

Situated Neuro-Fuzzy Control for Vision-Based Robot Localisation

Jianwei Zhang, Alois Knoll and Volkmar Schwert

Faculty of Technology, University of Bielefeld, 33501 Bielefeld, Germany

Tel: ++49-521-106-2951/2952

Fax: ++49-521-106-6440

E-mail: zhang | knoll @techfak.uni-bielefeld.de

Abstract

We introduce a neuro-fuzzy system for localising mobile robot solely based on raw vision data without relying on landmarks or artificial symbols. In an initial 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. Input data may, for example, be generated by an omnidirectional vision system obviating the need for active cameras. At run time the compressed input data are fed into different B-spline fuzzy controllers which determine the correspondence between the actual situation and the situation they were trained for. The matching controller may then directly drive the actuators to realise the desired behaviour. The system thus realises a tight coupling between a very high-dimensional input parameter space and the robot actuators. It is completely free of any internal models such as maps of the environment, the algorithms are straightforward to implement and the computational effort is much lower than with conventional vision systems. Experimental results validate the approach.

Key words: fuzzy control, supervised learning, B-splines, omnidirectional robot vision, localisation.

1 Introduction

Conventional one-step fuzzy control frequently encounters two problems when dealing with complex multi-input systems. Firstly, the dimension of the input space is too large for the control process to be tractable (“curse of dimensionality”). Secondly, the correlation between a given input and the desired output is often highly nonlinear. It is difficult to linearise the system about a certain point; the linearisation rules are in many cases hard to find. One

single controller, which interpolates between “similar” control situations, often appears inadequate to describe a global scenario with controller inputs distributed over large input range intervals.

The issue we address is the application of an adaptive neuro-fuzzy system for navigating a mobile robot in an environment which the robot – to some extent – has been made familiar with during an initial training phase. The task of determining its position and orientation is to be accomplished solely based on visual information, i.e. high dimensional input data, and without resorting to internal maps or environmental object models. The latter is particularly attractive because map and model building is not only cumbersome but it also limits the range of applications. 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. It is therefore fruitful to study the *localisation mechanisms* of animals. The questions are: How can cats, pigeons or salmons know “where they are” and find a desired place where they have been before only by having observed their environment (and possible distinctive features)? What kind of information has to be memorised? Does a general model exist for that purpose? What kind of features extracted from the visual data are most efficient? Are there features related to a specific metric, e.g. a (subconscious) floor plan? Current research indicates that hymenopterans such as bees, ants and wasps do not resort to the concept of the mental analogue of the topographic map (a metric map) for their navigation. Instead, they use a combination of landmark-based routes, snapshots and the pattern of polarised light in the sky, represented by so-called E-vectors. For further details the reader is referred to [14].

The localisation capability of animals may also be partly attributed to their extraordinary non-visual sensors for scent and magnetic fields, which are still too sophisticated to emulate. However, visual sensors are becoming more affordable and easier to use in robotic systems. We feel that 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). It should also be possible to do without symbolic signs, artificial landmarks, beacons and the like.

2 Previous Work

For navigating mobile robots, numerous approaches have been developed in recent years because the need for a robust and accurate localisation method is obvious. 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 within a short distance but fail in the long run due to sliding wheel, 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 pre-calibrated data. A simple technical solution suggests artificial landmarks, e.g. “beacons”. Solutions based on that approach are robust but mostly limited to industrial and commercial environments and expensive. Moreover, if normal 2-D camera images are used, it is often hard to tell translation changes from orientation changes during robot motion.

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. One way to reduce this number is to select the most expressive inputs with regard to the desired system output (Input Selection) [8] or by statistical analysis of the input patterns using techniques like the principal components analysis (PCA). Hancock and Thorpe [5] 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 [9] 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 [6] 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. [15] 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. [4] 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 [6], the generated vector is used as the input of a three-layered neuronal network.

3 Vision Based Situation Assessment

In our earlier work on robot navigation [18], we developed a “situation-based” control for input from simple infra-red 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, a robot position can be classified into an area. A fine localisation can be realised by a local controller which is specially trained for a situation, Fig. 1.

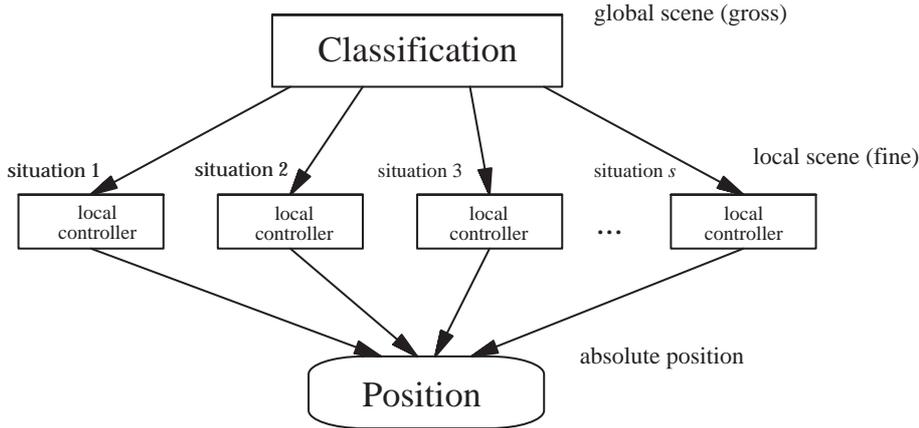


Fig. 1. Subdividing a global scene into local control problems.

3.1 Reducing the Input Complexity

If the sensor input is high-dimensional as in the case of raw vision data, general and scalable situation classifiers need to be developed. Theoretically, fuzzy c-means clustering may be used for this purpose. However, applying the fuzzy c-means method directly to a sequence of original images is computationally expensive. If a sequence of feature projections of the original images is considered, the input dimension is lower but there still exist two practical problems. If the dimension of the eigenspace is too small, the loss of information gives rise to ambiguity and complex situations cannot be clearly separated. If, by contrast, the dimension of the feature space is selected great enough to uniquely classify the desired situations, the computation time for cluster calculation is often not affordable. Additionally, it is necessary to find clusters which are coherent with regard to the desired outputs.

In the following, we first propose an approach of using *Output Related Features* (ORF), an improved variant of PCA for supervised learning, to reduce the high dimension of the input space. Such a method works well if the observed process is constrained within a low variance scenario where the input patterns bear a certain similarity. For a global scenario which does not meet this requirement, we suggest to apply the situation-based control scheme. The complexity raw input data, i.e. the visual images, is reduced by compression techniques and then fed into a situation classifier, which, in turn, steers several situation-specific controllers.

Situation-based control architecture has been applied in autonomous mobile robots to handle complex systems, [2,3]. Using fuzzy controllers to implement such an architecture is reported among others in [16,18,7,12]. Most of the above systems make use of low-dimensional sensor data. In this work, we use high-dimensional vision data for localisation without using complex algorithms for extracting geometric features. Therefore, how to represent, classify and

recognise situations using vision data becomes a new problem.

3.1.1 PCA

PCA has been mainly applied to data compression and pattern recognition [11,13]. Recently, this technique is also applied for reducing the dimension of the input space of a general control problem, [5],[10].

For an input space $X_1 \times X_2 \times \dots \times X_m$, if all the variables x_1 to x_m may vary over all their universes in the sampling procedure, input data will be scattered over the whole input space and possible compression ratios are low. Nevertheless, if the observed high dimensional process runs continuously, the input vector often varies only gradually. If the observations meet certain constraints, e.g. different images are taken within a local scenario of the robot environment while the robot is not moving too fast, in most cases the input vectors are similar to each other. In other words: the observed high-dimensional input data are highly correlated.

Let us assume an input vector \vec{x} made up of the random variables (x_1, x_2, \dots, x_m) originating from a pattern-generating process. In our case \vec{x} is the stacked one-dimensional image vector with 80×20 elements resulting from 80 pixel rows and 20 pixel lines with dimension m . First, the expectation value $\vec{\mu}$ and the covariance matrix \mathbf{Q} of these vectors are computed according to

$$\begin{aligned}\vec{\mu} &= (E(x_1), E(x_2), \dots, E(x_m))^T \\ \mathbf{Q} &= E[(\vec{x} - \vec{\mu})(\vec{x} - \vec{\mu})^T]\end{aligned}$$

The eigenvalues and eigenvectors can then be computed by solving

$$\lambda_i \vec{a}_i = \mathbf{Q} \vec{a}_i$$

where the λ_i are the m eigenvalues and the \vec{a}_i are the m -dimensional eigenvectors of \mathbf{Q} . Since \mathbf{Q} is positive definite, all eigenvalues are positive. Extracting the most significant structural information from the set of input variables (x_1, x_2, \dots, x_m) is equal to isolating the largest n eigenvalues $\lambda_1, \dots, \lambda_n$ with $(n < m)$ and their corresponding eigenvectors \vec{a}_i . If we now define a transformation matrix

$$\mathbf{A} = (\vec{a}_1 \dots \vec{a}_n)^T$$

we can reduce the dimension of \vec{x} by

$$\vec{p} = \mathbf{A} \cdot \vec{x}; \quad \dim(\vec{x}) = n$$

i.e. by projecting \vec{x} into the subspace spanned by $\vec{a}_1 \dots \vec{a}_n$. This projection is illustrated in the left part of Fig. 2. In a subspace spanned by the feature vectors, we can easily apply a universal function approximator, in our case a B-spline fuzzy controller, by covering ORFs with linguistic terms, see the right part of Fig. 2.

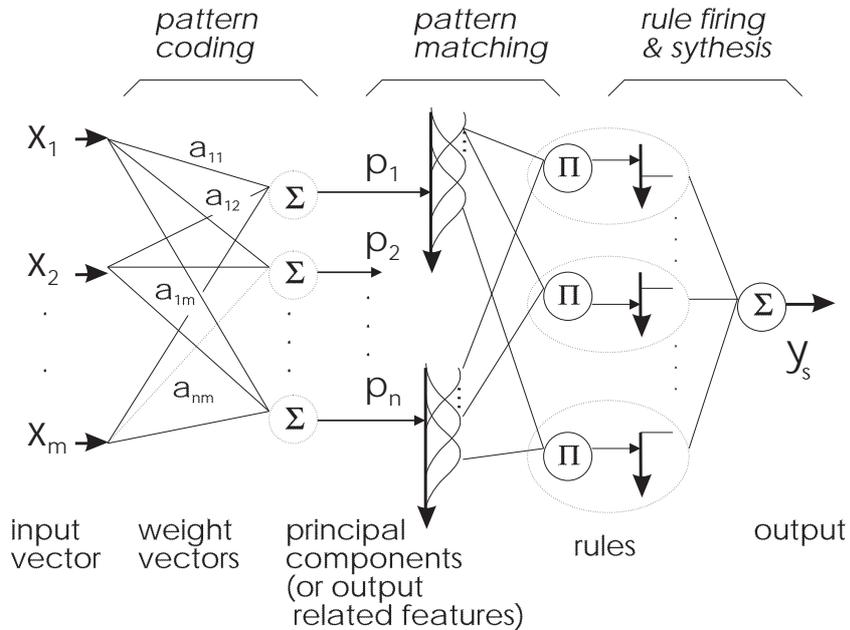


Fig. 2. A neuro-fuzzy model for visually-guided control.

Depending on how “local” the measuring data are and therefore how similar the observed sensor images look like, a small number of eigenvectors can provide a good summary of all the input variables. The dimension n should be determined depending on the discrimination accuracy needed for further processing steps vs. the computational complexity that can be afforded. It can be possible that two or three eigenvectors supply most of the information indices of the original input space. The determination of n for our experiments is currently purely heuristic and an automatic method is subject of further study.

The eigenvectors of a covariance matrix can be efficiently computed by the perceptron approach [13]. Each input data vector is multiplied with the eigenvectors and becomes a point in the eigenspace. A sequence of continuous input data is transformed into a manifold in the eigenspace. Fig. 3 shows an example of an image sequence in our scenario, Fig. 4 gives the projection of these images in a three-dimensional eigenspace.



Fig. 3. Sample images of a motion along a wall.

3.1.2 *Output Related Features*

As outlined above, PCA is a suitable approach for dimension reduction. 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 Related Features* (ORF).

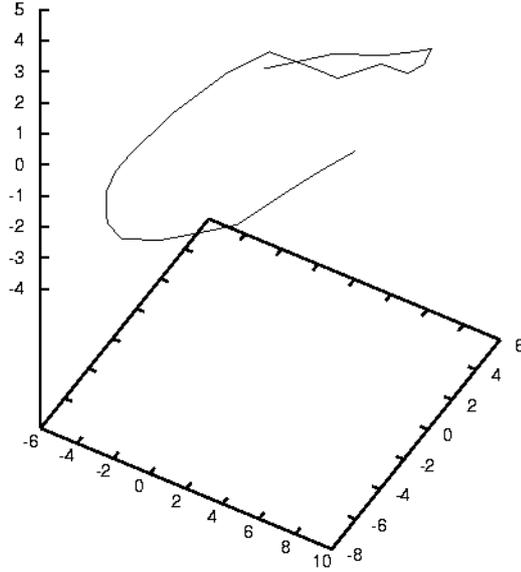


Fig. 4. Projection of images into a manifold in a 3D eigenspace.

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 ($j = 1, \dots, k$). If one ORF weight vector is trained which is denoted by \vec{a} , then the network output P is:

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

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:

$$\Delta a_j = \eta(Y_S - P)x_j \quad (2)$$

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. [17]. 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:

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

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 (alone) because they take into account the input-output relation. When modelling a complex non-linear 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. The rest of the task, i.e. modelling the non-linear part, is performed by a fuzzy controller.

3.2 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. Figs. 5 and 6 illustrate the process in a simplified manner. If 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. 5), more dimensions than shown in the figures are needed (12 in our experiment).

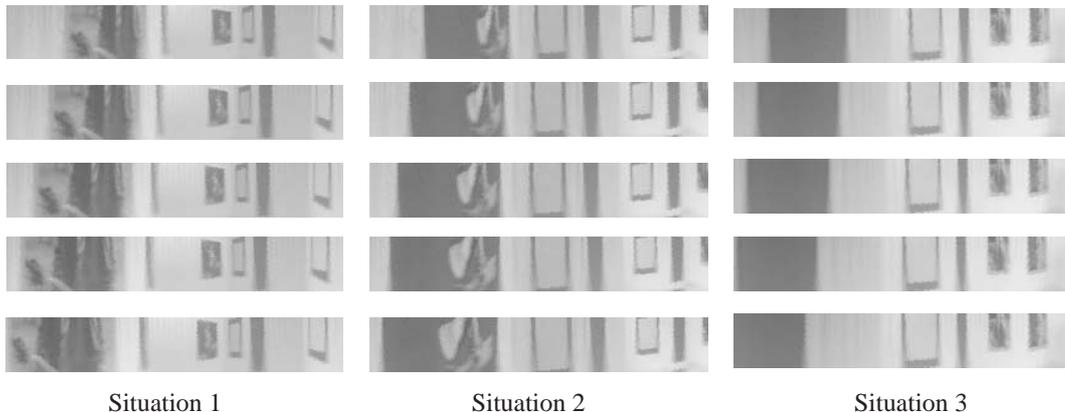


Fig. 5. Views from the robot camera used in Fig. 7.

3. Principal Component

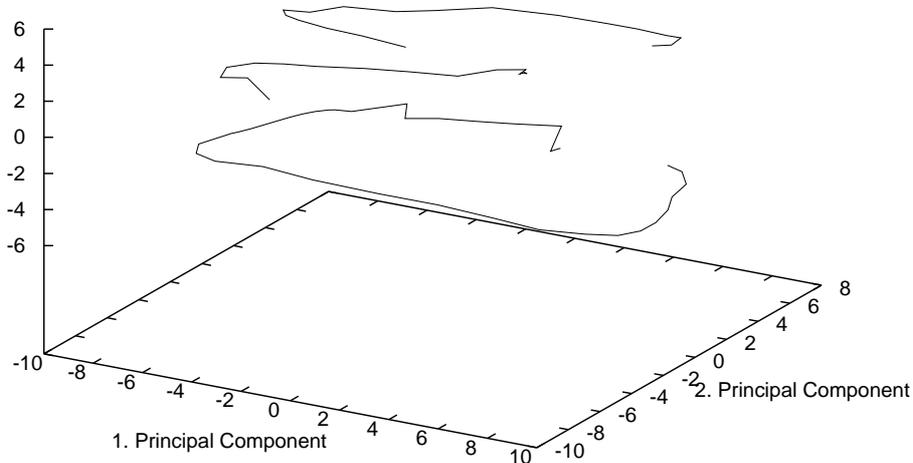


Fig. 6. Situation manifolds in eigenspace. These three manifolds represent parts of the situations shown in Fig. 5.

3.3 Transformation of the Omnidirectional Vision Data

For identifying the situations in our context we have developed the omnidirectional vision system for mobile robots shown in Fig. 7, which provides a complete view of the environment. To fully utilise the global and sometimes redundant information, it is separated into overlapping parts. These parts are input to the controller (Fig. 8). The viewing area of the camera is divided into multiple sections of the same size. In our experiment, it was found that a viewing area of 180° is in most practical cases sufficient for a unique localisation of our robot. 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. Afterwards, 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 the input of a fuzzy controller that interpolates the x -/ y -position.

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.

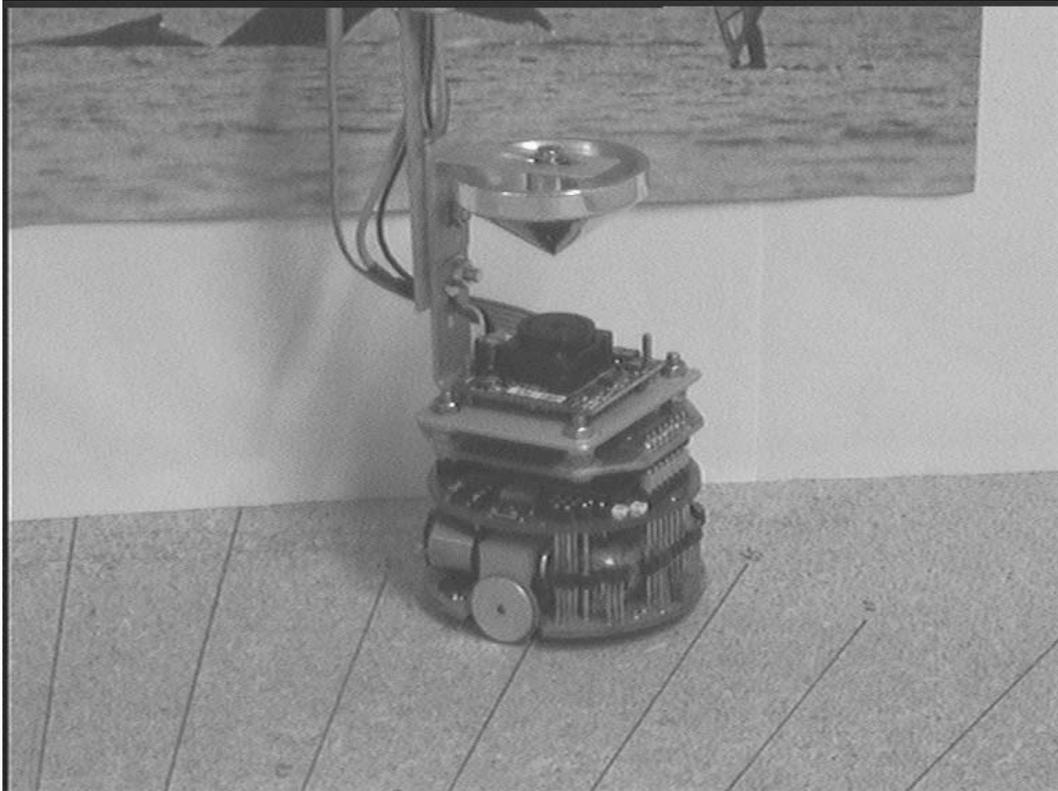


Fig. 7. The mobile robot with an omnidirectional camera.

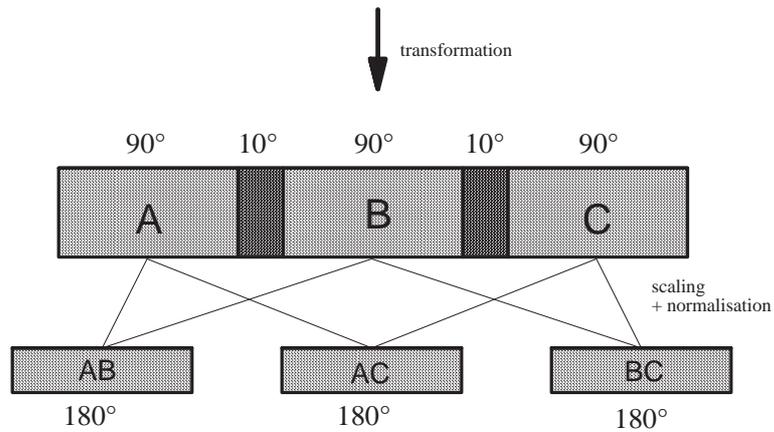


Fig. 8. Sectoring by constructing pseudo-segments.

3.3.1 Situation Classification without Global Eigenspace

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. To increase the reliability, the use of further sensors, e.g. the R-, G- and B-channel of colour images, may easily be added to increase the robustness of the classification.

4 Applications

In this section we describe both the hardware implementation of our localisation system and the software implementation. We also show a number of real images taken during several wall following runs of our robot.

4.1 *Experimental Setup*

The vision system consists of only two components: a subminiature B&W camera looking “upright” and a conical mirror of polished aluminium. The complete setup is shown in Fig. 7. The test environment consists of a miniature “doll’s house” of 40cm×40cm in size. The walls are coated with textured wall paper and the “room” includes several pictures, windows and doors. As a typical problem with high-dimensional input, we investigate the situation shown in Fig. 9.

To automate the process of *collecting training images*, the built-in infrared proximity sensors are used for guiding the robot along a wall. Readings of the motor encoders are reliable over a short distance when only a single motion is performed; in conjunction with the proximity sensors they are used as the input data for training the B-spline fuzzy controllers. All higher level tasks are performed on a standard *Pentium 100 PC* running under *Linux*. The PC and robot communicate via an RS-232 interface. Images are acquired by a low-cost frame-grabber.

4.2 Implementation

The robot starts at an arbitrary position in the room. It moves straightforward until sensor readings indicate the presence of a wall. This wall is recorded as the initial situation. As the robot detects the end of the wall, it turns parallel to the wall perpendicular to the first one. Every 15 mm an image is taken by the vision system and the position supplied by the wheel encoders is recorded. To enhance the robustness of the system against motion uncertainties like slight rotation, the robot is rotated deliberately at every position (in seven consecutive steps) to learn the effect of (unwanted) rotation on the camera input.

To make the input patterns less prone to changes in illumination conditions, all images are min-/max-normalised to the interval $[0, 1]$. The normalisation reduces the sensitivity against brightness fluctuations.

The concentric images are further converted into planar images via a standard tangent transformation. The converted images cover the range of -45 to 225 degrees. Only the slice between 0 and 180 degrees of this area is used, thus reducing the dimension of the input set to $m = 80 \times 20$ (pixels). Finally, the average image in each situation is subtracted from each image. This process is illustrated in Fig. 9.

Fig. 10 shows the situation-based control architecture we implemented on this robot. As described in subsection 3.3, after the sectoring the pseudo-segments are prepared first and then each of them is projected onto the ORF weight vectors. In our experiment, the three first ORFs extracted from the three pseudo-segments are used as inputs for the B-spline fuzzy controller, i.e. $n = 3$. This three-dimensional ORF vector plus the recorded internal x -/ y -positions are taken as input/output pairs for training the fuzzy controllers.

Fig. 11 shows four sequences of raw images that are assigned to different situations. Since the ORF vectors possess the same dimension as the original data, they can be converted into images. Fig. 12 and 13 illustrate the ORF vectors after addition of the average images, without any rotation and with seven “artificial” rotations, respectively.

4.3 Using the Adaptive B-Spline Neuro-Fuzzy Model

To interpolate between any two samples of the training set, fuzzy controllers can be used. In the following implementation, fuzzy controllers constructed according to the B-spline model [19] are used. This model provides an ideal implementation of the CMAC architecture [1].



(a) 2D image of a corner.



(b) Conical image of the complete scene.



(c) The tangent transformed image.

Fig. 9. Image transformation for the robot, (a) \rightarrow (b) \rightarrow (c).

In the B-spline model we define linguistic terms for input variables with B-spline basis functions and for output variables with singletons. This requires fewer parameters than other set functions such as trapezoid, Gaussian, etc. The output computation becomes very simple and the interpolation process is transparent (details are explained in [19]). We note that there are two main reasons for utilising a B-spline fuzzy controller in this context:

- The global minimum of the squared error function is guaranteed to be found if the partitioning of the input space is fixed;

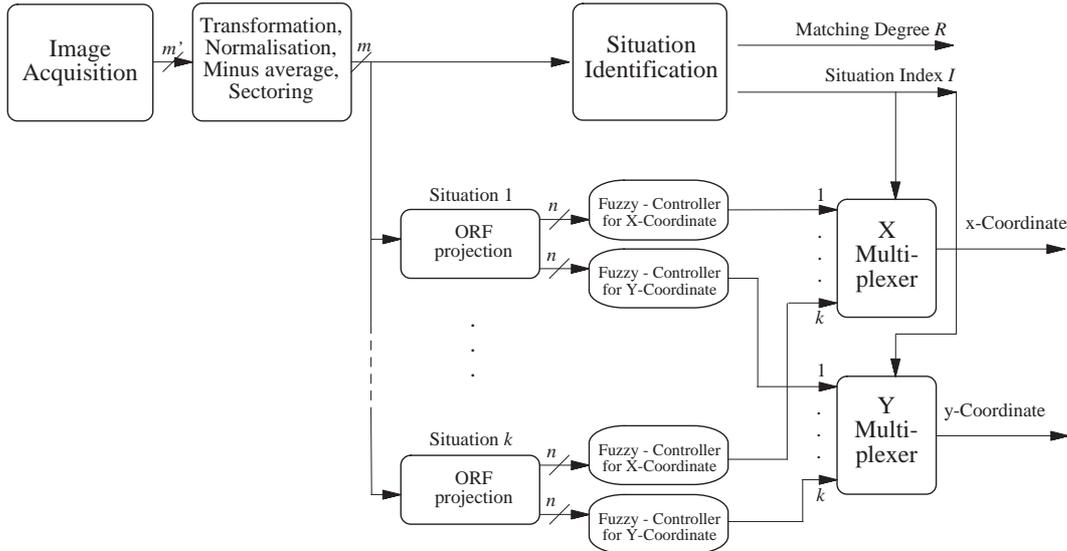


Fig. 10. Situation-based visual localisation controller.

- Supervised learning using gradient descent is easy to implement and enables rapid convergence.

At run time (i.e. after the learning phase) the input data are first projected onto the ORF vectors, then mapped to the output variables based on the fuzzy control model, Fig. 2. The fuzzy rule-base is generated by *gradient descent* supervised learning. The performance of the controllers can be enhanced further by optimising the shape of the B-spline basis functions used as fuzzy sets, Fig. 14. The principle of this optimisation is to move the fuzzy sets into the direction of the data point causing the maximum error.

Using the ORF-based fuzzy control approach to reconstruct the robot position, the maximum error with the test data can almost invariably be reduced to 3% or less. By comparison, the maximal test error of a PCA approach is at least ten times larger.

All controller outputs are interconnected using two multiplexers addressed by the situation index I . The implemented system currently achieves a refresh rate of about 10 ... 12 cycles per second (standard image processing hardware on the Pentium 100).

5 Discussion

We showed that a localisation approach based on the visual appearance of an arbitrary environment as the sole input is feasible. The experimental environment is comparable to a typical living room or office, where mobile robots can find potential applications for service jobs. The further development of this

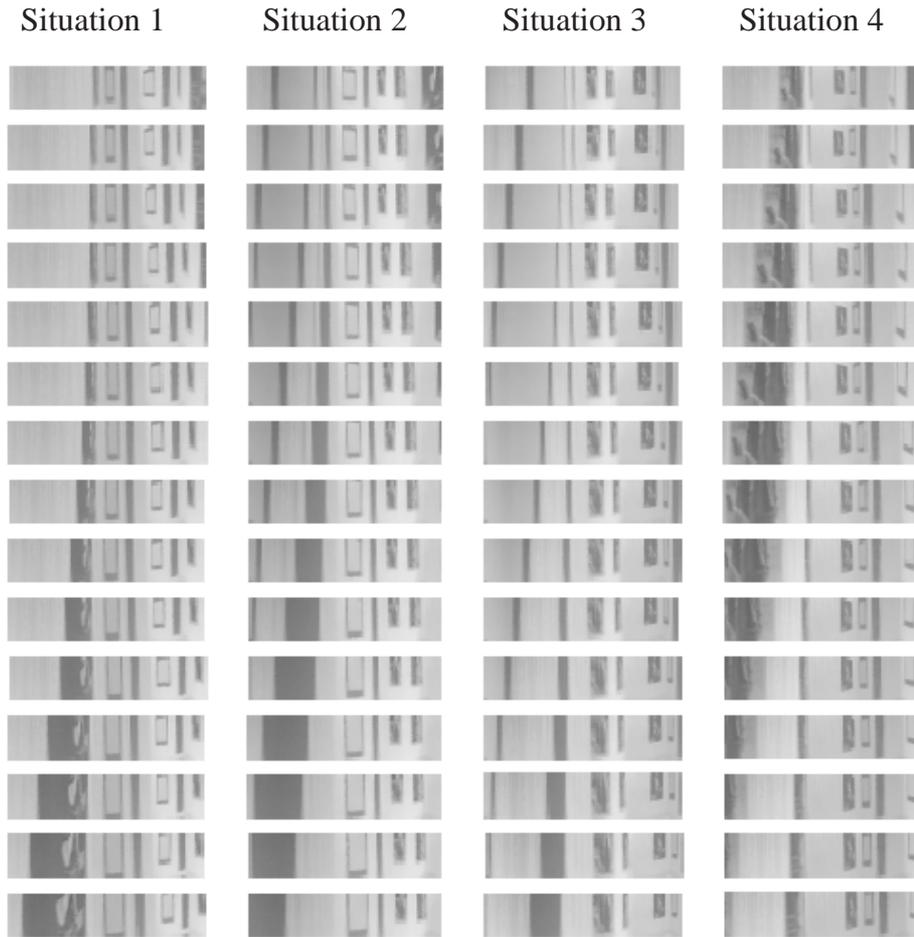


Fig. 11. Image sequence of the training situation 1 - 4.

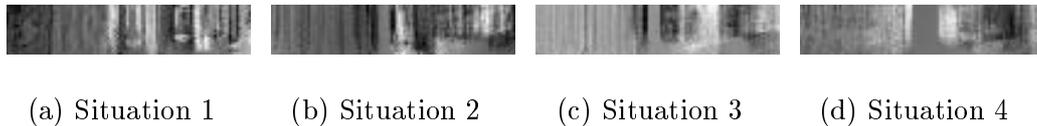


Fig. 12. The first ORF vectors with addition of the average image (0 rotations).

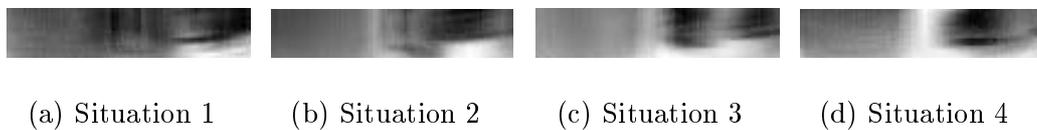


Fig. 13. The first ORF vectors with addition of the average image (7 rotations).

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

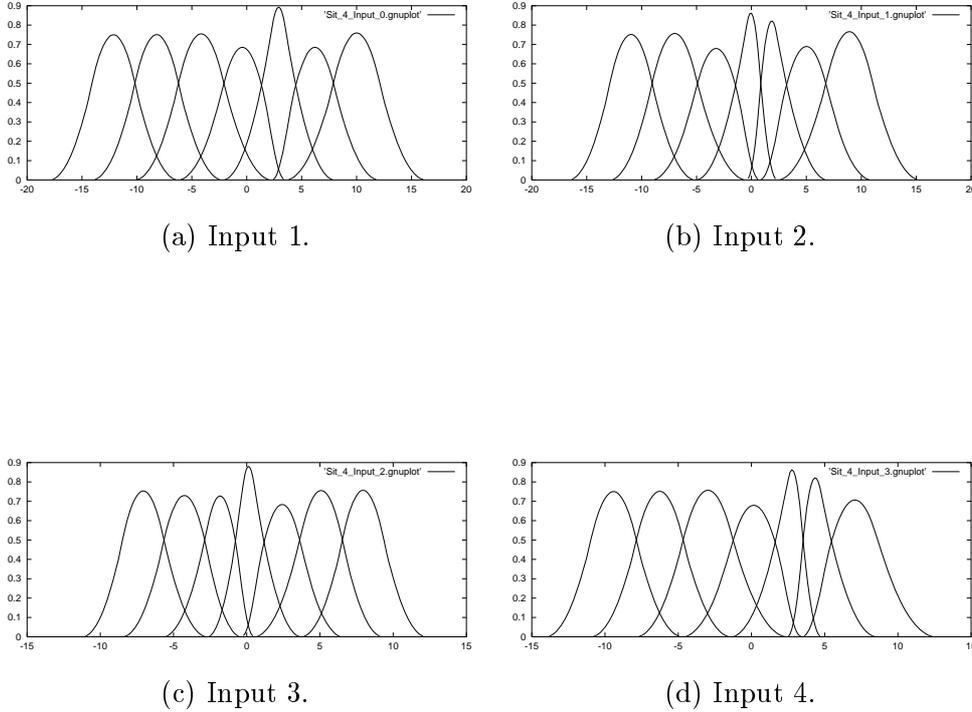


Fig. 14. Input variables (eigenvectors) modelled with B-spline basis functions.

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 including any kind of objects.

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.

This approach can be applied to a large set of control problems of complex systems. Actually, we have also used it successfully in the vision-based grasping with manipulators. The limits of single one-step fuzzy controllers are overcome; the system input need not be low-dimensional. The situation-based control scheme is also “cognitively adequate”. Computation complexity grows polynomially.

We are currently investigating some improvements. 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 interpolation precision required and the maximum memory allowed. 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 also necessary to combine this localisation approach with other mobile robot tasks like collision avoidance, object recognition (e.g. for grasping) and tracking.

Acknowledgements

The contributions of the anonymous reviewers that greatly helped us improve the quality of the paper are gratefully acknowledged.

References

- [1] J. S. Albus. A new approach to manipulator control: The Cerebellar Model Articulation Controller (CMAC). *Transactions of ASME, Journal of Dynamic Systems Measurement and Control*, 97:220–227, 1975.
- [2] R. C. Arkin. Integrating behavioural, perceptual, and world knowledge in reactive navigation. *Journal of Robotics and Autonomous Systems*, (6):105–122, 1990.
- [3] H. R. Berenji, Y-Y Chen, C-C Lee, J-S Jang, and S. Murugesan. A hierarchical approach to designing approximate reasoning based controllers for dynamic physical systems. In *Proceedings of the 6th Conference on Uncertainty in Artificial Intelligence, Cambridge, MA*, pages 362–369, 1990.
- [4] R. Cassinis *et al.*, Using Colour Information in an Omnidirectional Perception System for Autonomous Robot Localisation. Proceedings of EUROBOT'96, pp. 172-176.
- [5] J.A. Hancock and C.E. Thorpe, ELVIS: Eigenvectors for Land Vehicles Image System, CMU-RI-TR-94-43.
- [6] 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.

- [7] S. G. Goodridge, M. G. Kay, and R. C. Luo. Multi-layered fuzzy behaviour fusion of reactive control of an autonomous mobile robot: MARGE. In *IEEE International Conference on Fuzzy Systems*, pages 579–584, 1997.
- [8] J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice Hall, 1997.
- [9] 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.
- [10] S. K. Nayar, H. Murase, and S. A. Nene. Learning, positioning, and tracking visual appearance. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3237–3244, 1994.
- [11] E. Oja. *Subspace methods of pattern recognition*. Research Studies Press, Hertfordshire, 1983.
- [12] A. Saffiotti. The uses of fuzzy logic in autonomous robot navigation. *Soft Computing 1*, pages 180–197, 1997.
- [13] T. Sanger. *An optimality principle for unsupervised learning*. Advances in neural information processing systems 1. D. S. Touretzky (ed.), Morgan Kaufmann, San Mateo, CA, 1989.
- [14] R. Wehner, M. Lehrer, and W. R. Harvey. Navigation. *The Journal of Experimental Biology*, 199(1), January, 1996.
- [15] 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.
- [16] J. Yen and N. Pfluger. A fuzzy logic based extension to Payton and Roseblatt’s command fusion method for mobile robot navigation. *IEEE Transactions on System, Man and Cybernetics*, 25(6):971–978, June, 1995.
- [17] A.L. Yuille, D.M. Kammen, and D.S. Cohen. Quadrature and the Development of Orientation Selective Cortical Cells by Hebb Rules. *Biological Cybernetics*, 61, 1989.
- [18] J. Zhang, F. Wille, and A. Knoll. Modular design of fuzzy controller integrating deliberative and reactive strategies. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, 1996.
- [19] J. Zhang and A. Knoll. Constructing fuzzy controllers with B-spline models - principles and applications. *International Journal of Intelligent Systems*, 13(2/3):257–286, February/March 1998.