

# An Adaptive Model for Vision-Based Localisation

J. Zhang, V. Schwert, and A. Knoll

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

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

We introduce a neuro-fuzzy model for localising a mobile robot based on raw vision data. Not only does the approach involve adaptive fuzzy control but it also integrates data compression such as the principal component analysis (PCA) as well as pattern matching in eigenspace. A typical nonlinear real-world system may be modelled as a control process with a large number of sensor inputs. By observing the control process in a continuous time interval, high-dimensional sensor data and eventually some desired control information can be sampled. With the help of PCA, the original dimension of input data can be reduced to a tractable size. Subsequently the data can be clustered into a finite sequence of local “scenarios”. Within each local scenario, an adaptive fuzzy controller is trained to achieve the desired performance.

## 1 Introduction

One important issue of studying natural life is the localisation mechanism of animals through observing the environment. For such a task, human-beings rely to a great extent on reading maps, looking for street names in a structured city environment, or memorising some distinctive features. Some remaining questions are: How can cats, pigeons,

or salmons, etc., know “where I am” and find a desired place where they have been? What kind of information has to be memorised?

The localisation capability of these animals attributes partly to the extraordinary non-visual sensing like scent and magnetic sensing, which is still too sophisticated for the current technical sensors to emulate. However, visual sensors are becoming more affordable and easier to use in robot systems. An interesting but nontrivial problem is to reconstruct the robot position/orientation only from visual data without using any artificial landmarks or symbolic signs. Does a general model exist for that purpose? What kind of features extracted from the visual data are most efficient?

An appropriate scenario for technical experiment is the navigation and localisation of a mobile robot. For that purpose, numerous approaches have been developed during the recent years. The need for a robust and accurate localisation method is obvious. The major objective of most localisation approaches is to update and 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 wheel slide and frequent orientation changes. It is therefore common to combine odometric sensors with standard cameras. With a normal 2-D camera image, however, it is often hard to tell translation from orientation change during robot motion. By contrast, an omni-directional vision system provides a greater degree of uniqueness of the acquired images. With an omni-directional system changes in translation and orientation always result in different image. In the work by Cassinis [2], radial lines are extracted via edge detection and a neural network is trained to identify and interpolate between known patterns.

This method and similar approaches share the same limit: The complexity of the control system rises exponentially with the number of inputs used. One way to reduce this number is to select the most expressive inputs with regard to the desired system output [4] or by statistical analysis of the input patterns using techniques like PCA. Some navigation applications [3] using only the eigenvector information of an input set to autonomously navigate a car proved very successful.

## 2 Concept of Reducing Complexities

In the following, we first propose using PCA to solve the problem of high-dimensional input space. Such a method works well if the observed procedure is constrained within a scenario where the input patterns are similar to a certain degree. For a global scenario which does not meet this requirement, we suggest to apply a situation-based control scheme.

### 2.1 PCA as Pre-selection of Controller Inputs

#### 2.1.1 Projection in Eigenspace

For an input space  $X_1 \times X_2 \times \cdots \times X_m$ , if all the variables  $x_1$  to  $x_m$  may vary in all their universes in the observed sampling procedure, input data will be scattered over the whole input space. Nevertheless, in a high-dimensional input space, if the observed process runs continuously, the input vector varies only gradually. Under an appropriately separated observation, e.g. within a local scenario of the robot environment, in most cases the input vectors possess a large grade of similarity. In other words: the observed input data are correlated to a large degree in the high-dimensional input space.

An eigenvector, noted as  $EV_i$ , is computed as  $[a_{1,i}, a_{2,i}, \cdots, a_{m,i}]^T$ . The eigenvectors form an orthogonal basis for representing the original individual sensor images. Assume that the eigenvectors  $EV_1, EV_2, \cdots$  are sorted according to their eigenvalues in a descending order. An eigenspace with a reduced dimension  $n$  can be formed with the first  $n$  eigenvectors.  $EV_i$  accounts for the  $i$ th dimension in the eigenspace. The projection of an input vector  $[x_1, x_2, \cdots, x_m]$  onto eigenvector  $EV_i$ , called the  $i$ th principal component, is  $p_i = a_{1,i}x_1 + a_{2,i}x_2 + \cdots + a_{m,i}x_m$ . All projections of the sample data sequence form a manifold in the eigenspace.

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. It can

be possible that two or three eigenvectors supply most of the information indices of the original input space. In a very high-dimensional case, an efficient dimension reduction can be achieved by projecting the original input space into the eigenspace. This step is illustrated in the left part of Fig. 1.

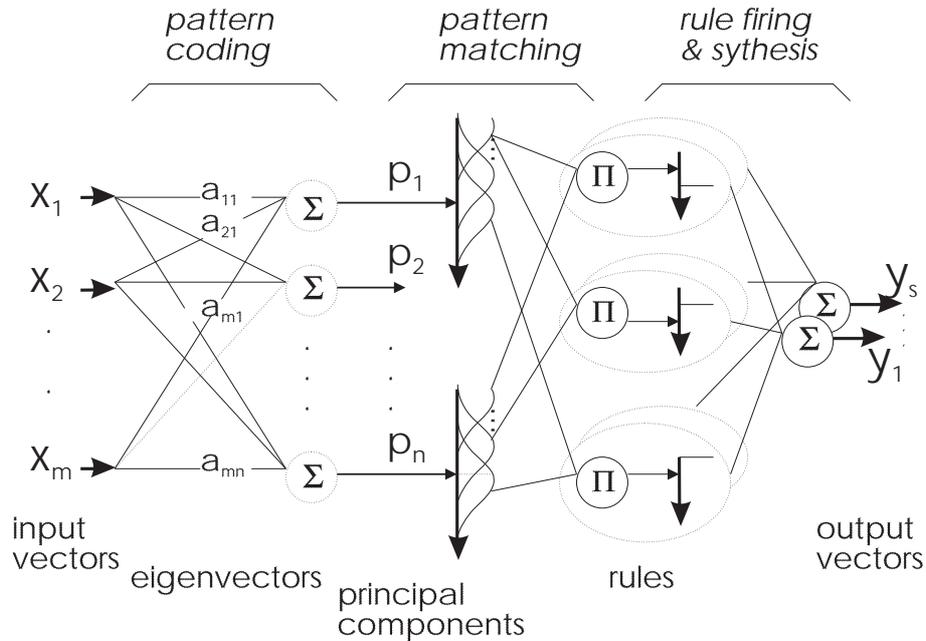


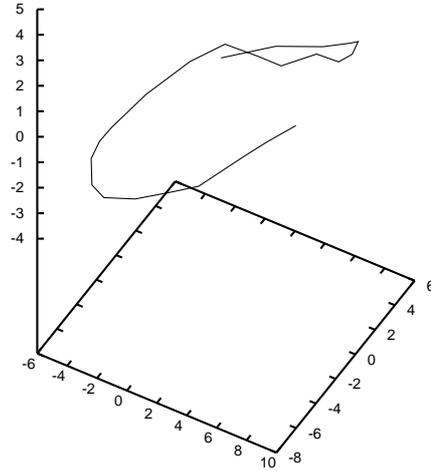
Figure 1: A neuro-fuzzy model for visually-guided control.

The eigenvectors of a covariance matrix can be efficiently computed by the perceptron approach [6]. This approach has three advantages over Jacobian method if the number of the eigenvectors are pre-selected. First, the Jacobian method calculates all eigenvectors of the covariance or explicit covariance matrix of the input data; the perceptron method only calculates the first  $n$  eigenvectors, thus saves computation time. Second, Jacobian requires the construction of the covariance matrix of the input data. With large input data sets this is often critical regarding the amount of memory needed. Thirdly, the found eigenvectors with the perceptron method are already in descending order.

Fig. 2(a) depicts an example of an image sequence in our scenario, Fig. 2(b) the projection of these images in a three-dimensional eigenspace.



'manifold.ps' ———



(a) Sample images.

(b) The manifold in a 3D eigenspace.

Figure 2: Projection of images into eigenspace – motion along a wall.

### 2.1.2 Fuzzy Partitioning into Eigenspace

Partitioning of eigenvectors can be done by covering eigenvectors with linguistic terms, see the right part of Fig. 1. In the following implementations, fuzzy controllers constructed with the B-spline model [8] are used. This model provides an ideal implementation of CMAC [1]. We define linguistic terms for input variables with B-spline basis functions and for output variables with singletons. Such a method requires less parameters than the other set functions such as trapezoid, Gaussian function, etc. The output computation becomes very simple and the interpolation process is transparent. We also achieved good approximation capability and rapid convergence of B-spline fuzzy controllers.

The details about the B-spline fuzzy controller are explained in [8]. There are mainly two reasons to use a B-spline fuzzy controller: a) The global minimum is guaranteed if the partition of the input space

is fixed; b) Simple supervised learning using gradient descent is easy to implement and enables rapid learning.

In the on-line application, the input data are first projected into the eigenspace, then mapped to output based on the fuzzy control model, Fig. 1.

## **2.2 Situation-Based Control Scheme**

### **2.2.1 Classifying Situations**

The general strategy for coping with non-linear high-dimensional control problems is “divide and conquer”: classify the whole control task 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. By applying the approach proposed in section 2.1 to reduce the high-dimensional input data into a small number for each local situation, adaptive fuzzy control can be employed to realise the desired local control task. The key point is then how to find a scheme which is scalable to a general problem. For practical control this means that the computational complexity is polynomial instead of exponential.

In our earlier work [7] on robot navigation, we developed “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 for local collision avoidance and it must reduce the weight of goal tracking. A “situation evaluator” was constructed by heuristic fuzzy rules.

If the sensor input is high-dimensional like in the case of raw vision data, general and scalable situation classifiers need to be studied. Fuzzy c-means clustering is theoretically suitable to a task of situation classification. However, directly applying the fuzzy c-means method to a sequence of original images is computationally expensive. If a sequence of eigenspace projections of the original images is considered, there still exist two practical problems. If the dimension of eigenspace is selected to be too small, complex situations cannot be clearly separated because ambiguity occurs due to the loss of information. If the

dimension of the eigenspace is selected to be big enough to uniquely classify the desired situations, the computation time is still an obstacle to the cluster calculation. Additionally, it is necessary to find clusters which are coherent with regard to the desired outputs. Therefore, instead of fuzzy c-means, we employ the method of “Nearest Neighbourhood” for classifying manifolds into an eigenspace for arbitrarily many situations.

### 2.2.2 High-Dimensional Classifier in Eigenspace

If a global eigenspace is used to project all the situation-related images, the projections of each situation can be approximately viewed as a single manifold. If the dimension of the eigenspace is big enough, these manifolds are uniquely separated. Fig. 3 and 4 illustrate the process in a simplified way. To really differentiate between the situations (“walls” in Fig. 3), more dimensions are needed (12 in our experiment). If represented in this way, the *matching degree* of a situation of a new image can simply be defined as the distance between the projection point and the manifold of this particular situation.

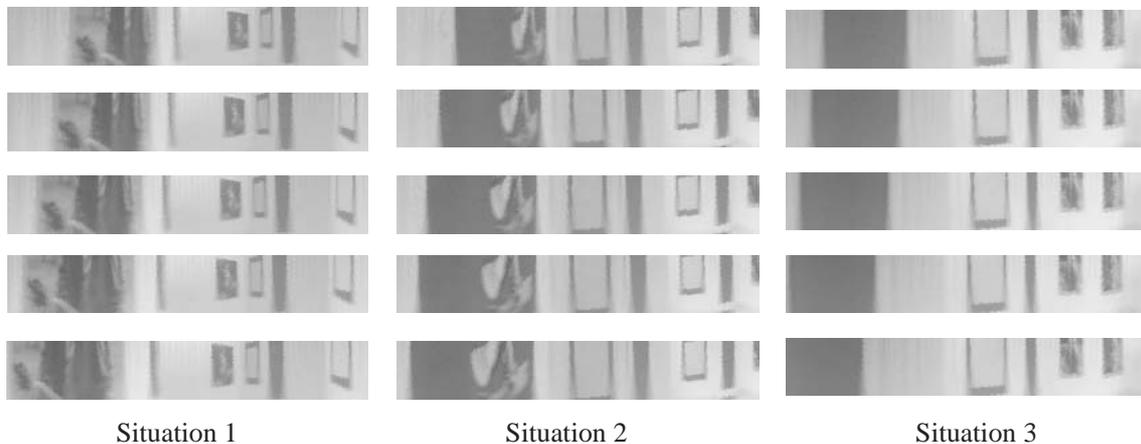


Figure 3: Views of the robot camera used in Fig. 5.

3. Principal Component

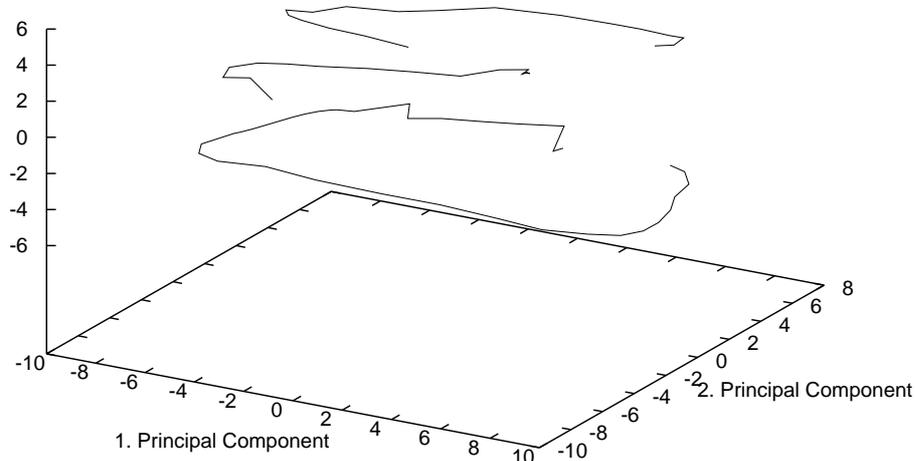


Figure 4: Situation manifolds in eigenspace. These three manifolds represent parts of the situations shown in Fig. 3, respectively.

## 3 Application

### 3.1 Experimental Setup

The test environment consists of a miniature room of  $40\text{cm} \times 40\text{cm}$  in size. The walls are coated with textured wall paper and the room includes several pictures, windows and doors. An omni-directional vision system is mounted on top of a *Khepera* mini-robot. The vision system has only two components: a subminiature B&W camera looking “upright” and a conical mirror of polished aluminium. The total setup is shown in Fig. 5. As a typical problem with high-dimensional input, we investigate the situation shown in Fig. 6.

The distance to the walls and other objects is measured using the built-in infrared proximity sensors. The robot’s internal micro-controller is only used for internal localisation by analysing the robot’s motor encoders, and motor/sensor control. All higher level tasks are done on a standard *Pentium 100 PC* running *Linux* as operating system. The PC and robot communicate via a RS232 interface. Images are

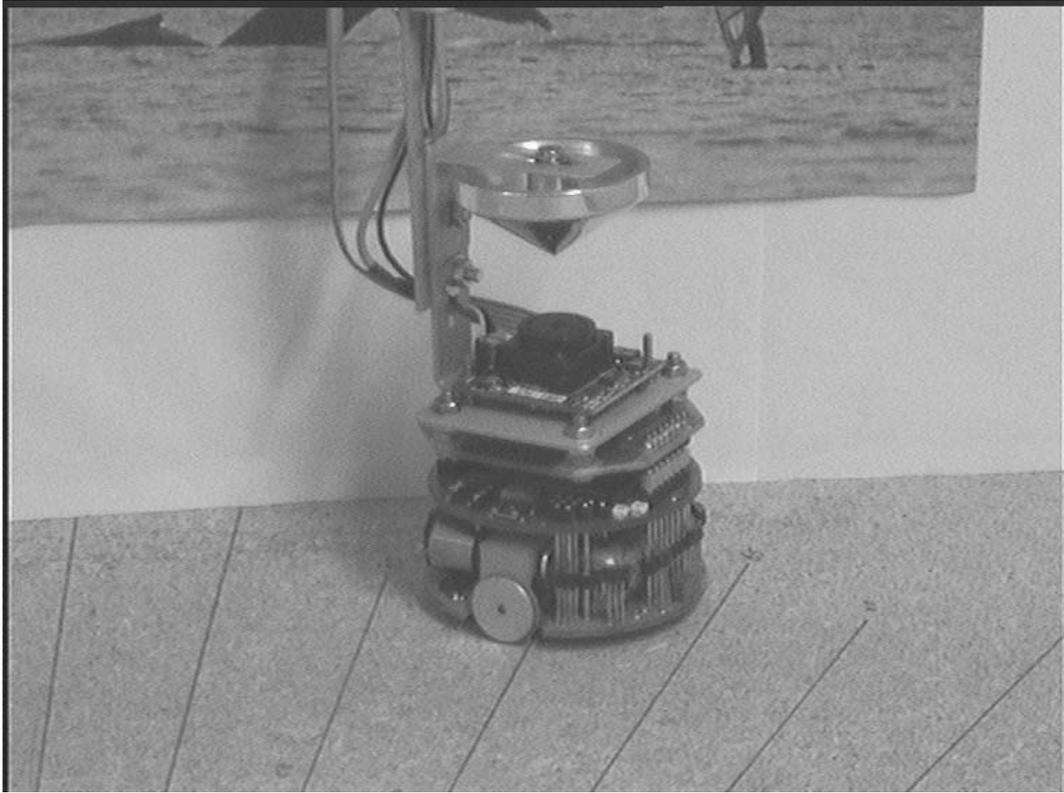


Figure 5: The mobile robot mounted with an omni-directional camera.

acquired by a low-end framegrabber.

It is a long-term research goal to find a general model which transforms raw image data directly to action values. For instance, an image has  $512 \times 512$  pixels, each pixel is measured by brightness level. If no extra image processing is performed, then a control system with about a quarter of million input variables needs to be modelled. The outputs of the system are the position/orientation or the motion values of a robot.

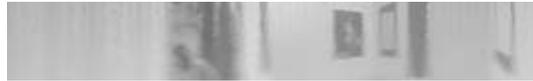
### 3.2 Implementation

The robot moves along the walls until it finally reaches its start position. At equidistant time steps images from the omni-directional vision system are taken and the position/orientation of its internal localisation module is recorded. To make the input patterns less prone to changes in illumination conditions all images are min-/max-normalised to the interval  $[0, \dots, 1]$ . The concentric images of dimension  $M$  are



(a) 2D image of a corner.

(b) Conical image of the complete scene.



(c) The transformed image.

Figure 6: Image pre-processing for the robot, (a)  $\rightarrow$  (b)  $\rightarrow$  (c).

further converted into planar images via tangent transformation. The converted images cover the range of  $-45$  to  $225$  degrees. Only the slice of  $0$  to  $180$  degrees of this area is used, thus reducing the dimension of the input set to  $M'$  ( $80 \times 20$  pixels). This process is illustrated in Fig. 6.

Fig. 7 shows an implemented situation-based control architecture for localisation. Using PCA over the complete set of input images, the first  $G$  (12 in this application) eigenvectors are constructed. All images are projected on these vectors to build the *situation manifold* in the *global eigenspace*. Each situation represents one wall, i.e. the distance between two corners. New images are also projected into the eigenspace. By applying the *Nearest Neighbour* classifier the next known situation is found. The classifier returns a *matching degree*  $R$  and the *situation index*  $I$ . If no situation is found within a certain distance, the situation number is 0, otherwise it is in the interval  $[1, \dots, k]$ .

For construction of the *local eigenspace* of situation  $i$  only the images taken in situation  $i$  are used, thus covering much less variance

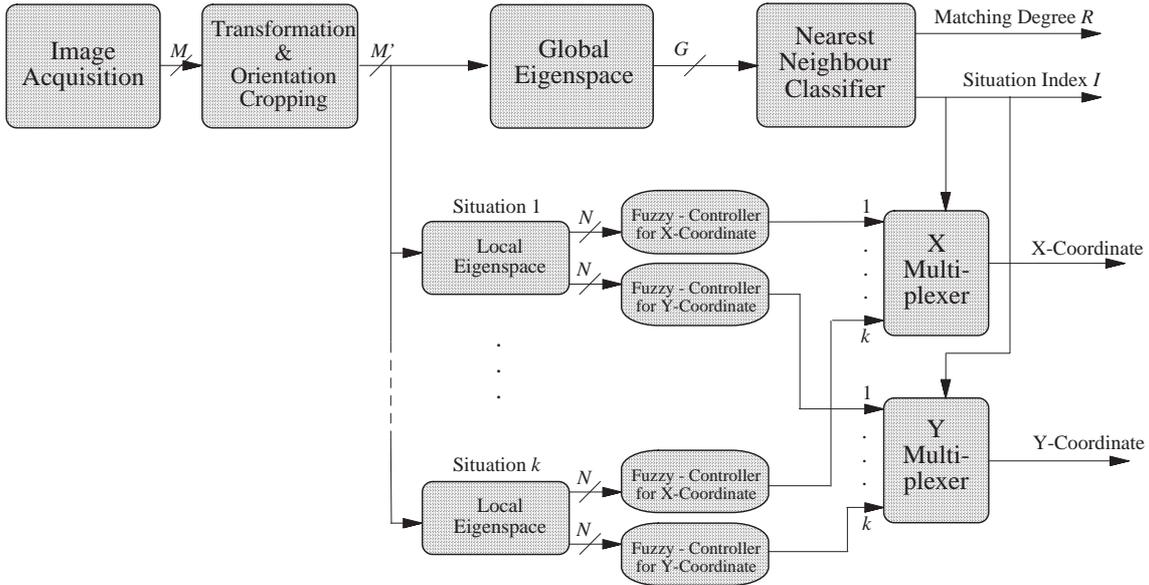


Figure 7: Situation-based visual localisation controller.

than the global image set. The *position manifold* is built the same way as the *situation manifold*. The local eigenspace has dimension  $N$ . The eigenspace projected images and the recorded internal  $x$ -/ $y$ -positions are taken as input/output pairs for the learning of the fuzzy controllers.

### 3.3 Using Adaptive B-Spline Neuro-Fuzzy Model

B-Spline fuzzy controllers are used to interpolate between any two samples of the training set. Learning of the fuzzy rule-base is done by applying *gradient descent* supervised learning to the controllers. The performance of the controllers can further be enhanced by optimising the shape of the B-spline basis functions used as fuzzy sets, Fig. 8. The principle of this optimisation is to move the fuzzy sets into the direction of the data point causing the maximum error. After the complete learning cycle, the manifold data is no longer needed and can be discarded. Experiments using only *Nearest Neighbourhood* for position control showed that much more input samples are needed to achieve a reasonably low position error and good interpolation quality. Using the situation-based fuzzy control approach, the maximum error with training data can always be reduced to 10 mm and less.

All controller outputs are proliferated using two multiplexers addressed

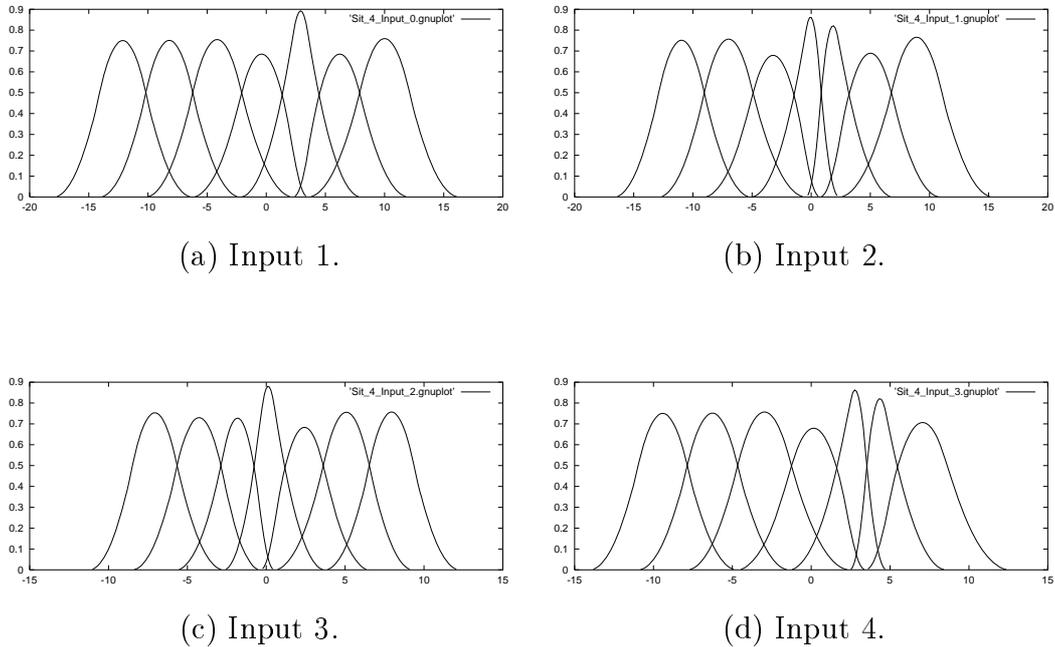


Figure 8: Input variables (eigenvectors) modelled with B-spline basis functions.

by the situation index  $I$ . The implemented system currently achieves a performance of about 10 - 12 cycles per second (standard image processing hardware on a Pentium 100).

## 4 Discussion

We showed that a localisation approach with the appearance of its environment as the sole input gives reliable results. The major advantage of this work is that no additional information of the environment is needed. Without usage of sophisticated geometric models, the computational costs are reduced. The amount of resources needed is linear to the number of situations the robot should learn.

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

adequate". Computation complexity grows polynomially.

Our current work is on the automatic selection of those system inputs that bear the highest information content to be exploited. At the moment the *Situation Classifier* takes the first 12 dimensions to construct the *global eigenspace*. By adding new situations to the system, this number will increase to uniquely separate the situation manifolds. One approach to limit this number is to use not the first 12, but the eigenvectors with the highest information content regarding the system performance. These eigenvectors represent the *Most Distinguishing Features (MDF)* of the eigenspace corresponding to the output. The MDF of the global eigenspace are characterised by maximising the inter-class distance and simultaneously minimising the intra-class distance.

One side-effect of using the *MDF* is the possibility to visualise what one situation distinguishes it from another. For interpolation within a situation it is desirable to use as few as possible inputs for the fuzzy controller. To limit the dimension of the *local eigenspace* on one side and to minimise the interpolation error on the other, it is reasonable to extract the *Most Expressive Features (MEF)* of the local eigenspaces for each single fuzzy controller.

Another question is the size of the visual area to receive good interpolation results. Because the amount of memory needed for the local eigenspaces directly relates to the size of the eigenvectors it would be best to make the input images as small as possible. If the images are too small, major distinctive features are lost. An automatic adaptation to the best size is an important objective. Further it is feasible to replace the crisp situation multiplexing by a soft-switching controller. As final application it is also necessary to combine this localisation approach with other mobile robot tasks, like collision avoidance, automatic mapping, and object recognition and tracking.

## 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. Cassinis *et al.*, Using Colour Information in an Omnidirectional Perception System for Autonomous Robot Localisation. Proceedings of EUROBOT'96, pp. 172-176.
- [3] J.A. Hancock and C.E. Thorpe, ELVIS: Eigenvectors for Land Vehicles Image System, CMU-RI-TR-94-43.
- [4] J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice Hall, 1997.
- [5] E. Oja. *Subspace methods of pattern recognition*. Research Studies Press, Hertfordshire, 1983.
- [6] 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.
- [7] 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.
- [8] 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.