learning models for such places. For example, as we can see in Fig. 6, three of such places of interest appear in front of workstations in the laboratory.

4 ENLARGING THE MAP

The experimental results obtained suggest that, from its initial position, it is unlikely that the robot can model correctly all the trajectories and interesting places in the environment. This is expected to happen

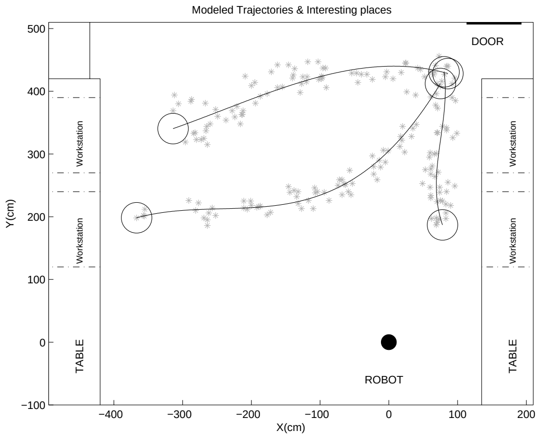


Figure 6: Examples of modeled trajectories and places of interest.

due to occlusions and the high uncertainty assigned to distant regions.

Trajectory models based on observations made from the robot's initial position are highly affected by occlusions. Besides the incorrect models, occlusions can lead to a misclassification of the regions of the environment labelled as "interesting places." For example, from the viewpoint of the robot, one can describe (or model) the region where a door is placed as a region where people usually appear and disappear. In most cases, and if occlusions are not present, such a description would suffice to correctly distinguish the object door from other "interesting places" in the environment. However, if occlusions are present, the trajectory points where they occur would be incorrectly modelled as regions corresponding to doors.

From these considerations, it can be concluded that it is necessary that the robot, based on the initial model built from its initial position, changes its position in the environment and restart the observation process, aiming to validate the current model. A strategy that allows the robot to choose the new viewpoint, given the current (and partial) metric map, should now be developed. New measurements could then be compared to the old ones through odometry readings.

Once a trajectory has been validated, the robot could start the topological mapping. The validated trajectory would be followed by the robot, while capturing images and building the map in a incremental way. Each image would be assigned to a map node, representing a position in the environment. The idea consists in representing the robot environment as a topological map, storing a (usually large) set of landmark images. To speedup the comparison of the robot views with these landmark images, it is advantageous to use low-dimensional approximations of the space spanned by the original image set. One example is to use principal component analysis (PCA) that uses the set of input images to extract an orthonormal basis (or model) of a lower dimensional subspace (eigenspace) that approximates the input images.

In the traditional approach to calculate these eigenspace models, known as *batch method*, the robot must capture all the images needed to build the map and then, using either eigenvalue decomposition of the covariance matrix or singular value decomposition of the data matrix, calculate the model. This approach has some drawbacks, however. Since the entire set of images is necessary to build the model, it is impossible to make the robot to build a map while visiting new positions. Update of the existing model is only possible from scratch, which means that original images must be kept in order to update the model, thus requiring a lot of storage capability.

To overcome these problems, some authors (Murakami and Kumar, 1982; Hall et al., 1998) proposed algorithms that build the eigenspace model incrementally (sometimes referred to as subspace tracking in the communications literature). The basic idea behind these algorithms is to start with an initial subspace (described by a set of eigenvectors and associated eigenvalues) and update the model in order to represent new acquired data. This approach allows the robot to perform simultaneous localization and map building. There is no need to build the model from scratch each time a new image is added to the map, thus making easier to deal with dynamic environments. Recently, Artač et al (Artač et al., 2002) improved Hall's algorithm (Hall et al., 1998) by suggesting a way to update the low dimensional projections of the images, thus allowing to discard the image as soon as the model has been updated. Whenever the robot acquires a new image, the first step consists in determining whether or not this image is well represented by the existing subspace model. The component of the new image that is not well represented by the current model is added to the basis as a new vector. Then, all vectors in the basis are "rotated" in order to reflect the new energy distribution in the system. The rotation is represented by a matrix of eigenvectors obtained by the eigenvalue decomposition of a special matrix (see (Freitas et al., 2003) for details).

Through this IPCA algorithm, it is possible to make the transition from geometric to appearance models. The robot will follow the metric trajectory based on odometry, while acquiring images and building the topological map of that trajectory incrementally.

5 CONCLUSIONS AND FUTURE WORK

Currently, the temporal analysis modeling level is under development, and experimental results will be available soon . A further development of the modeling system could consist of the addition of a *Functional Level*. This level would be associated with the *affordances* of the environment, perceived by the robot. According to Gibson (Gibson, 1979), "*the affordance of anything is a specific combination of the properties of its substance and its surface taken with reference to an animal*." In other words, the term affordance can be understood as the function or role, perceived by an observer, that an object plays in the environment. Such functionalities are quickly perceived through vision, and full tridimensional object models are not always required so that their functionalities in the environment could be perceived.

Even though a robot had a full tridimensional model of the environment and information about the movement of the objects, it wouldn't have a human-like scene vision. When human beings (and animals) observe a scene, they "see" several *possibilities* and *restrictions* (Sloman, 1989), such as possibilities of acquisition of more information through a change in the viewpoint and possibilities of reaching a goal through interaction with objects present in the environment. Hence, Gibson's affordances are closely related to these *possibilities* and *restrictions*. Once the affordances represent a rich source of information to understand the environment, it is important to develop a strategy to identify and extract them from the images captured by the robot. Then, it is possible that the observation of people while executing common tasks reveal some affordances in the environment. For example, one can assign to the doors of an environment the affordance "passage." If the robot could observe people appearing and disappearing in a specific region, it would perceive that region as an access to such an environment.

While the robot is building the map or navigating based on a map previously built, it is likely that the robot faces an object or a person in its way. In order to avoid the collision, it is necessary to develop an obstacle detection algorithm and an obstacle avoidance strategy based on information that can be extracted from images. Besides, an environment inhabited by people is subject to changes in its configuration. If these changes are not detected by the robot and represented in the environment model, the map would not be a correct representation of the environment anymore. Hence, it is also necessary to develop a methodology to detect changes in the environment configuration.

REFERENCES

Appenzeller, G., Lee, J., and Hashimoto, H. (1997). Building topological maps by looking at people: An example of cooperation between intelligent spaces and robots. *Proceedings of the International Conference on Intelligent Robots and Systems (IROS 1997)*, 3:1326–1333.

Artač, M., Jogan, M., and Leonardis, A. (2002). Mobile robot localization using an incremental eigenspace model. *Proceedings of the International Conference on Robotics and Automation (ICRA 2002)*.

Bennewitz, M., Burgard, W., and Thrun, S. (2002). Using EM to learn motion behaviors of persons with mobile robots. *Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2002)*, 1:502–507.

Bennewitz, M., Burgard, W., and Thrun, S. (2003). Adapting navigation strategies using motions patterns of people. *Proceedings of the International Conference on Robotics and Automation (ICRA 2003)*, 2:2000–2005.

Freitas, R., Santos-Victor, J., Sarcinelli-Filho, M., and Bastos-Filho, T. (2003). Performance evaluation of incremental eigenspace models for mobile robot localization. In *Proc. IEEE 11th International Conference on Advanced Robotics (ICAR 2003)*, pages 417–422.

Fukui, R., Morishita, H., and Sato, T. (2003). Expression method of human locomotion records for path planning and control of human-symbiotic robot system based on spacial existence probability model of humans. *Proceedings of the International Conference on Robotics and Automation (ICRA 2003)*.

Gaspar, J., Winters, N., and Santos-Victor, J. (2000). Vision-based navigation and environmental representations with an omni-directional camera. *IEEE Transactions on Robotics and Automation*, 16(6):890–898.

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston.

Gracias, N. and Santos-Victor, J. (2000). Underwater video mosaics as visual navigation maps. *VisLab-TR 07/2000 - Computer Vision and Image Understanding*, 79(1):66–91.

Gutchess, D., Trajković, M., Cohen-Solal, E., Lyons, D., and Jain, A. K. (2001). A background model initialization algorithm for video surveillance. *International Conference on Computer Vision*, 1:733–740.

Hall, P., Marshall, D., and Martin, R. (1998). Incremental eigenanalysis for classification. *British Machine Vision Conference*, 14:286–295.

Kruse, E. and Wahl, F. (1998). Camera-based monitoring system for mobile robot guidance. *Proceedings of the International Conference on Intelligent Robots and Systems (IROS 1998)*, 2:1248–1253.

Murakami, H. and Kumar, B. (1982). Efficient calculation of primary images from a set of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(5):511–515.

Sloman, A. (1989). On designing a visual system (towards a gibsonian computation model of vision). *Journal of Experimental and Theoretical AI*, 1(4):289–337.