Learning based situation recognition by sectoring omnidirectional images for robot localisation

Jianwei Zhang, Kai Huebner and Alois Knoll
Technical Computer Science, Faculty of Technology,
University of Bielefeld, 33501 Bielefeld, Germany
Tel.: ++49-521-106-2951, Fax: ++49-521-106-6440
E-mail: zhang|khuebner|knoll@techfak.uni-bielefeld.de

Abstract

We have developed omnidirectional vision systems by combining digital colour video cameras with conical and hyperbolic mirrors and applied it in mobile robots in indoor environments. A learning based approach is introduced for localising mobile robot mainly based on the vision data without relying on landmarks. In an off-line learning step the system is trained on the compressed input data so as to classify different situations and to associate appropriate behaviours to these situations. At run time the compressed input data are used to determine the correspondence between the actual situation and the situation they were trained for. The matching controller may then directly realise the desired behaviour. The algorithms are straightforward to implement and the computational effort is much lower than with conventional vision systems. Preliminary experimental results validate the approach.

1 Introduction

The issue we address is the application of a learning system for navigating a mobile robot in an environment which the robot has been made familiar with during an initial training phase. The task of determining its position and orientation is to be accomplished mainly based on visual information, i.e. high dimensional input data. Even though humans are used to relying on maps for "outdoor navigation", they manage to find their way in environments they have seen before without being mentally aware of a map, e.g. for their own home.

Visual sensors are becoming more affordable and easier to use in robotic systems. Given the current state of technology, it should be possible to accomplish the task without a plethora of sensors that are not commonly found in living organisms (such as laser scanners, infrared detectors and large circular arrays of ultrasonic transducers). The ultimate goal is to work without symbolic signs, artificial landmarks, beacons and the like.

The need for a robust and accurate localisation method is obvious; therefore numerous approaches have been developed for navigating mobile robots in recent years. The major objective of most localisation approaches is to update and to re-calibrate the internal control with external sensor inputs. Internal sensors like wheel encoders are accurate over short distances but fail over longer paths due to sliding wheels, e.g. during orientation changes. It is therefore common to combine odometric sensors with standard cameras. The vision system can then be applied to recognise certain positions of the environment and determine the robot position and orientation by using precalibrated data. A simple technical solution are artificial landmarks, e.g. "beacons". Solutions based on this approach are robust but mostly limited to structured industrial environments and expensive.

2 Related Work

In the following two subsections, robots are briefly reviewed which use optical systems for localisation. Other approaches based on non-optical sensors, e.g. *GPS*, radio navigation and localisation with ultrasonic sensors, which are not directly relevant to the approach we propose, will not be discussed.

2.1 Navigation

When a multi-sensor system is used for navigation, the complexity of the control system grows exponentially with the number of its inputs. These inputs may be generated by individual physical sensors, or they may be drawn from *logical sensors* sharing the same physical sensing device but evaluating its output according to different

principles. One way to reduce the complexity of the input is to select the most expressive inputs with regard to the desired system output (Input Selection) [5] or by statistical analysis of the input patterns using techniques like the principal components analysis (PCA). Hancock and Thorpe [4] implemented eigenvector-based navigation of an autonomous vehicle. In their experiment, the image sequence of the vehicle motion and the corresponding steering motion of a human tutor are recorded. The collected training images are compressed with PCA. A new image without any steering information is first projected onto the computed eigenvectors. While the original image is reconstructed with the principal components, the steering parameters can also be reconstructed.

In [6] the robot task is to navigate along a trained path within a corridor. All the images along the path and the associated steering vectors are stored. Based on a fast algorithm for pattern matching, the position and orientation of the robot can be calculated from the information pre-stored in the image sequence. To minimise the computation complexity, images are stored with very coarse resolution (32×32 image pixels). Since the image bank can increase very rapidly, the approach is only applicable in small working spaces.

2.2 Localisation

Based on a monocular camera system, the robot system proposed by Dudek and Zhang [3] tried to calculate the exact robot position in a room. A camera image is taken at each training position with constant orientation. The image set is preprocessed with conventional approaches like edge detection, extraction of parallel edges, and is fed into a three-layered neural network. The interpolation error of unknown positions is very small. However, the approach is very sensitive to rotational changes of the robot.

A flexible approach to localisation is the use of an omnidirectional vision system. With such a vision system a global view of the environment can be acquired without rotating the camera. Furthermore, it is relatively simple for the localisation system to deal with new objects. Approaches employing an omnidirectional vision system can be grouped according to the method of extracting information and how the information is further processed. Yagi et. al. [7] extracted edges of objects and then generated a mathematical model of the environment. The interpolation with unknown images is performed by solving a linear equation system generated with the training image set.

The POLLICINO system by Cassinis et. al. [1] can be viewed as a extension of the system proposed by Yagi. The detected edges are classified according to their colours and combined into a colour vector. In a similar way to Dudek [3], the generated vector is used as the input of a three-layered neuronal network.

Drocourt et. al. [2] propose a system with an omnidirectional stereo vision system. It consists of a camera with a mirror that is moved relative to the robot and thus can get images at two different places. The system uses probabilistic methods to search the associated parts of the two images. The used features are edges and the colour of the areas between them.

In the work presented in [10], the feasibility of localisation of a Khepera robot in a small-scale environment has been demonstrated by using a subspace projection method.

3 Experiment Systems

The first version of our system (Setup 1) is a camera combined with a conical mirror installed on a mini-robot Khepera. The vision system consists of only two components: a subminiature camera looking "upright" and a conical mirror of polished aluminium. The complete Setup 1 is shown in Fig. 1. The test environment consists of a miniature "doll's house" of 40cm$\times$40cm in size. The walls are coated with textured wall paper and the "room" includes several pictures, windows and doors.

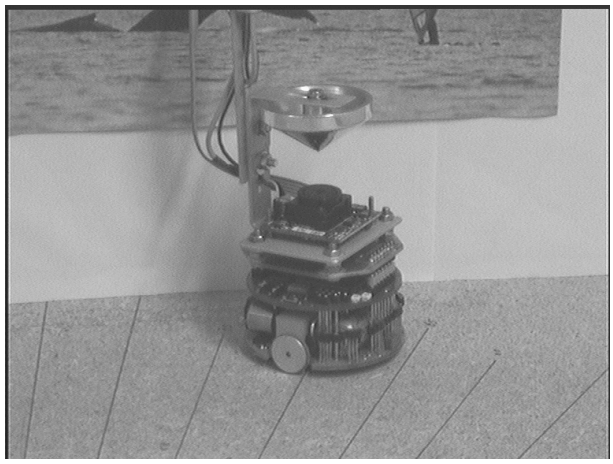


Figure 1: Setup 1: the small mobile robot mounted with an omnidirectional camera using a conical mirror.

We have developed the second version of robot vision system (Setup 2) for natural office environments. The omnidirectional vision system is mounted on the top of a pioneer mobile robot (Fig. 2), which consists of a camera facing upwards and a hyperbolic mirror above it (Fig. 3). To avoid image disturbance and to achieve a complete 360 degree omnidirectional view of the environment, the mirror is placed on a transparent plastic cylinder. The images taken with this system are used to localise the robot in an

environment that was learned beforehand. The robot is intended for use in an unmodified real-world office environment.

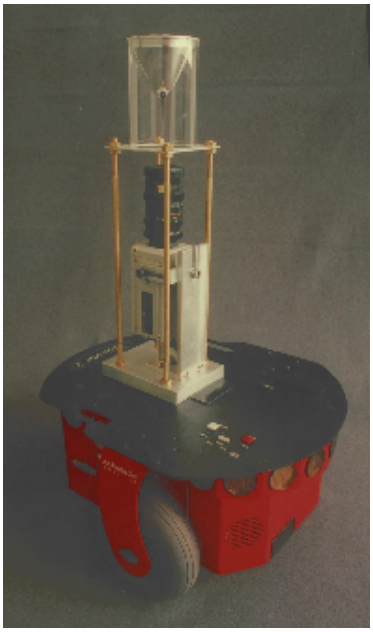
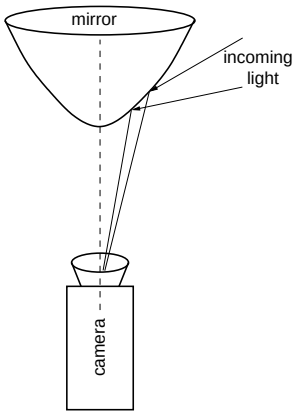


Figure 2: Setup 2: The pioneer 2 DX robot with an omnidirectional vision system using a hyperbolic mirror.



,

Figure 3: The omnidirectional vision system based on a hyperbolic mirror.

The camera records cyclically distorted images (Fig. 4) that are converted to cylindric coordinates to get a panoramic view (Fig. 5) of the environment. These panoramic images (or features extracted from them) are used to determine the robot position in an environment learned off-line.

Figure 4: The cyclic view.

Figure 5: The panoramic view.

4 Image Processing

4.1 PCA vs ORF

PCA can be used as an approach for dimension reduction to select features. With the first n dimensions of the eigenspace, the original image can be reconstructed to a pre-defined resolution. Since the magnitude of the eigenvalue corresponds to the variability of a random variable, problems may occur with input variables whose variance is low but that are nevertheless significant for controlling the process. Think of a traffic scene in which a small light that changes from green to red is much less salient than, say, the large changes in the image caused by cars passing by.

In such situations, with pure PCA applied to the input data set, a large number of eigenvectors are needed to represent control input variables in an appropriate way. A solution to this problem is to use a set of vectors that directly correlate input and output space, instead of using the eigenvectors of the input data. Features that should affect the output are called *Output Relevant Features* (ORF).

Based on a single-layer feed-forward perceptron network, the ORFs can be extracted through training with the *Hebbian learning rules*. Assume that the training data are denoted by x_j ($j = 1, \ldots, k$). If one ORF weight vector is trained which is denoted by $\vec{a}$, then the network output P is:

$$P = \sum_{j=1}^{k} a_j x_j = \vec{a}^T \vec{x} = \vec{x}^T \vec{a}. \quad (1)$$

Unlike PCA, which maximises the variance of the input data along the weight vector (eigenvector), the learning rule for the ORF weight vectors is to minimise the direct error, i.e. the difference between the desired and real values of the output. Obviously, this requires both the input x and the desired output Y_S (in our case the absolute position of the robot in a given coordinate system) to be available. Then, one element a_j of the weight vector $\vec{a}$ can be modified as follows:

$$\Delta a_j = \eta(Y_S - P)x_j \tag{2}$$

where η is the learning rate. To calculate more than one ORF weight vectors, denoted by $\vec{a_i}, (i = 1, 2, \dots)$, we use an approach similar to that proposed by Yuille et al. [8]. The computation begins with the first ORF weight vector (i=1) using (2). For calculating further $\vec{a_i}(i > 1)$, all the input data are projected onto the last ORF vectors, i.e. $\vec{a}_1, \dots, \vec{a}_{i-1}$, through which the components of the input vector, lying parallel to the ORF vector, are calculated. These components are subtracted from the input. The element a_{ij} of the vector $\vec{a_i}$ can be then adapted by:

$$\Delta a_{ij} = \eta\,(Y_S - P_i)\left(x_j - \sum_{k=1}^{i-1} P_k a_{kj}\right). \tag{3}$$

Unlike the eigenvectors the ORF weight vectors are not orthogonal. Therefore, they cannot be used for reconstructing the original data unambiguously. However, for a supervised learning system, ORFs are more efficient than principal components because they take into account the input-output relation. When modelling a complex nonlinear system, the benefit of finding the ORFs is to determine a small number of the most significant features and to isolate them through a linear transformation.

4.2 Overlap Measure

To interpolate the actual position based on some training examples, the similarity of the features should increase when the distance between the corresponding positions decreases. In other words: images taken at locations close to each other must result in similar features and the features computed based on an image from a more distant position must not be more similar.

For the development of the visual localisation system, we need to select some image pre-processing algorithms. These algorithms were to emphasise the contents of the images that are important for localisation, and to suppress those contents that are caused by position-independent changes. To select the best feature extraction algorithms, we suggest a measure of overlap.

Assume we have a set of images I_1 to I_n taken at positions p_1 to p_n. These positions lie on a straight line, so that positions p_{i-1} and p_{i+1} have the smallest distances to position p_i. We then compute the feature vectors $F_1 = f(I_1)$ to $F_n = f(I_n)$ using different algorithms. We define the distance between features as

$$d(i,j) = |F_i - F_j| \tag{4}$$

and the non-ambiguous radius in feature space:

$$r(i) = max\big(d(i,i+1), d(i,i-1)\big) \tag{5}$$

A useful feature should hold the following condition for all i:

$$r(i) < d(i,j) \quad \forall j \notin \{i-1, i, i+1\} \tag{6}$$

This means that the two nearest images I_{i+1} and I_{i-1} result in the most similar features.

To check this condition we first define the **absolute overlap** o'

$$o'(i,j) = \begin{cases} r(i) - d(i,j) & \forall \quad d(i,j) < r(i) \\ & \text{and} \quad j \notin \{i-1, i, i+1\} \\ 0 & \text{else} \end{cases} \tag{7}$$

which computes how far the feature vector of j reaches into the smallest non-ambiguous radius of i. If all $o'(i,j)$ are summed up for one position p_i, we can see whether there is an ambiguity at this position.

$$a(i) = \sum_{j=1}^{N} o'(i,j) \tag{8}$$

If $a(i) = 0$, the condition (6) is met at position p_i, otherwise there is an ambiguity. This test is suitable to automatically divide a long sequence of images into smaller sequences (situations) at positions that cause an ambiguity. Within these smaller sequences a numerical interpolator should be able to determine the position out of one single image.

To have a measure for the whole sequence of images, all $o'(i,j)$ are summed up over i and j and divided by the appropriate $r(i)$:

$$o = \sum_{i=1}^{N} \frac{1}{r(i)} \sum_{j=1}^{N} o'(i,j) \tag{9}$$

In different environments, suitable features can be selected from diverse modalities such as the complete colour and intensity images, image regions, energy-normalised edge images, HSI histograms, PCA and ORF projections, etc. For an optimal feature the **relative overlap** o should be zero and for all others it tells how unsuitable a feature is for navigation.

4.3 Sectoring

To fully utilise the global and sometimes redundant information, the viewing area of the camera can be divided into multiple sections of the same size. Theoretically, the sectoring can be achieved by arbitrarily fine resolution. In our exemplary experiment, it was found that a viewing area of 180° is sufficient in most practical cases. As an example of the Setup 1, each of sectors **A**, **B** and **C** covers an angle of 90°. All sectors are independently transformed and normalised. This way, an object in arbitrary colour will not influence the normalisation of other sectors. With the help of the sectoring technique, an unexpected new object or change of the environment at run-time can be detected and the corresponding sector can be discarded for interpolation.

For situation recognition using ORF, which will be described in section 5.2, these sectors are combined into pairs which are denoted as pseudo-segments. A pseudo-segment covers a viewing area of 180°. In the experiments, one ORF vector is computed for each combined viewing area. The projections of all three ORF vectors are associated with the robot positions.

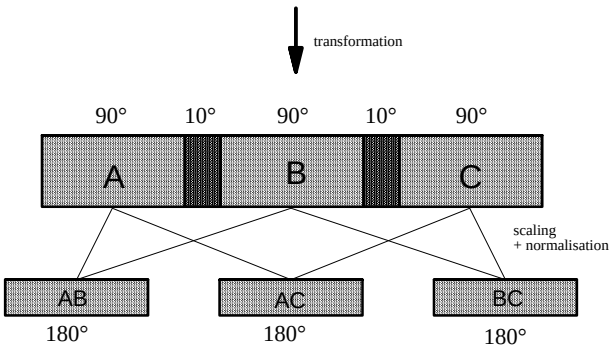


Figure 6: Sectoring by constructing pseudo-segments. Here 10 degree "blocking region" is used to avoid that small unknown objects affects two segments simultaneously.

5 Vision Based Situation Assessment

In our earlier work on robot navigation [9], we developed a "situation-based" control for input from simple infrared proximity sensors. The aim was to *differentiate between situations*: if the robot encounters many new obstacles it has to give more weight to local collision avoidance and it must temporarily reduce the weight given to goal tracking. For this purpose a "situation evaluator" was constructed by heuristic fuzzy rules.

In a situation-based model the complete robot navigation areas are coarsely classified. The whole control task is broken down into subtasks which can be performed in local "situations" so that within each situation the input patterns needed for control correlate to a certain degree. The classification criterion can be the physical neighbourhood or a set of distinctive features. If a learned situation is recognised to correspond to a known area, then, in a second step, a fine localisation can be implemented by a local controller which is specially trained for a situation, Fig. 7.

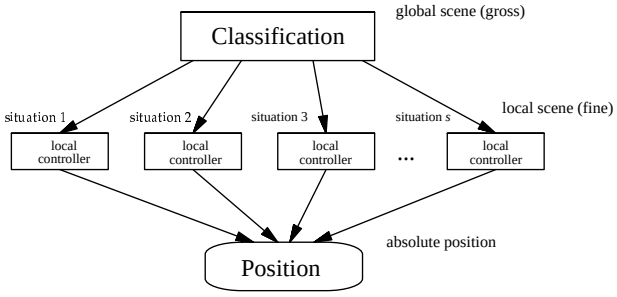


Figure 7: Dividing a global scene into local control problems.

5.1 Fundamentals of Situation Representation

In principle, if a *global eigenspace* is used to project the situation-related images, the projections of the images that fall into one situation form a specific manifold. If the dimension of the eigenspace is large enough, these manifolds are easy to separate, i.e. situations can be distinguished simply by identifying the point F in the eigenspace that the images are projected onto. Fig. 8 and 9 illustrate the process in a simplified manner. Represented this way, the *match* between a situation and a new image can simply be defined as the Euclidean distance between F and the manifold of this particular situation. To differentiate between the situations ("walls" in Fig. 8), more dimensions than shown in the figures are needed (12 in our experiment with the Setup 1).

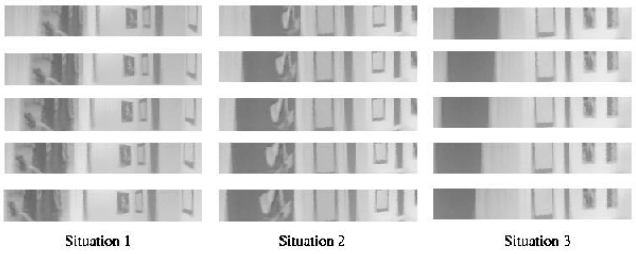


Figure 8: Views from the robot camera used in Fig. 1.

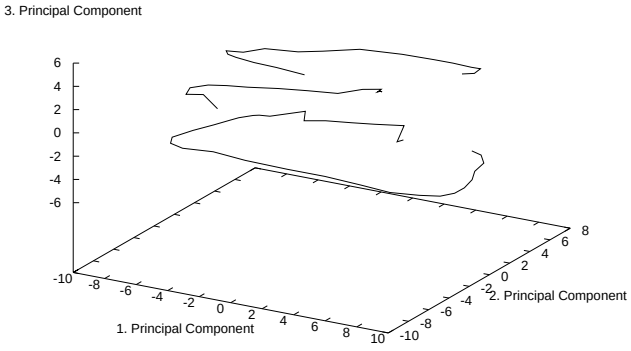


Figure 9: Situation manifolds in eigenspace. These three manifolds represent parts of the situations shown in Fig. 8.

5.2 Situation Recognition Using ORF

Although the global eigenspace provides a universal approach for representing different situations compactly, it is memory-intensive because all the eigenvectors as well as the situation manifolds must be stored. On-line projection into the global eigenspace and search in the manifolds to find the nearest neighbour are computationally expensive and hence time-consuming.

To classify situations, the variance of the projections of the pseudo-segments on their respective ORF vectors are used as follows:

- If the robot is located in a situation which it has been trained for, all the projections deliver the same variance.
- If the robot is located in other situations, all three projections differ very much.

Therefore, the situation with the smallest variance is identified as the correct one. Since ambiguity of certain degree in the grey-level image-based perception always exists, the correctness of such a situation classification is evaluated in a probabilistic sense. Further information, e.g. the energy-normalised edge images and the hue histogram, can easily be added to increase the reliability of the classification.

6 Position Learning in an Office Corridor

Besides the experiment with the Setup 1 in a small-scale environment, we also achieved some preliminary results with the Setup 2 in a real office environment. The robot's position in this environment is estimated by using image features which have been learned from a small number of training images. The learning is based on using ORFs.

As mentioned above, it is difficult to find relevant features in a couple of taken images and the search and evaluation of those features have to be directly implemented to the robots image processing algorithms, e.g. edge or corner detection. In our experiments, omnidirectional images of the environment are used as input vectors, which result in estimated position coordinates with the help of ORFs.

The images for the described tests have been taken across the corridor of our working group (Fig. 10). Unfortunately, this environment is poor in colour (grey floor, doors and white painted walls). The most information is contained in the grey-level images.

Figure 10: Testing area is the unmodified office corridor.

The first experiment includes 52 images taken at the center of the corridor with a constant distance of 10cm between two neighbouring images. Thus we tested on a line with a length of 5.10m, lying between two pairs of opposing office doors (Fig. 11). For this problem, ORF networks were trained to deliver the x-position on the line. Note that the door crossing the corridor is a glass door.

Experiments were made by splitting each dataset into training data and test data. As anticipated, the method works with neglectable error using all images for training. The main interest is to reduce the used training data without losing performance in results for not trained test data.

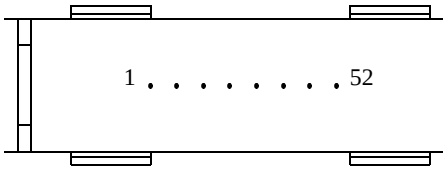


Figure 11: The training and test positions.

The database for the first experiment contained 52 images of size 612 x 384. We first took every fifth image (let $l = 5$ indicate this), which is equal to one image every 0.5m, to train the ORF vector. We started with image 6 and finished with 46 in order to additionally get some information about the method's extrapolation abilities.

Using the original iamges, the extrapolation results of images 1–5 and 47–51 were expectedly poor, but on the other hand the interpolation of unknown positions works very well with small variances. We made another test using the orignial image from the mirror and also downscaled images of size 128×96 (Fig. 12). The training images correspond to the filled dots, the test images to the unfilled triangles. Note that in this case, the given image numbers are set on the $x-$axis and the estimated ones on the $y-$axis, which means that a perfect position estimation would results in a straight diagonal line of symbols.

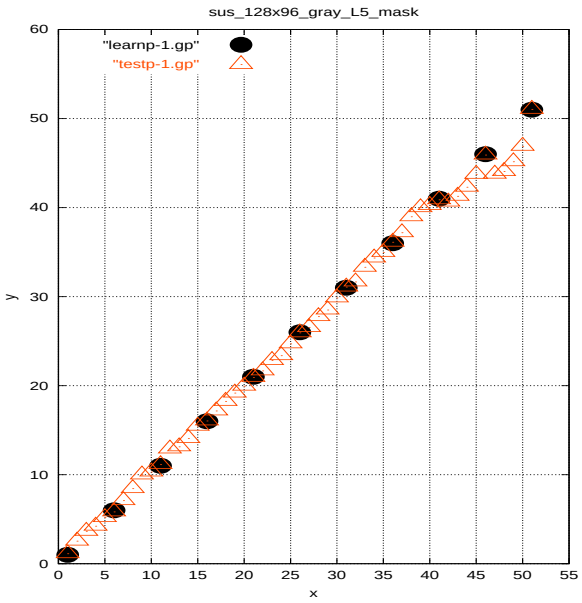


Figure 12: Test results of the robot position with respect to the image number, $l = 5$.

In spite of this loss of image information, the method still delivers good results. Thus we decreased the number of training images by setting $l = 10$, which means a distance of 1m between two training images. The results are still acceptable although the estimation errors increase.

Using the sectoring method, good estimations of robot position along the x-direction can also be achieved. Therefore, the proposed methods in section 4.3 can be applied to eliminate the problem of partial occlusions.

7 Discussions

We showed that a localisation approach based on the whole and sectors of the omnidirectional images of an arbitrary environment is feasible. The experimental environment is part of a typical living room or office, where mobile robots can find potential applications for service jobs. The further development of this approach aims at achieving the following features:

Scalability. The situation-based approach can be scaled almost arbitrarily. If the movement area is extended, new situations can be learned to cover the new area. Additionally, the computation time required and memory expenses are only linear in the number of situations.

No geometric model. No additional information of the environment is needed. Without usage of sophisticated geometric models, the direct mapping leads to a significant reduction of computational costs.

Universal method. The conventional robot vision algorithms based on segmentation, geometric feature extraction, etc. must always be adapted to specific environments. The proposed method is generally applicable to environments where geometric or color features are difficult to be found and followed robustly.

Low cost. The necessary hardware components are off-the-shelf low-cost standard products. The performance/price ratio is very good in comparison with other systems that need special hardware.

Obviously, many problems need to be solved to make the approach applicable in arbitrary environments (with too few or ambiguous objects for differentiation, large degree of unexpectness, fluctuations of the illumination, etc). The probability of the correct situation recognition and localisation can be increased by combining knowledge-based methods and fusion of redundant modules evaluating hybrid sensor information. At the moment the *Situation Classifier* is realised by physical grouping. It is desirable that in the future the learning system be

capable of *automatically dividing* a large number of sequences into appropriate situations according to the relative overlapping measure. Another issue is the size of the visual area to receive good interpolation results. Since the amount of memory needed for the local controllers is directly related to the size of the feature vectors, the input images should be as small as possible. If, by contrast, the images are too small, major distinctive features are lost. An automatic adaptation to the best size is an important objective. Furthermore, it is feasible to replace the crisp situation multiplexing with a soft-switching controller. Moreover, it is necessary to automate the learning process to make the approach simple to use.

Acknowledgment

The authors thank Torsten Scherer for discussions on using ORF and Christian Lange for implementing the overlapping measure.

References

[1] R. Cassinis, D. Grana, and A. Rizzi. An Adaptive Model for Vision-Based Localisation. In *Proceedings of EUROBOT'96*, pages 172–176, 1996.

[2] Cyril Drocourt, Laurent Delahoche, Claude Pegard, and Arnaud Clerentin. Mobile robot localization based on an omnidirectional stereoscopic vision perception system. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pages 1329–1334, Detroid, Michigan, May 1999.

[3] G. Dudek and C. Zhang. Vision-based robot localisation without explicit object models. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 76–82, 1996.

[4] J. A. Hancock and C. E. Thorpe. ELVIS: Eigenvectors for land vehicle image system. Technical Report CMU-RI-TR-94-43, The Robotics Institue, Carnegie Mellon Univeristy, 1994.

[5] J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice Hall, 1997.

[6] Y. Matsumoto, M. Inaba, and H. Inoue. Visual Navigation using View-Sequenced Route Representation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 83–94, 1996.

[7] Y. Yagi and M. Yashida. Real-time generation of environment map and obstacle avoidance using omnidirectional image sensor with conic mirror. In *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*, pages 160–165, 1991.

[8] A.L. Yuille, D.M. Kammen, and D.S. Cohen. Quadrature and the Development of Orientation Selective Cortical Cells by Hebb Rules. *Biological Cybernetics*, 1989.

[9] J. Zhang and A. Knoll. *Integrating deliberative and reactive strategies via fuzzy modular control*, chapter 15, pages 367–387. In "Fuzzy logic techniques for autonomous vehicle navigation", edited by A. Saffiotti and D. Driankov, Springer, 2000.

[10] J. Zhang, A. Knoll, and V. Schwert. Situated neuro-fuzzy control for vision-based robot localisation. *Robotics and Autonomous Systems*, 28:71–82, 1999.