

A Generic Learning Approach to Multisensor Based Control

Yorck von Collani, Jianwei Zhang and Alois Knoll
 Technical Computer Science, Faculty of Technology,
 University of Bielefeld, 33501 Bielefeld, Germany

Abstract— We propose a concept for integrating sensors in real-time robot control. To increase the controller robustness under diverse uncertainties, the robot systematically generates series of sensor data (as robot state) while memorising the corresponding motion parameters. Based on the collection of (multi-) sensor trajectories, statistical indices like principal components for each sensor type can be extracted. If the sensor data are pre-selected as output relevant, these principal components can be used very efficiently to approximately represent the original perception scenarios. After this dimension reduction procedure, a non-linear fuzzy controller, e.g. a B-spline type, can be trained to map the subspace projection into the robot control parameters. We apply the approach to a real robot system with two arms and multiple vision and force/torque sensors. These external sensors are used simultaneously to control the robot arm performing insertion and screwing operations. The successful experiments show that the robustness and the precision of robot control can be enhanced by integrating additional sensors using this concept.

Keywords— neuro-fuzzy control, learning, B-splines, on-line sensor fusion

I. INTRODUCTION

In our research work on sensor-based robot control [10], we are faced with many high-dimensional problems concerning a large number of input variables which importance and interdependence are not clearly known. It is well-known that general rule descriptions of systems with a large number of input variables suffer from the problem of the “curse of dimensionality”. But in many real-world applications it is difficult to identify the decisive input parameters and thus to reduce the number of input variables to the minimum. Hence a general solution to building control models is not only interesting from a theoretical point of view, it may also extend the range of applications of intelligent control to more complex intelligent control problems.

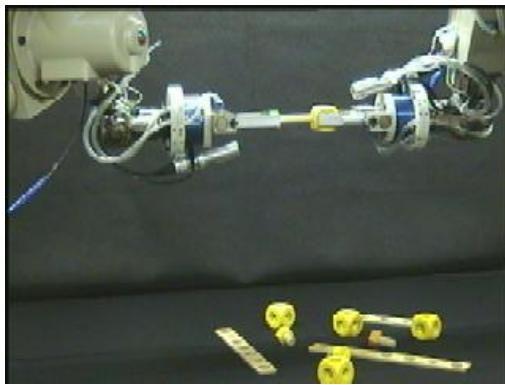


Fig. 1. The experimental setup for assembly.

A. Vision/Force-Guided Robot Motion

Assembly skills like screwing are part of the most important and most demanding sensor-based manipulation skills of coop-

erating robots. A special case of vision-action transformation is camera-supported fine-motion control. *Affine Visual Servoing* [5] may be applied to such tasks. The changes in the shape of image contours are the input of a feed-forward controller. Another interesting approach was proposed in [18]: visual servoing for positioning with an uncalibrated camera system. However, it requires a special test motions for performing well.

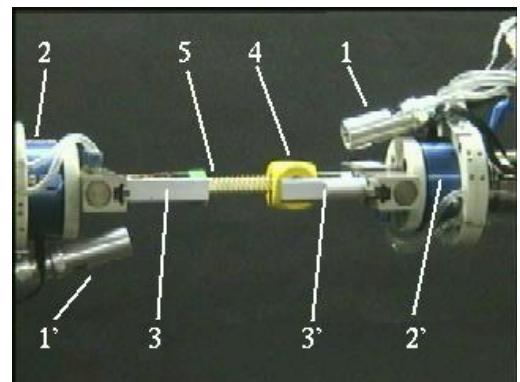


Fig. 2. 1,1': hand-camera; 2,2': force / torque sensor; 3,3': parallel jaw-gripper; 4: nut; 5: screw.

The use of force feedback is the mostly used sensor information source in robotics. In recent years, visual feedback and especially the integration of both had been of great interest. Conventional techniques try to exploit a common representation space and achieve a fused model of the environment [12], [4]. In [6] this is achieved by describing sensor observations in terms of uncertain geometry using probabilistic fusion methods. In [1] vision together with an internal strain gauge is used to gather information about the contact forces acting on a hand during grasping. In [15] force and vision feedback are combined using the so-called *vision and force resolvabilities*. With an approach presented in [2], the force and vision information is fused by using a task frame formalism. As an example a vision algorithm reconstructs the 3D position of a feature point, using also the distance information from a force sensor. Nearly all these approaches need an explicit modeling of the sensor properties in order to combine the information. In our work presented here, we fuse visual information from one or two uncalibrated cameras as well as from one camera and a force/torque sensor. Instead of any explicit models, we employ an adaptive neuro-fuzzy scheme to *learn* the appropriate robot motions necessary to perform a complex screw task.

In recent years, using uncalibrated cameras for visual guidance becomes an interesting topic [18], [8]. CMAC neural networks may tackle the problem of dimensionality. In [13] 12

inputs represent four joint positions of the robot, four image parameters and their desired changes. The outputs are the control signals for the four robot joints. In [14], learning of vision-based positioning based on visual appearance information was introduced. The image data set is compressed using *principal component analysis* to obtain a low-dimensional input space. A parametric eigenspace representation is used for describing the different objects as well as object locations. The positioning problem is thus transformed into finding the minimum distance between a point and a manifold in the eigenspace. As far as we know, no work on mapping the multiple images and force/torque data directly into action values has been reported.

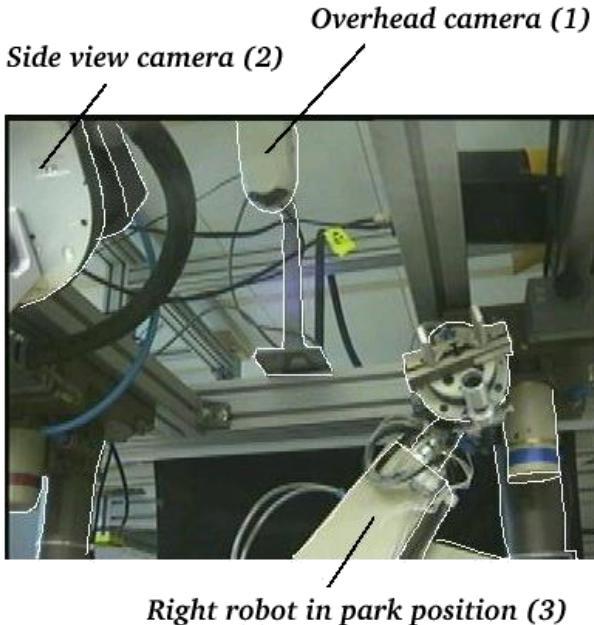


Fig. 3. Location of the cameras: (1) overhead, (2) side-view, (3) robot in park position.

B. Existing Solutions to Multivariate Problems

Two main methods to solve the problem of input dimension reduction are *input selection* and *hierarchy*. Input selection, e.g. [9], [3], is an experimental method to find the most important input variable in a large set of input variables. With this approach, all the combinatorial possibilities of the low-dimensional fuzzy model are considered and approximately tested. The inputs which result in the best outputs are viewed as the most important ones to build an exact neuro-fuzzy model. The problems with this method are the loss of information and the number of combinations that must be tested.

Hierarchical structuring assumes that the input information can be classified into groups, see [11] for an example. There is, however, no general approach to realise such a grouping.

II. PROBLEM DESCRIPTION

A. Experiment Setup

The performance of our approach is demonstrated by a screw operation (Fig. 1 and 2). Screwing a screw (Fig. 2(5)) into a nut (Fig. 2(4)), originates from our interdisciplinary collaborative project SFB 360 which aims at assembly of aggregates with

wooden toy construction sets. The manipulators are installed upside down and can grasp the required assembly components from the assembly table. Each robot is equipped with a force sensor (Fig. 2(2,2')) on which a pneumatic parallel-jaw gripper (Fig. 2(3,3')) is mounted. A small camera (Fig. 2(1,1')) is fixed over the gripper. The manipulators are two Puma 260, and the host computer is a Sun UltraSPARC. We consider general inserting and screwing without using any fixture devices.

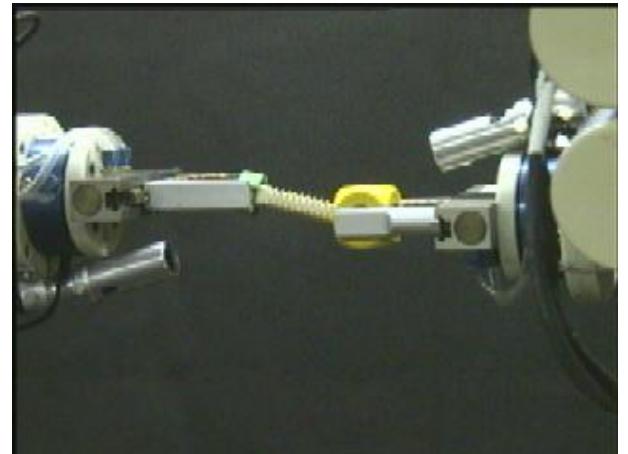


Fig. 4. An inconvenient start-situation for screwing.

B. Uncertainties

For a general-purpose arm/gripper system, the following two types of uncertainties must be taken into account:

- Grasping precision: Although we have applied a hand camera in a “self-viewing” configuration, which significantly improved the grasping precision in comparison with the open-loop positioning, regrasping still engenders deviation of the screw from the rotation axis of the gripper.
- Slippage of the part in the hand: Due to the effect of the resulting forces, the screw grasped by a jaw gripper may easily slip during the screwing process.

The uncertainties cause the following two concrete problems in a screwing process:

1. The screw is not centrically grasped: the rotating axis of the screw does not match the axis of the gripper.
2. The screw is obliquely grasped, see Fig. 4.

III. VISION-BASED CONTROL

Without using sensors the screw operation can fail under each of the uncertainties discussed above. Therefore, sensor-based compensation motions become necessary.

The resulting forces in the normal and orientation directions should be minimised and stable. Additionally, to guarantee a successful screwing-in phase, a constant force in the approach direction should be exerted. The forces and/or torques give no sufficient information about the orientation of the screw. Here supplementary approach is to monitor the scene with external cameras and correct the orientation before contact is made between the screw and the nut.

Now we have performed multiple approaches to adjust the orientation of the screw after a contact is made:

1. We fuse the images from two different cameras to determine the orientation of the screw (Fig. 7).
2. We fuse the information of a force/torque sensor and the related camera [16].

The displacement of the screw in the gripper is the output of the controller and hence the output can be used directly to correct the manipulator. Fig. 5 shows a sequence of typical views of the scene during the screw operation. We therefore employed a method that extracts automatically the needed features from one or two fused images to compensate the uncertainties.

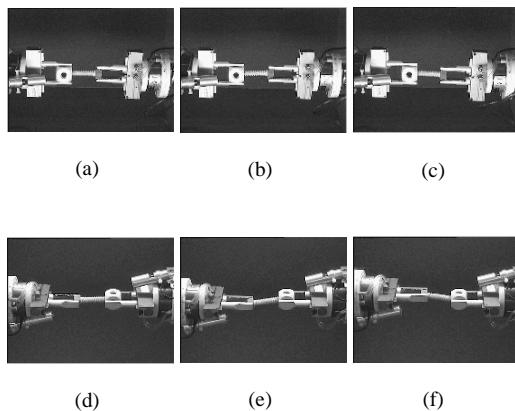


Fig. 5. Typical images taken by the external cameras (a)-(c) viewpoint from above, (d)-(f) side view.

IV. B-SPLINE NEURO-FUZZY MODEL

A. The B-Spline Controller Model

The controller can be efficiently realised using the B-spline fuzzy controllers proposed in our earlier work [19]. This type of controller may be characterised by the following features distinguishing it from standard fuzzy controllers:

- By choosing the order n of the basis functions, the output is C^{n-2} continuous. However, too high an order will bring computational burden and additional rules. In practice, order 2, 3, 4 are suitable for modelling membership functions.
- Each controller output is defined by a set of fuzzy singletons (control vertices). The number of control vertices is equal to the number of the rules and their optimal values can be iteratively found through learning. This adaptation procedure is equivalent to weight adjustment in an Associated Memory Neural Network.
- One problem with learning in conventional fuzzy controllers is that too many parameters must be adjusted. With B-spline fuzzy controllers, a simple modification of control vertices causes the change of the control surface. As far as concerned supervised learning, the partial differential with respect to each control vertex is a convex function, if the square error is selected as the quality measure. As for unsupervised learning, the learning-process descent will also show stable asymptotic behaviour [19], if the error of the cost function is approximately piecewise proportional to the error of the control values.

In our earlier work, we showed the advantages of this learning controller approach. The robot controller learns actively and online to control the compensation motion according to measured

forces and is quite simple to design. The learning process converges rapidly, the output is smooth if B-spline functions of order 3 or higher are used (C^{k-2} continuous, where k is the order of the B-spline).

B. Dimension Reduction

It is one of the long-term research goals to find a general model which transforms raw image data directly into action values. Our grey-scale images have 101×41 pixels and if no image processing is performed then a control system with about 4,000 input variables (i.e. one for each pixel) needs to be modeled; the system output would be the motion values for the robot(s).

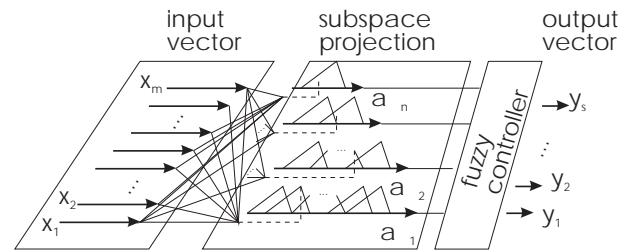


Fig. 6. The structure of a fuzzy controller based on subspace projection.

If the dimension of the input space is small enough, the input variables can be directly covered by fuzzy sets. Each item of the rule is human readable and may be interpreted as describing a special instance of a general situation. However, if the image of a camera is regarded as a vector, this high-dimensional sensor image is too large to build a corresponding rule base. Fortunately, sensor images are often observed in a local context and a subspace containing all necessary information for determining the action values can be found.

C. Projection into Subspace

C.1 Principal Component Analysis

A well-known technique for dealing with multivariate problems in statistics is the *principal component analysis* (PCA). As shown in [14], this technique is also suitable for reducing the dimension of the input space of a general control problem. It was introduced for the use of visual learning by [17]. An eigenvector, denoted as \vec{a}_i , is a linear combination of all the original input variables

$$\vec{a}_i = a_{1,i}x_1 + a_{2,i}x_2 \dots a_{m,i}x_n$$

where $a_{i,j}$ is the coefficient of the input variable x_j , $j = 1, \dots, n$. \vec{a}_i accounts for one dimension in the eigenspace. The eigenvectors form an orthogonal basis for representing the original individual sensor patterns.

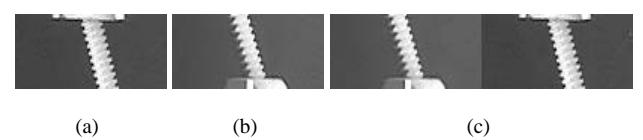


Fig. 7. Clipped images from camera 1 (a) and camera 2 (b) and the resulting merged image (c).

Depending 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 seven or eight eigenvectors supply the most information indices of the original input space.

The eigenvectors and eigenvalues of a covariance matrix can be computed by the Jacobi method [7]. Some algorithmic improvements enable the high-dimensional computation. In the example of Fig. 8, the computed eigenvectors are shown.

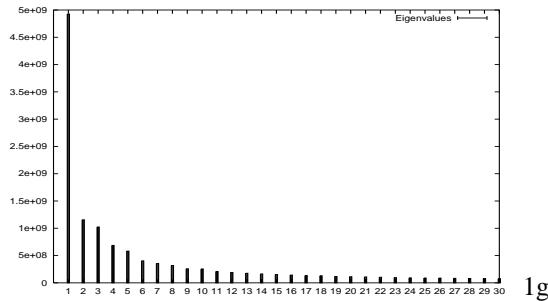


Fig. 8. Sorted eigenvalues of covariance-matrix.

Assume that the eigenvectors $\vec{a}_1, \vec{a}_2, \dots$ 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. \vec{a}_i defines the i th dimension in the eigenspace. The projection of an input vector

$$X = [x_1, x_2, \dots, x_m]^T$$

onto eigenvector \vec{a}_i , called the i th principal component, is $p_i = a_{1,i}x_1 + a_{2,i}x_2 + \dots + a_{m,i}x_m$. The complete projection can be represented as:

$$[\vec{a}_1, \dots, \vec{a}_n]^T \cdot X = [p_1, \dots, p_n]^T$$

All projections of the sample data sequence form a manifold in the eigenspace. Such a projection can be viewed as a layer of neural network, see the connection layer of the two left parts of Fig. 10.

In Fig. 9 we show a result of such an image projection into the eigenspace and the visualised transform matrix of the fused image data. The brighter the pixel, the more relevant the component in the image.

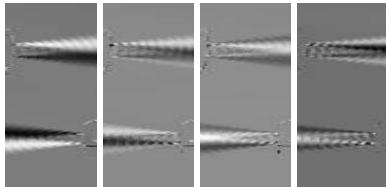


Fig. 9. Visualisation of the transformation matrix: first to fourth principal component.

D. Output Relevant Features

The dimension reduction with PCA has the problem that there is no correlation between the input and the output of the system.

Only the variance is taken into consideration. If the needed information is not in the variance, many eigenvectors are needed, which results in a high-dimensional B-spline controller.

Another approach is to find a set of vectors which have a direct correlation between the input and the output data. The set of vectors is called *Output Relevant Features* (ORF) [20].

To find the ORF vectors, a perceptron is learned by supervised learning. The outputs of the perceptron are a projection of the input data onto the feature vectors. The learning rule minimises the difference between the desired and the actual values:

$$\Delta w_j = \eta(Y_S - V)\xi_j$$

The perceptron itself is defined as $V = \sum w^T \xi$. w is the weight matrix and ξ is the input vector.

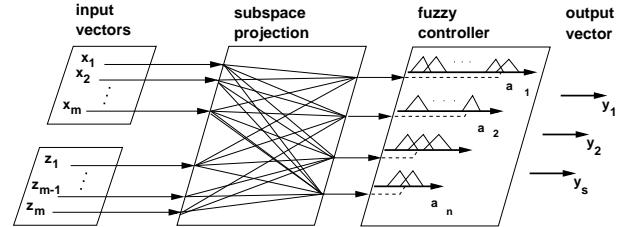


Fig. 10. The structure of a fuzzy controller based on subspace projection by fusing images.

E. The Fusion Approach

Our approach of dealing with 3D uncertainties is to project local parts of a grey-scale camera image into a subspace (Fig. 6). We have two different approaches:

1. We merge small and local parts of different grey-scale camera images and project the resulting image into an eigenspace (Fig. 10).
2. We project the images from one camera into a subspace and take this and the data from a force/torque sensor as input for our controller (Fig. 11).

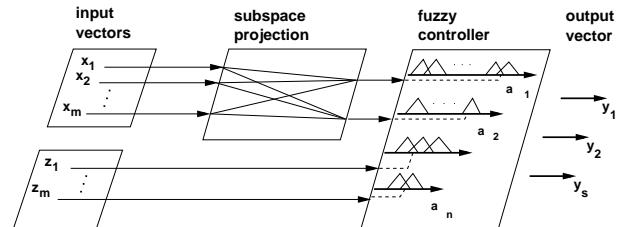


Fig. 11. The structure of a fuzzy controller for fusing force and vision.

V. IMPLEMENTATION ISSUES

A. Sampling Training Data

For training, the input data and desired output values have to be recorded. It is desirable that all typical input data are generated. As outlined above there are different orientations of the screw. For recording, the robot moves to the ideal orientation for screwing. Subsequently, it moves to several other orientations. The deviations from the ideal orientation are recorded for each displacement. For screwing the ideal orientation, the forces and torques are recorded.

B. Calculating Projection Vectors

After the input data are sampled, the following steps are necessary:

1. The (potentially merged) images are normalised so that the energy of each image becomes one. Then the average image can be subtracted.
2. The images are stacked into vectors.
3. If using PCA, the covariance matrix of the vectors is calculated and the eigenvectors and eigenvalues are calculated. By using ORF, the projection vectors are iteratively determined.
4. Each image is projected into the subspace.

C. Training the Fuzzy Controller

For the B-spline controller the training procedure is as follows:

1. Select the n projection vectors with the largest n eigenvalues (PCA) denoted as $\vec{a}_1, \dots, \vec{a}_n$ or with the largest correlation (ORF).
2. For the second task (Fig. 11) select the components from the force/torque vector which have the greatest relevance to the desired controller output. In this application these are the forces in N- and O- direction and torques around the N- and O- vector of the robot tool, because the rotation of the gripper is around the approach vector.
3. Select the order of the B-spline basis function for each input.
4. Determine the knots of the B-spline basis functions for partitioning each input.
5. Project images onto the selected vectors.
6. Initialise the control vertices for the output.
7. Learn the control vertices with the projected values from the images and the data from the force/torque sensor using the gradient descent method.
8. If the results are satisfying, terminate.
9. Modify the knots for vectors, go to 5.

It is important to determine the right parameters for the fuzzy set. If too few eigenvectors are used, then the fuzzy controller cannot distinguish all situations. If too many vectors are used, then the memory requirements of the fuzzy set and the number of required training samples are not manageable. Similarly, the correct partition of each input variable is important. If the partitioning is too fine, the fuzzy controller generalises insufficiently.

VI. NUMERICAL RESULTS

A. Fusing two cameras

After merging the two images and projecting them into the reduced eigenspace, the four largest eigenvectors are used as input for the B-spline fuzzy controller. The first eigenvector is covered with 10 B-splines as membership functions, the second with 9, etc.

In comparison, Table I shows the mean square error of the controller, the maximum error and the largest worst case errors of the test images around the N- and O-direction with and without fused images. The worst case is defined as a movement of the manipulator in the wrong direction, which causes a larger displacement.

Our first experiment shows that merging the images produces a much better controller than the separated controller for each

	Mean square error [° ²]	Maximum error [°]	Worst case error [°]
N- direction			
Overhead camera	8.81	11.14	7.63
Side-view camera	54.41	29.85	29.85
Fused images	0.69	3.14	—
O- direction			
Overhead camera	53.24	26.74	26.74
Side-view camera	10.47	12.75	12.75
Fused images	0.92	4.04	—

TABLE I
MEAN SQUARE ERROR, MAXIMUM ERROR AND WORST CASE ERROR FOR ANGLE AROUND N-AND O-DIRECTION.

	Mean square error [° ²]	Maximum error [°]	Worst case error [°]
N- direction			
Overhead camera	1.98	5.73	—
Cameras + torque	1.91	8.22	—
O- direction			
Side-view camera	2.85	6.19	—
Cameras + torque	2.63	6.14	—

TABLE II
MEAN SQUARE ERROR, MAXIMUM ERROR AND WORST CASE ERROR FOR ANGLE AROUND N- AND O-DIRECTION COMBINING FORCE AND VISION.

camera and direction. The mean square error and maximum error are smaller and with the merged images in contrast there is no output of the controller which moves the manipulator in the wrong direction. This approach results in a very robust and rapid technique to correct the orientation of the screw.

A.1 Combining visual and force/torque information

Similar to the first task we also use the four largest eigenvectors from a single image as input for the B-spline fuzzy controller. Additionally, the torque around the N- and around the O-axis of the tool is used as input (Fig. 11). The eigenvectors are covered with 10 to 7 B-splines and the torque is covered with 5 B-splines as membership functions. Table II also shows the mean square error of the controller, the maximum error and the largest worst case errors of the test images around the N- and O-direction with additional forces. This experiment shows that the fusion of vision and force/torque data produce better results in comparison with the unfused case. It also shows that the B-spline neuro-fuzzy model is capable of fusing different sensor data.

With ORF as dimension reduction method, we only need one weight vector and consequently a one-dimensional B-spline controller. Table III shows the numerical results with two fused images. The controller has one dimension and this dimension is covered by 10 B-splines. Comparing Table II and Table III shows that the result with ORF is better than with reduction by

	Mean square error [\circ^2]	Maximum error [\circ]	Worst case error [\circ]
N- direction	2.72	4.48	—
O- direction	2.64	4.08	—

TABLE III

MEAN SQUARE ERROR, MAXIMUM ERROR AND WORST CASE ERROR FOR ANGLE AROUND N-AND O-DIRECTION WITH FUSED IMAGES (ORF).

PCA. The mean square error is bigger, but the maximum error is lower in comparison to the reduction with PCA and only a one-dimensional controller is used. Fig. 12 shows the visualised transform matrix in the same manner as Fig. 9. These figures show the difference between PCA and ORF. The PCA method considers both views of the scene whereas the ORF method prefers one view (right half resp. left half of the image).

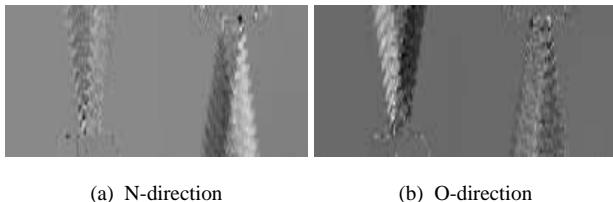


Fig. 12. Visualisation of the output relevant features.

VII. CONCLUSIONS

We have shown that the proposed neuro-fuzzy model can be utilised for sensor fusion and high-dimensional problems such as visually guided fine-motion. The dimension reduction with PCA should be used if the needed information is in the variance of the sensor data. This is, for example, the displacement of an object in an image. The efficient dimension reduction with ORF is also possible, but its computation cost is higher than with PCA. In other cases or if unknown data is used, ORF should be used to reduce the input space. Additionally, data from different sensor types should not be merged before reduction (Fig. 11). The calculation of the PCA fails due to the different characteristics of different sensor data.

The advantages of our approach are:

- Projecting the high-dimensional input space into a reduced subspace the most significant information for control is maintained. A limited number of transformed inputs can be partitioned with the B-spline model.
- By merging the different kinds of sensor data a sufficient precision can be obtained for determining the robots orientation correction.
- To solve this problem the statistical indices provide a suitable solution to describe the information in images with a lot of uncertainties.
- A vector in the subspace is directly mapped onto the controller output based on the B-spline model. This makes real-time computation possible.

- Designing the controllers is simple and identical for both low and high dimensional controllers. Both force and vision controllers are of the same type. The B-spline fuzzy controller can be trained in a straightforward manner because modification of control vertices only results in local change of the control surface.

In this approach no complex programming and knowledge about vision and force control is needed. We have shown that this approach is very promising for realising efficient robot assembly skills based on sensorimotor coordinations.

REFERENCES

- [1] P. K. Allen, A. T. Miller, P. Y. Oh, and B. S. Leibowitz. Integration of vision and force sensors for grasping. In *Proc. IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems*, 1996.
- [2] J. Baeten, Herman Bruyninckx, and J. De Schutter. Combining eye-in-hand visual servoing and force control in robotic tasks using task frame. In *IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 1999.
- [3] S. L. Chiu. Selecting input variables for fuzzy models. *Journal of Intelligent and Fuzzy Systems*, 4:243–256, 1996.
- [4] J. L. Clark and A. L. Yuille. *Data Fusion for Sensory Information Processing Systems*. Kluwer Academic Publishers, 1990.
- [5] C. Colombo, B. Allotta, and P. Dario. Affine visual servoing: A framework for relative positioning with a robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 464–471, 1995.
- [6] Hugh F. Durrant-Whyte. *Integration, Coordination and Control of Multi-Sensor Robot Systems*. Kluwer Academic Publishers, 1988.
- [7] J. Greenstadt. The determination of the characteristic roots of a matrix by the jacobi method. Technical report, IBM Report, 1961.
- [8] M. Jägersand, O. Fuentes, and R. Nelson. Experimental evaluation of uncalibrated visual servoing for precision manipulation. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 2874–, 1997.
- [9] J. S. R. Jang, C. T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice Hall, 1997.
- [10] A. Knoll, B. Hildebrandt, and J. Zhang. Instructing cooperating assembly robots through situated dialogues in natural language. In *IEEE International Conference on Robotics and Automation*, 1997.
- [11] V. Lacrose and A. Tilti. Fusion and hierarchy can help fuzzy logic controller designers. In *IEEE International Conference on Fuzzy Systems*, 1997.
- [12] G. T. McKee. What can be fused? In J. K. Aggarwal, editor, *Multisensor Fusion for Computer Vision*, NATO ASI Series. Springer-Verlag, 1993.
- [13] W. T. Miller. Real-time application of neural networks for sensor-based control of robots with vision. *IEEE Transactions on System, Man and Cybernetics*, 19:825–831, 1989.
- [14] 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.
- [15] B. J. Nelson and P. K. Khosla. Force and vision resolvability for assimilating disparate sensory feedback. *Transaction on Robotics and Automation*, 12(5):714–, 1996.
- [16] Y. v. Collani, C. Scheering, Z. Zhang, and A. Knoll. A neuro-fuzzy solution for integrated visual and force control. In *IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 1999.
- [17] Y. v. Collani, Z. Zhang, and A. Knoll. A neuro-fuzzy solution for fine-motion control based on vision and force sensors. In *IEEE International Conference on Robotics and Automation*, 1998.
- [18] B. H. Yoshimi and P. K. Allen. Active, uncalibrated visual servoing. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 156–161, 1994.
- [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.
- [20] J. Zhang, A. Knoll, and V. Schwert. Situated neuro-fuzzy control for vision-based robot localisation. *Robotics and Autonomous Systems*, 28:71–82, 1999.