
Object Recognition and Visual Servoing: Two Case Studies of Employing Fuzzy Techniques in Robot Vision

Alois Knoll, Jianwei Zhang, Thorsten Graf, and André Wolfram

Faculty of Technology, University of Bielefeld, Germany

Summary. The capabilities of observing the world and manipulating objects based on visual information are basic requirements in robotic applications. Typically, for manipulating objects in real environments it is necessary to recognise and locate the substantial objects robustly. Due to external influences, e.g. partial occlusions of objects and illumination changes, as well as to internal influences, e.g. noisy imaging hardware, inaccurate measurements and quantization effects, the recognition systems have to cope with incomplete, uncertain and inaccurate information.

In the first part of this paper we present the general framework of a robust approach for recognising partially occluded objects. It combines the popular concept of using invariant shape descriptions, i.e. descriptions of objects which remain unaffected by certain variations of the intrinsic and extrinsic camera parameters, with the flexibility and readability of rule-based fuzzy systems by applying invariant object shape descriptions in fuzzy if-then classification rules.

In the second part, we propose a fuzzy control approach for learning fine-positioning of parallel-jaw robot gripper using visual sensor data. The first component of the used model can be viewed as a perceptron network that projects high-dimensional input data into a low-dimensional eigenspace. The second component is a fuzzy controller serving as an interpolator whose input space is the eigenspace and whose outputs are the motion parameters. Instead of undergoing cumbersome hand-eye calibration processes, our system is trained in a supervised learning procedure using systematical perturbation motion around the optimal grasping pose.

1 Introduction

Vision may aid robots (both mobile and fixed) in many tasks such as navigation, location and inspection of assembly parts, tracking objects, avoiding collisions, detecting errors, determining distances or even learning from a human instructor by analysing hand gestures. Unfortunately, unlike human vision, machine implementations of vision systems tend *not* to be particularly robust, i.e. they are susceptible to variations in lighting, to unexpected shadows, lens and perspective distortions, specular reflections etc. Moreover, these systems often cannot deal with more complex scenes in which objects occlude each other or in which parts change their orientation with respect to a reference image.

These uncertainties seem to lend themselves naturally to be dealt with by fuzzy techniques. Interestingly, such techniques have not met with much acceptance in the computer vision community even though they promise to alleviate two of the main methodological problems in computer vision: the ubiquitous application of crisp thresholding and (probabilistic) model matching (see [11,27])¹.

Both thresholding and model matching result in a more or less drastic loss of information that is not recoverable in higher processing stages of a vision system – data are irretrievably discarded either when they do not exceed a certain threshold or do not lie within a certain distance to the model. The obvious advantage of fuzzy techniques is that they do not necessarily discriminate sharply between data that are below and above the threshold. Instead, the threshold can be replaced by a membership function and valuable information retained for higher level processing stages. The same applies to models (of object form, pose, invariants, etc.) and clustering: while (probabilistic) models are normally based on Bayesian prior distributions with few degrees of freedom, fuzzy methodology easily combines the notion of membership with complex rule bases that make it easy to express and incorporate human expert knowledge. It is also underpinned by powerful learning or adaptation algorithms whose learning results are transparent, sometimes even human readable.

Examples of typical complete vision systems that have successfully incorporated fuzzy techniques are:

- path tracking of autonomous indoor vehicles through fuzzy control [4]. The main advantage of the fuzzy controller was considered to be its ability to extract heuristics from experiences. An outdoor mobile platform that is also controlled by a fuzzy rule base is presented in [17].
- visual inspection of assembly parts (integrated circuits) for their correctness with respect to their specification [6]. It was noted that both the fuzzy algorithm used there and expert systems represent human domain knowledge in the form of production rules; the former, however, does not run the risk of a combination explosion.
- identification of objects in a robotics scene based on a verbal description [8]. The noisy high-level features of an object are compared with a natural language expression.
- object extraction for navigation through fuzzy feature matching [14]. The main goal here is to recognise office chairs under varying viewing angles or in the case of occlusions by evaluating local features. Similarly, [15] use corners as local features for the recognition of transistor packages on an assembly line.

¹ These two papers contain references to other survey papers; [23] discusses some further issues related to uncertainty.

- recognition of occluded objects [25] using fuzzy linguistic statements as input for the training of a neural network. The classification of the partially hidden objects is then performed based on adequate local features.

In this paper we address two issues in robot vision: the recognition of partially occluded objects in complex scenes of (known) objects and the fine-positioning of a robot gripper for grasping an object. We demonstrate the performance of our methods for a specific domain: a set of the wooden toy objects of a construction kit for children (Fig. 1, for details see [13]). Our experiments show that a robust recognition of occluded objects is possible even under a high degree of uncertainty (object type, object pose, highly structured background, change of perspective, lighting variations). The same is true for our “near-field” recogniser that guides the robot towards its gripping position.

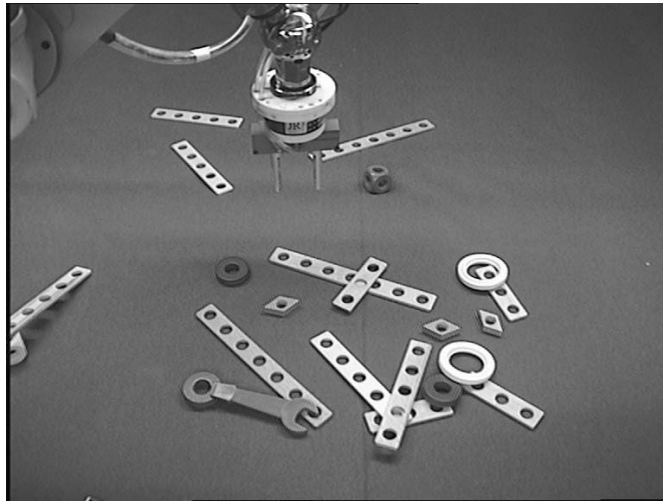


Fig. 1. Typical test scenario

The rest of the paper is organised as follows: in section 2 we describe in detail a recognition system for the extraction of objects from images taken of a robotics scene. The images are recorded by a standard camera with a large area of view. Section 3 is devoted to the hand-camera based fine-positioning system, which “takes over” once the position of the objects in the scene have been determined and the gripper has been moved into the vicinity of the object to be grasped. Section 4 describes the whole “run” from the first image of the scene to the grasping. Finally, in section 5, we conclude with some remarks on the qualities and deficiencies of the systems and possible lines of future research.

2 Recognition through Fuzzy Invariant Indexing

2.1 Motivation

The recognition of partially occluded objects is undoubtedly one of the most challenging tasks in computer vision. Recent research has indicated that the use of invariants as shape descriptors is a promising and powerful approach to tackle this problem. Mathematically, invariants are functions of geometric configurations remaining unaffected under particular classes of transformations (for good introductory papers see [7,19]), e.g. the class of projective transformations modelling the camera mappings of a vision system. Since these invariants are independent of the viewpoint of the camera, the measured projective invariant values of an object can be used efficiently in the hypothesis generation as an index into an object-lookup table. This technique is called invariant indexing.

Several recognition systems based on invariant theory have been developed, e.g. an early system based on geometric hashing technique [16], the LEWIS-system [24] or the MORSE-system [18]. Since images taken from real-world scenes (and using real-world equipment) are generally discrete, cluttered and noisy, the observed projective invariant values fluctuate when different perspective views of an object are recorded. This problem must be handled within the indexing stage of every recognition system based on invariants. In our context, indexing means to assign image features to corresponding model features and therefore to generate object hypotheses.

Usually, invariant indexing hashes into a discrete index space, where all points belonging to an object are marked. For indexing, a hashing function is evaluated for the measured invariant values of a geometric configuration part of an object. The number of the independent invariants depends on the underlying geometric configuration. For example, the planar geometric structure of a conic and three straight lines has three independent projective invariants (see sect. 2.2). To overcome the fluctuation, not only a single point of the index space is marked but also the neighbouring ones. Invariant values of a certain neighbourhood are hence mapped to the same object with equal weight.

Contrary to this, we utilize an invariant indexing technique based on fuzzy if-then-classification-rules and fuzzified invariant values modelling the fluctuation. This technique is called Fuzzy Invariant Indexing (FII).

2.2 The Fuzzy Invariant Indexing Technique

The basis of the fuzzy invariant indexing technique are disjunctively connected fuzzy if-then-rules that incorporate fuzzified invariant values. These classification rules have the following form:

$$\text{IF } i_{m1}^k = \tilde{I}_{m1}^k \text{ AND } \dots \text{ AND } i_{mN_m^k}^k = \tilde{I}_{mN_m^k}^k \text{ THEN } o_m^k = \tilde{O}^k \quad (1)$$

$$\begin{aligned} & k=1, 2, \dots, K \\ \text{with } & m=1, 2, \dots, M^k \\ & n=1, 2, \dots, N_m^k \end{aligned}$$

where i_{mn}^k denotes the n -th input variable of subrule m for the k -th object, \tilde{I}_{mn}^k the corresponding fuzzy invariant value, o_m^k the output variable of subrule m and \tilde{O}^k the k -th object class modelled as a fuzzy singleton. The total amount of antecedents (N_m^k) depends on the number of independent invariants of the underlying geometric configuration of subrule m for object k and the total amount of subrules (M^k) depends on the number of different geometric configurations for object k .

The main problem with the fuzzy if-then-rules is to find appropriate membership functions to model the fuzzy invariant values \tilde{I}_{mn}^k in (1) for a given object. Our investigation of invariant values measured in views of different perspectives has indicated that the fluctuations can be adequately approximated by bell-shaped membership functions $\mu_{\tilde{I}_{mn}^k}$:

$$\mu_{\tilde{I}_{mn}^k}(u) = e^{-\frac{(u - \alpha_{mn}^k)^2}{2\beta_{mn}^k}}, \quad u \in \mathbb{R} \quad (2)$$

where the parameters $\alpha_{mn}^k, \beta_{mn}^k$ that shape the function are chosen as follows:

- The parameter α_{mn}^k determines the position of the maximum of the bell-shaped function (2). Therefore, this parameter should be the mean of the fluctuating invariant values: $\alpha_{mn}^k = \frac{1}{N} \sum_l I_l$, where I_l , $1 \leq l \leq N$ are the invariant values for an object taken from N different views.
- The parameter β_{mn}^k determines the position of the inflexions of (2), which are located at $\alpha \pm \beta$. This parameter should be the standard deviation of the invariant values: $\beta_{mn}^k = \left(\frac{1}{N} \sum_l (I_l - \alpha_{mn}^k)^2\right)^{\frac{1}{2}}$.

For example, consider the well-known invariant of a conic and two lines under plane projective transformations [19]:

$$I(\mathbf{C}, \mathbf{l}_1, \mathbf{l}_2) = \frac{(\mathbf{l}_1^t \mathbf{C}^{-1} \mathbf{l}_2)^2}{(\mathbf{l}_1^t \mathbf{C}^{-1} \mathbf{l}_1) (\mathbf{l}_2^t \mathbf{C}^{-1} \mathbf{l}_2)} \quad (3)$$

where \mathbf{C} is the conic coefficient matrix and $\mathbf{l}_1, \mathbf{l}_2$ denote lines expressed in homogeneous coordinates. Figure 2 shows the distribution of invariant values for a test object 'nut' measured in 30 images. To provide a better discrimination we apply Eq.(3) to feature groups of a conic \mathbf{C} and three lines $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$

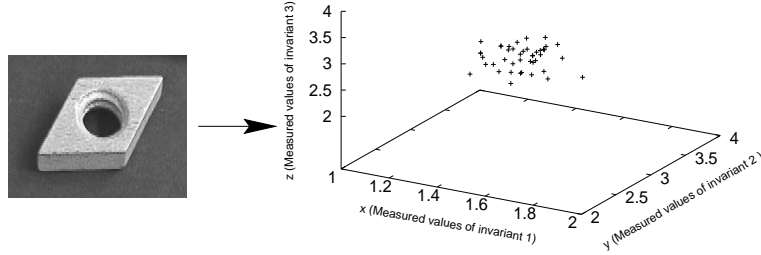


Fig. 2. Distribution of measured invariant values for test object 'nut' (30 images)

by determining the three independent invariant values:

$$I_1(\mathbf{C}, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3) = I(\mathbf{C}, \mathbf{l}_1, \mathbf{l}_2) \quad (4)$$

$$I_2(\mathbf{C}, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3) = I(\mathbf{C}, \mathbf{l}_1, \mathbf{l}_3) \quad (5)$$

$$I_3(\mathbf{C}, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3) = I(\mathbf{C}, \mathbf{l}_2, \mathbf{l}_3) \quad (6)$$

In Fig. 2, the invariant values on the x -axis are calculated using Eq. (4), the invariant values on the y -axis are calculated using Eq. (5) and the invariant values on the z -axis are calculated using Eq. (6). The distributions of these invariant values are depicted in the histograms shown on the left hand side of Figure 3. Thus, for the test object 'nut' we get the values $\alpha_{11}^k = 1.4$, $\beta_{11}^k = 0.086$ for the parameters of the first fuzzy invariant value, $\alpha_{12}^k = 3.2$, $\beta_{12}^k = 0.145$ for the second and $\alpha_{13}^k = 2.5$, $\beta_{13}^k = 0.14$ for the third fuzzy invariant value, which leads to the fuzzy invariant values shown on the right hand side in Figure 3. Note that the (fuzzy) model of an object may be composed of many such invariants. Finally, the fuzzy invariant values are incorporated into the following if-then-classification rule:

$$\begin{aligned} \text{IF (inv1} \approx 1.4) \text{ AND (inv2} \approx 3.2) \text{ AND (inv3} \approx 2.5) \\ \text{THEN (object is NUT)} \end{aligned} \quad (7)$$

Object hypotheses are generated through the FII-technique by evaluating the fuzzy rules using standard fuzzy inference techniques:

$$\mu_{o_m^k} := \min_{1 \leq n \leq N_m^k} \mu_{\bar{I}_{mn}^k}(i_{mn}^k) \quad (8)$$

where $\mu_{o_m^k}$ is the output of m -th subrule for object k and i_{mn}^k are the measured invariant values. These subresults are combined disjunctively:

$$\mu_{o^k} = \max_{1 \leq m \leq M^k} \mu_{o_m^k} \quad (9)$$

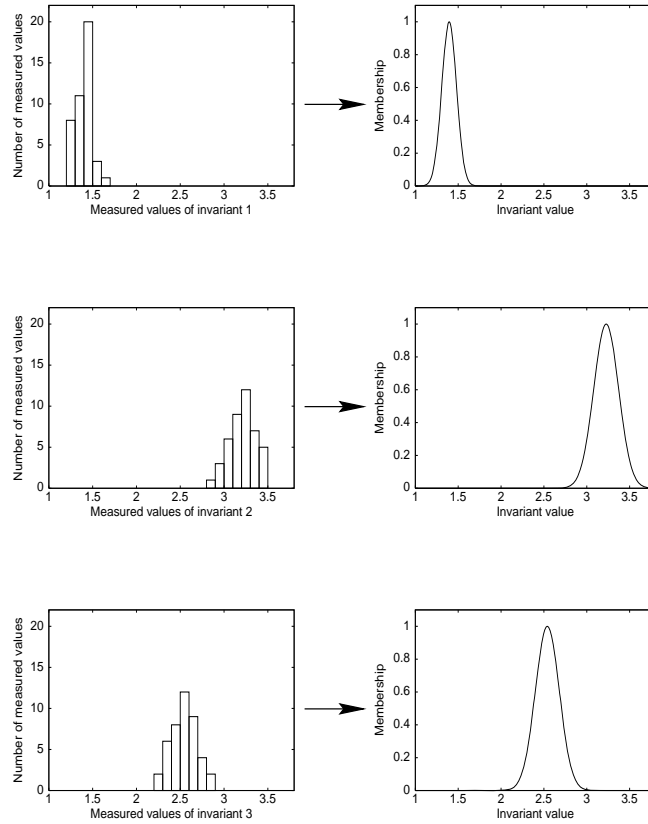


Fig. 3. Example for the generation of fuzzy invariant values

The final result is the indexed k -th object model with the measured credibility μ_o^k .

2.3 Recognition System

The fuzzy invariant indexing technique outlined above has been integrated into an object recognition system for partially occluded (quasi-)planar objects. The structure of the system is similar to other systems ([16,24]) but differs completely in applying a fuzzy rule base in the hypothesis generation stage.

As indicated in Tab. 1 the system provides two different processing phases: an off-line acquisition process and an on-line object recognition process.

Acquisition Process. The acquisition process is capable of learning the rules of the fuzzy invariant indexing automatically. The first stage of the acquisition process is the edge detection stage. In the implemented system we

Table 1. Acquisition / Recognition Process

	Acquisition (offline)	Recognition (online)
1	Edge detection	
2	Feature extraction	
3	Computation of invariants	
4	Generation of fuzzy invariant values	Hypothesis generation
5	Generation of fuzzy rules	Hypothesis verification

use the Canny edge detector [5], which takes a greyscale or color image as input, and generates as its output linked edge points. Next, in the feature extraction stage, geometric primitives are fitted to the extracted edge points, where the primitives that are used depend on the objects to be recognised. For the objects shown in Fig. 1 straight lines and ellipses are suitable. The features are grouped into configurations for which invariants can be computed. In the implemented system we use the invariants of two geometric configurations: the invariants of a conic and three lines (see Eqs. (4),(5),(6)), and the invariants of a pair of coplanar conics [21].

The last and most expensive stage is the model and rule generation, in which new objects are learned automatically, including both the object model and the fuzzy if-then classification rules. The object model is stored in a model base and consists of an object name, the extracted features and the computed invariant values. The fuzzy rules are generated as described in Section 2.2.

Recognition Process. The first three steps of the recognition process are equivalent to those of the acquisition process: the edge points are extracted in an image, features are fitted and invariants are calculated.

Next, in the hypothesis generation step, the classification rules of the fuzzy rule base are evaluated as described in Section 2.2. If the resulting credibility μ_{o^k} of an indexed object is above a threshold², a new object hypothesis is generated. This hypothesis consists of the object name, the credibility and the features used to compute the invariant values. The recognition process ends in the verification of the generated object hypotheses. This is done as usually: The hypothesized object model is mapped into the image and verified against the extracted features. Although this system is implemented for recognising (quasi-)planar objects only, this is no principle limitation.

2.4 Recognition Examples

The performance of the FII-recognition system is demonstrated for the aforementioned object domain of quasi-planar, colored wooden toy objects, such as rims, tyres, nuts and slats as shown in Fig. 1.

The first scene, Figure 4, consists of four three-hole-slats, two seven-hole-slats, two orange nuts, two red rims, two white tyres and one “unknown”

² The threshold is used to rule out the very unlikely hypotheses only.

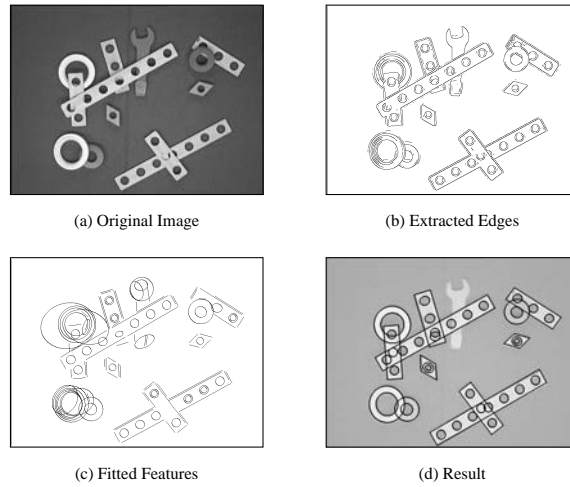


Fig. 4. Example scene 1

(i.e. unmodelled) spanner, which partially overlap each other. Since the extracted edge points (Figure 4b) as well as the fitted features (Figure 4c) provide a reliable image description, the system recognises all of the known objects (4d).

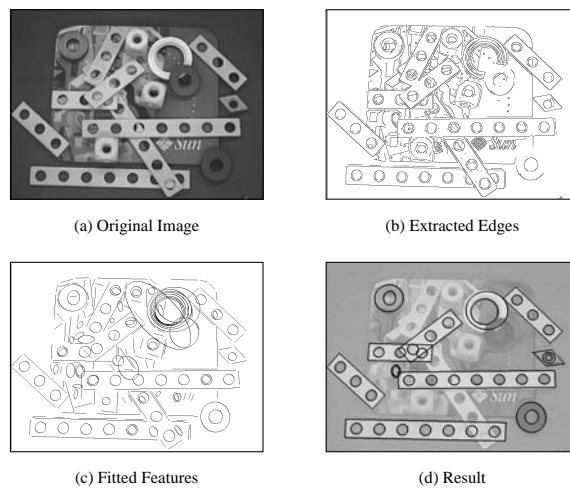


Fig. 5. Example scene 2

In the second scene (Figure 5a) several slats, three rims, one tyre, one nut and three unknown objects on a highly textured background are used. Figure 5b shows the detected edge points and Figure 5c the fitted features.

Although the complexity of this test scene is very high, the recognition system performs well (see Fig. 5d). The system detects all of the known objects except for two 3-hole-slats and one rim. The problems here are a consequence of an inaccurate feature extraction. Since the system fails to extract the topology of these objects correctly, no invariant values can be computed. False positives, however, are not made.

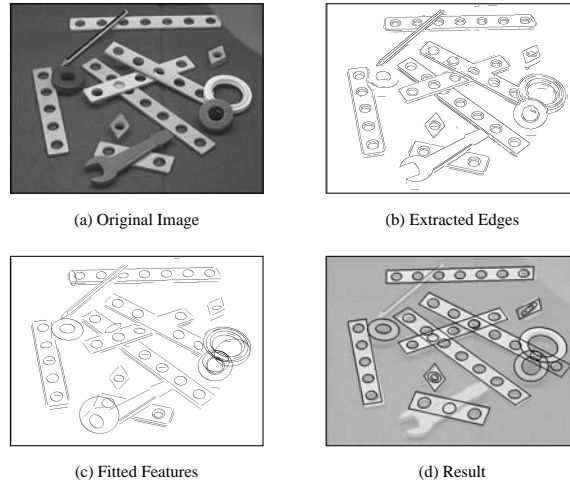


Fig. 6. Example scene 3

Since the utilized invariants remain unaffected by the intrinsic and extrinsic camera parameters, the recognition system can be applied to different imaging conditions. Figure 6a shows a test scene taken at a viewing angle of 45° . Again, despite the resulting perspective distortion, the recognition system performs well. Just one 5-hole-slat has been misclassified as a 7-hole-slat (Fig. 6d). Further recognition results are shown in Fig.14 for hand-camera images.

To demonstrate how the fuzzy rules can easily be extended by integrating further attributes, we added color attributes to the fuzzy rules learned before. We measure the RGB color information of an object along the underlying geometric structures of the fuzzy rules and transform it into the HSV color space. Depending on the saturation of the object color we use the hue or the intensity for generating and evaluating the fuzzy rules, e.g. the rule for the nut (7) changes to:

$$\begin{aligned}
 &\text{IF (inv1} \approx 1.4) \text{ AND (inv2} \approx 3.2) \text{ AND (inv3} \approx 2.5) \\
 &\quad \text{AND (hue} \approx 283) \\
 &\text{THEN (object is NUT)}
 \end{aligned} \tag{10}$$

Experimental results show that this extension generally reduces the number of generated object hypotheses by 25%, which further reduces the likelihood of false matches. For example, the extended fuzzy rules decrease the number of hypotheses for Fig. 4a from 4078 to 3119, for Fig. 5a from 5840 to 4385 and for Fig. 6a from 4094 to 3444. The integration of further attributes enhances the performance of the recognition system in two ways: It speeds up the recognition process because fewer object hypotheses must be investigated into in the time consuming verification stage and, secondly, the robustness of the system is increased, because fewer false positives are established.

Furthermore, we have compared the FII-technique to a crisp invariant indexing technique (where we have replaced the membership functions with intervals). It turns out, that the FII-technique provides better recognition results than the crisp one; this is true, especially in the difficult case of very similar objects (for details, see [9]).

We conclude this section by noting, that the average time for recognising the objects in Fig. 4a is 49 seconds, in Fig. 5a is 77 seconds and in Fig. 6a is 50 seconds on a standard PC (K6-300MHz) with the potential of a straightforward distribution on multiple processors and a resulting drastic reduction in time.

We now turn to our second fuzzy robot vision system that directly controls the movement of a robot arm based on visual input.

3 Fine-Positioning and Object Grasping: Turning Visual Observations into Action.

3.1 Motivation

The task is the fine-positioning of a manipulator once the coarse positioning has been completed. The object to be grasped is visible in the image of a “self-viewing” eye-in-hand camera (Fig. 7), which sees an area of about $11\text{ cm} \times 9\text{ cm}$ of the x - y -plane. The aim is to move the robot hand from its current position (Fig. 8 left) to a new position so that the hand-camera image matches the optimal grasping position (Fig. 8 right). Some of the objects in the image have the same shape but different colors. It is therefore mandatory that a general image processing technique be applied, which needs *no specialised algorithm for each object* and shows stable behaviour under varying object brightness and color. In other words: for dealing with the general case of handling objects whose geometry and features are not precisely modelled or specially marked, it is desirable that a general control model can be found which, after an initial learning step, robustly transforms raw image data *directly into action values*.

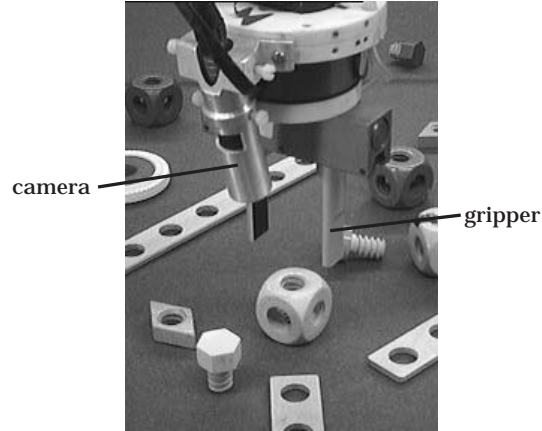


Fig. 7. The end-effector of the manipulator with a hand-camera (positioned optimally over the yellow cube)

3.2 The Neuro-Fuzzy Model

One approach to developing such techniques is neural network-based learning, which has also found applications in grasping: [20,28,12] use geometric features as input to the position controller.

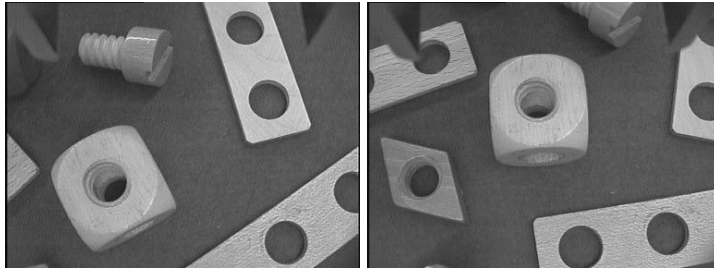


Fig. 8. A cube viewed from the hand-camera – before and after fine-positioning

Our idea to solve the fine-positioning control problem is to use a direct, linear method to reduce the input dimension and then apply the non-linear B-spline model [30] to map the projection on the subspace further to the control output.

Depending on how “local” the measuring data are and, therefore, how similar the observed sensor patterns appear during variations within a given situation, a more or less small number of eigenvectors calculated by a principle component analysis (PCA) [32] can provide a sufficient summary of the state of all input variables (see the left part of Fig. 9). Our experimental results show under the most diverse conditions that it is very likely that three or

four eigenvectors provide all information indices of the original input space necessary for the positioning task. Therefore, in the case of very high input dimensions, an effective dimension reduction can be achieved by projecting the original input space into the eigenspace.

Eigenvectors can be partitioned by covering them with linguistic terms (the right part of Fig. 9). In the following implementations, fuzzy controllers constructed according to the B-spline model are used [30]. This model provides an ideal implementation of CMAC proposed by Albus [1]. We define linguistic terms for input variables with B-spline basis functions and for output variables with singletons. Such a method requires fewer parameters than other set functions such as trapezoid, Gaussian function, etc. The output computation is very simple and the interpolation process is transparent. We also achieved good approximation capabilities and rapid convergence of the B-spline controllers.

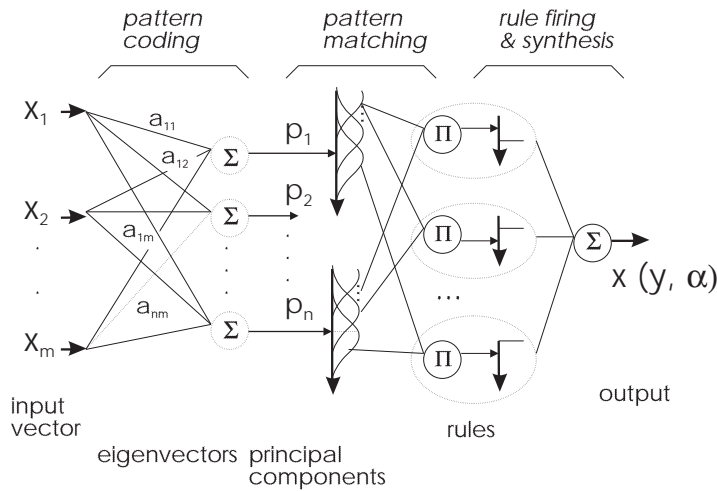


Fig. 9. The task-based mapping can be interpreted as a neuro-fuzzy model. The input vector consists of many thousands of pixels of a grey-scale image

3.3 Implementation

The working system implements two phases: off-line training and on-line evaluation. In the off-line phase, a sequence between 10 and 100 training images showing the same object in different positions is taken automatically, i.e. without human intervention. For each image, the position of the manipulator in the plane and its rotation about the z -axis, both with respect to the optimal grasp position for the current object, is recorded.

In the on-line phase, the camera output is transformed into the eigenspace and is then processed by the fuzzy controller. The controller output is the end-effector's position and angle correction (Fig. 10).

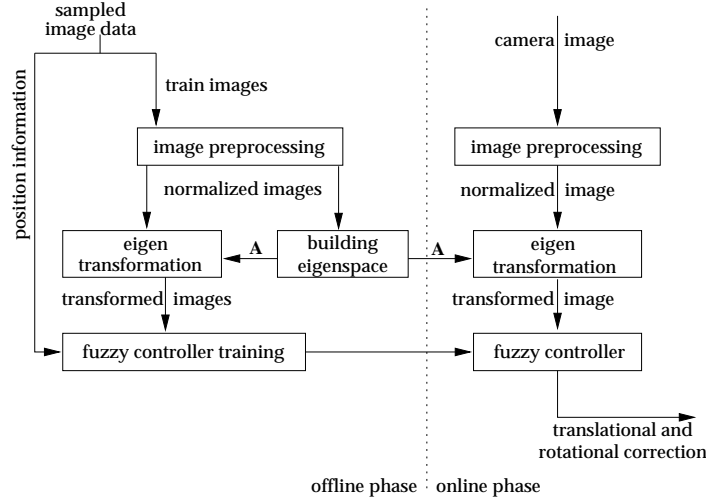


Fig. 10. The training and the application of the PCA neuro-fuzzy controller

Preprocessing. Fig. 14 (left) shows typical pictures taken by the hand-camera. Fig. 14 (right) shows the images after a clipping process utilising the FII recognition system (Sec. 2). This clipping process leaves only the object of interest in the images. After clipping, all images are normalised with respect to their “energy” [22]:

$$x_j^i = \frac{\tilde{x}_j^i}{\sqrt{\sum_{l=1}^{dim} (\tilde{x}_l^i)^2}}$$

where \tilde{x}_j^i is the intensity of the j -th pixel in the i -th image, x_j^i is the intensity of the j -th pixel in the corresponding normalised image and dim is the number of pixels in the image.

For detecting the rotation of an object, one more preprocessing step is necessary: since most of the variance in the images is caused by translations, the rotation cannot be learned properly from the eigen-transformed images. To eliminate the variance caused by changes in the position, we shift the region of interest to the centre of the image. As this removes the translational information from the images, two eigenspaces must be computed: one based on the original images and one based on the shifted versions.

Fuzzy controller training. With the image intensity values and the corresponding desired action values, a B-Spline fuzzy controller is trained. We use third order splines as membership-functions and between 3 and 5 knot points for each linguistic variable. The distribution of these points is equidistant and constant throughout the whole learning process. The coefficients of the B-Splines (de Boor points) are initially zero. They are modified by the rapid gradient descent method during training [30].

On-line phase. In the on-line phase, the same image preprocessing as in the off-line phase is applied. Then, the stacked image vector \mathbf{x} is transformed into the eigenspace. The resulting vector is fed into the fuzzy controller, which, in turn, produces the position and angle of the object in the image. These values are then used to move the robot closer to the target object. This sequence is repeated several times; normally no more than 3 steps are necessary until all parameters (i.e. deviation in x and y direction and residual angular deviation) are below a specific threshold (e.g. 0.5 mm and 1 degree).

To improve the raw algorithm outlined above several aspects were refined:

Color images: Instead of the gray-scale images, the saturation parts of color images in the *Hue-Saturation-Intensity* color-space may be used. For objects with full colors (“rainbow”-colors) the saturation part is high; for colors like teal, pink or light blue this component is low and for all grey-tints including black and white it is zero. This increases the contrast between objects and background when compared with the intensity image. Thus, in the case of colored objects, the controller becomes highly independent of the hue of the objects.

Boosting image vectors: The complexity reduction method is not limited to one image per vector. For example, the vector \mathbf{x} could consist of the intensity image, the saturation image, and a Sobel-filtered intensity image. This can help to suppress inaccuracies due to unusual lighting conditions. Obviously, further (possibly object-dependent) improvements can be achieved with specialised feature detectors (lines, angles, etc.).

Hierarchy: If, for a very difficult object, the discrimination accuracy of the neuro-fuzzy controller is not sufficient, a hierarchical system may be built. The camera images are separated into regions, then an appropriate classifier detects in which region in the image the object to be grasped is located, and, based on this information, the robot moves approximately to the optimal grasping position. After this movement, a neuro-fuzzy controller is trained. The training images for it need only show the object near the optimal position. Such a system is even more accurate than the neuro-fuzzy controller alone.

Optimal choice of training images. Appearance-based vision is frequently criticised for the fact that the *training images* must be chosen man-

ually, which often leads to simple trial-and-error. To cope with this problem, we developed a method for automatically determining the positions where the camera images should be taken. Since the robot is allowed to do several steps, high accuracy is only needed near the optimal grasp position $\mathbf{d} = (x_0, y_0, \alpha_0) = (0, 0, 0)$.

Rotation: The angles at which the images are taken depend on the object symmetry S . For objects with an S of less than 360 degrees there is more than one optimal grasp position. That is because it makes no difference whether a cube is grasped by the front and rear side or at the left and right side. So near the angles $0, S, 2S, \dots$ more images are needed. The objects in Fig. 12 possess the following symmetries: For the slat S is 180 degrees, for the cube S is 90 degrees and for the screw head S is 60 degrees. To limit the number of images for objects with a small S , the following changes are made: If S is smaller than 90 degrees, then it is multiplied by the smallest integer that produces a value of greater than or equal to 90 degrees. This leads, for example, to an S of 120 degrees for the screw head. The following heuristic formula produced acceptable results:

$$\mathbf{W} = \bigcup_{i \in \mathbf{N}_0} \left\{ \left\lfloor \frac{S}{2} \frac{1}{2^i} \right\rfloor + j \cdot S, S - \left\lfloor \frac{S}{2} \frac{1}{2^i} \right\rfloor + j \cdot S \right\}$$

$$j = 0, 1, \dots, 360/S - 1$$

For the cube, this formula gives the set of angles $\mathbf{W} = \{45, 23, 67, 11, 79, 6, 84, 3, 87, 1, 89, 0, 90, 45+90, 23+90, \dots\}$ (in degrees), with \mathbf{W} containing 48 elements. Due to the clipping described in section 3.3 for rotation, only training images near the optimal grasping position are taken, at the points with coordinates $(0, 1), (1, 0), (0, 0), (0, 1),$ and $(1, 0)$. Long objects like the slat can lie partly outside the image. In this case, images with 0, 90, 180 and 270 degrees are added at the 4 positions $(\pm 25 \text{ mm}, \pm 25 \text{ mm})$.

Translation: Images at the learning positions are taken with 0 degrees rotation. In most cases the resulting accuracy for the x - and the y -controller is satisfying with these images. If not, either the controller for x or that for y can be selected. If the y -placement is not correct, then we rebuild the y -controller with images at those positions where $x = 0$.

Preparation of continuous output for learning. Since the fuzzy controller learns to approximate a function, it works correctly only if the function to be learned is continuous, i.e. a differential change of the input will result in a differential change of the output. The correction angle α of the objects to be grasped has different rotation symmetry (lying screw: 360°, slat: 180°, block: 90°, standing screw with six-edge head: 60°). Therefore, we need to find a set of functions which meet the following conditions: a) continuous output values can be generated for fuzzy controller learning; b) the original correction angle can be uniquely reconstructed given the values of these functions. We propose the following two learning functions (Fig. 11):

$$L_a = \sin\left(\frac{1}{2}\alpha\right), \quad L_s = \sin(\alpha)$$

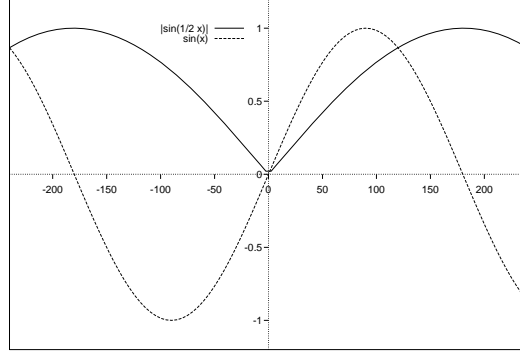


Fig. 11. The two functions which are used for fuzzy controller learning

Two fuzzy controllers are needed to learn L_a and L_s separately. The correction angle can be reconstructed as follows:

- The function arcsine supplies a value between -90° and $+90^\circ$. $|2 \arcsin(L_a)|$ supplies the absolute value of the correction angles.
- The sign of $\arcsin(L_s)$ provides the information on whether the object is rotated clockwise or counter-clockwise with respect to the gripper.

In the application phase, the gripper motion should be corrected in the reverse direction of the object rotation. Therefore, the correction angle α can be calculated:

$$\alpha = -\text{sign}(L_s) \cdot |2 \arcsin(L_a)| \quad (11)$$

These two functions can be extended for objects with the symmetry S :

$$L_a = \sin\left(\frac{360^\circ}{S} \frac{1}{2}\alpha\right); \quad L_s = \sin\left(\frac{360^\circ}{S}\alpha\right)$$

The reconstruction of the angle is then:

$$\alpha = -\frac{S}{360^\circ} \text{sign}(L_s) \cdot |2 \arcsin(L_a)|$$

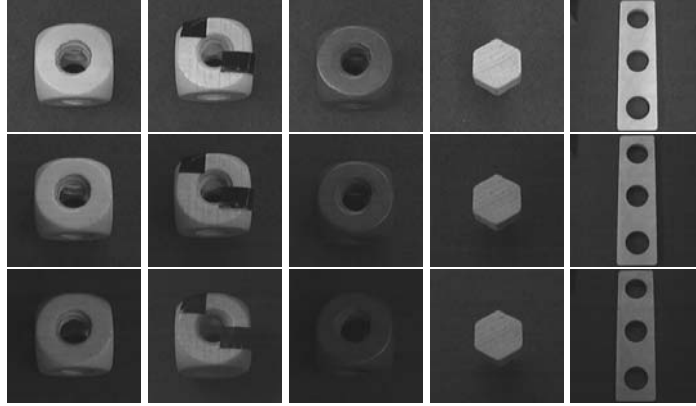


Fig. 12. 15 grasping scenarios; from left to right: yellow cube, partly covered yellow cube, blue cube, yellow screw head, 3-hole-slat; from top to bottom: optimal, worse and poor illumination

3.4 Experimental Results

The approach was applied to the grasping of different objects: a yellow cube, a partly covered yellow cube, a blue cube, a yellow screw head, and a 3-hole-slat (Fig. 12). All training images were taken under optimal lighting conditions. For each object a specific controller was trained, except for the three cubes, where training (not grasping!) was restricted to the yellow cube. For the slat, different training images for x and y were used.

Only the eigenvectors corresponding to the three largest eigenvalues were used as input to the fuzzy controllers. The four largest eigenvalues for rotation/translation and the corresponding eigenvectors, which have the same dimension as the training images and can hence be interpreted as images. The eigenspace and the fuzzy controller that were derived from these data were applied to 15 different scenarios: the manipulator was to be positioned over the five objects, each with optimal, worse, and poor illumination (Fig. 12) and from the most remote starting position. The accuracy of the controllers was determined as the average error of 50 positioning sequences for each scenario.

Table 2 shows the *RMS error* for x , y , and the rotation angle α for positioning above the objects. Obviously, the positioning is correct even for the blue cube with the controller trained on the yellow one. It is easy to see that for the translation it makes hardly any difference whether the illumination is optimal or less optimal. The performance deteriorates under poor lighting conditions but it is still good enough to grasp the object. The rotation is more dependent on the illumination, in particular with the blue cube. That is because the vertical edges of the cube are practically invisible.

Table 2. RMS-errors for the three objects under different lighting conditions. Controllers with three input dimensions and four linguistic terms for each dimension were used.

yellow cube, completely visible				yellow cube, 20% covered			
illumination	x[mm]	y[mm]	α [degree]	illumination	x[mm]	y[mm]	α [degree]
optimal	0.399	0.665	0.608	optimal	0.832	1.093	0.997
worse	0.595	1.525	2.606	worse	0.524	2.373	1.141
poor	3.126	1.038	6.059	poor	6.395	4.728	19.786

blue cube				screw head			
illumination	x[mm]	y[mm]	α [degree]	illumination	x[mm]	y[mm]	α [degree]
optimal	1.658	0.946	1.481	optimal	0.630	0.535	1.850
worse	0.494	2.020	1.979	worse	0.323	0.851	1.897
poor	1.006	0.928	10.803	poor	0.610	0.751	1.281

3-hole-slat			
illumination	x[mm]	y[mm]	α [degree]
optimal	0.272	0.728	0.452
worse	0.940	0.704	0.386
poor	1.198	0.612	0.404





3.5 Linguistic Interpretation of the Controller













One main advantage of the neuro-fuzzy system in comparison with other adaptive systems like the multi-layer perceptron is the interpretability of the controller's function. Since we can transform the projections in the eigenspace back into the original input image space, the control rules can be given an interpretation as follows:

IF Antecedent THEN Consequent

where *Antecedent* is a back-transformed image and the *Consequent* (the controller output) is the x -, y - value or the correction angle α .

The following example illustrates the rules for a two-dimensional controller, each input variable with four linguistic terms. Therefore there are $4 \cdot 4 = 16$ rules altogether. The rotation control looks as follows:

if		then $ \Delta\alpha = 2.2^\circ$	if		then $ \Delta\alpha = 6.7^\circ$
if		then $ \Delta\alpha = 5.8^\circ$	if		then $ \Delta\alpha = 7.6^\circ$

if 	then $ \Delta\alpha = 7.8^\circ$	if 	then $ \Delta\alpha = 19.7^\circ$
if 	then $ \Delta\alpha = 7.9^\circ$	if 	then $ \Delta\alpha = 22.6^\circ$
if 	then $ \Delta\alpha = 11.3^\circ$	if 	then $ \Delta\alpha = 24.8^\circ$
if 	then $ \Delta\alpha = 13.6^\circ$	if 	then $ \Delta\alpha = 28.2^\circ$
if 	then $ \Delta\alpha = 14.9^\circ$	if 	then $ \Delta\alpha = 29.3^\circ$
if 	then $ \Delta\alpha = 17.8^\circ$	if 	then $ \Delta\alpha = 34.3^\circ$

4 A Complete Recognition-Grasping Example

We now show a complete sample run of the hybrid system composed of the subsystems of sec. 2 and sec. 3. For the scene shown in Fig. 1 all steps are performed in an integrated way: a top view image is taken (Fig. 13a); the FII-recognition system recognises the objects of interest (Fig. 13b); the manipulator moves approximately above the object to be grasped; the evaluation of hand-camera images guides the gripper directly to the grasp position; the object is grasped.

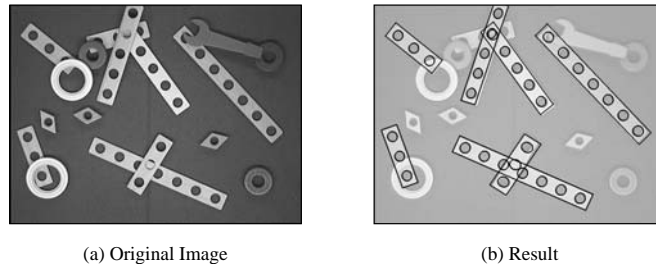


Fig. 13. Test Scene (left) and recognised slats (right)

The task to be solved is the grasping of a slat (Fig. 13a). The FII-recognition system provides all the slats in the image (occluding each other or not), see Fig. 13b.

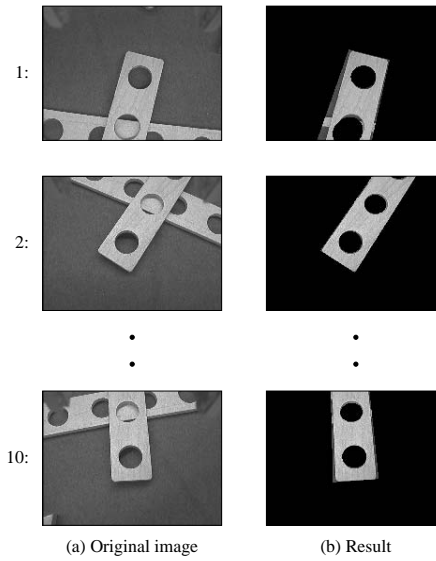


Fig. 14. Images of the hand camera taken at different time stamps

As we have not yet implemented heuristics that automatically pick a slat according to some given criteria, we manually choose the one that is not difficult to grasp: the 3-hole-slat lying on top of the 7-hole-slat. Figure 14 shows a part of the image sequence taken by the hand camera; the right row shows the results with the 7-hole-slat clipped based on the position information obtained by the FII-recogniser.



Fig. 15. Grabbed 3-hole-slat

Finally, as shown in Fig. 15, the slat is grasped successfully.

5 Conclusions

In this paper we have shown that fuzzy rule based methodologies provide an added level of capabilities for robotic vision. We combined fuzzy rule systems with classic vision algorithms (invariants, PCA). This makes some challenging tasks like handling occlusion and calibration-free visual servoing feasible.

The object recognition system that utilizes fuzzy invariant values and fuzzy if-then-rules for the hypotheses generation step combines the approach of using invariants as object shape descriptors with the strength of fuzzy set theory. It is capable of recognising partially occluded objects from different viewpoints robustly, without the need of time-consuming and difficult camera-calibration.

The PCA in conjunction with neuro-fuzzy control is a practical and fast technique for performing multi-variant task-oriented image processing tasks. It is a general method, which needs learning. It is also a calibration-free approach and works robust even when the camera focus is not correctly adjusted or objects are soiled.

A complete recognition-grasping example has demonstrated the combined capabilities of both algorithms. It started with the recognition of partially occluded objects in a complex real robotic scenario and ended with the grasping process of an object of interest.

In sum the systems work without calibration and geometric models can be learned automatically. To a high degree they are:

- immune to uncertainties (illumination, perspective, ...)
- flexible, i.e. it easily adapts to new object poses, object types
- capable of recognising occluded objects in complex scenes

Future research directions will concentrate on some deficiencies of the presented systems:

As mentioned in sec. 2, the implemented FII-recognition system is able to recognise (quasi-)planar objects only. Therefore, future research will focus on the recognition of three dimensional objects. Additionally, the recognition system will be distributed on multiple processors to reduce the recognition time.

The proposed neuro-fuzzy control method for fine-positioning is based on supervised learning. The further development of the approach will be on extending the approach to reinforcement learning which is life-long learning. The similar representation of robot state for fuzzy control can be used for reinforcement learning. The next challenging task is to use vision system to automatically evaluate the quality of grasping after each trial in order to supply the learning correct reward value. That task will demand on integration of active vision, multiple-view and diverse image processing algorithms. Fuzzy methodology can play an important role in the development.

References

1. Albus, J. S., *A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)*, Transactions of ASME, Journal of Dynamic Systems Measurement and Control, Vol. 97, pp. 220–227, 1975.
2. Binford, T. O. and Levitt, T. S., *Quasi-Invariants: Theory and Experiments*, Proceedings of ARPA Image Understanding Workshop, Washington, DC, USA, pp. 819–829, 1994.
3. Black, M.J. and Jepson, A. D., *Eigen-Tracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation*, Proceedings of the European Conference on Computer Vision, Cambridge, Great Britain, pp. 329–342, 1996.
4. Blöchl, B., *Fuzzy Control in Real-time for Vision Guided Autonomous Mobile Robots*, In Klement, E. P., Slany, W. (eds.), *Fuzzy Logic in Artificial Intelligence*, Proceedings of the Austrian Artificial Intelligence Conference, Linz, Austria, pp. 114–125, 1993.
5. Canny, J. F., *A Computational Approach to Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp. 679–698, 1986.
6. Chen, Y.H., *Computer Vision for General Purpose Visual Inspection: A Fuzzy Logic Approach*, Optics and Lasers in Engineering, Vol. 22, pp. 181–192, 1995.
7. Forsyth, D. A., Mundy, J. L., Zisserman, A. P., Coelho, C., Heller, A. and Rothwell, C. A., *Invariant Descriptors for 3-D Object Recognition and Pose*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 10, pp. 971–991, 1991.
8. Farreny, H. and Prade, H., *On the Problem of Identifying an Object in a Robotics Scene from a Verbal Imprecise Description*, In Danthine, A., Gérardin, M. (eds.), *Advanced Software in Robotics*, Elsevier Science, pp. 343–351, 1984.
9. Graf, T., Knoll, A. and Wolfram, A., *Recognition of Partially Occluded Objects through Fuzzy Invariant Indexing*, Proceedings of the IEEE International Conference on Fuzzy Systems, IEEE World Congress on Computational Intelligence, Anchorage, Alaska, USA, Vol. 2, pp. 1566–1571, 1998.
10. Graf, T., Knoll, A. and Wolfram, A., *Fuzzy Invariant Indexing: A General Indexing Scheme for Occluded Object Recognition*, Proceedings of the International Conference on Signal Processing, Beijing, China, pp. 908–911, 1998.
11. Keller, J. M., *Fuzzy set theory in computer vision: A prospectus*, Fuzzy Sets And Systems, Vol. 90, No. 2, pp. 177–182, 1997.
12. Kamon, I., Flash, T. and Edelman, S., *Learning to Grasp Using Visual Information*, Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2470–2476, 1996.
13. Knoll, A., Hildebrandt, B. and Zhang, J., *Instructing Cooperating Assembly Robots through Situated Dialogues in Natural Language*, Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, NM, pp. 888–894, 1997.
14. Kosako, A. and Ralescu, A. L., *Feature Based Parametric Eigenspace Method for Object Extraction*, Proceedings of the IEEE International Conference on Fuzzy Systems, Yokohama, Japan, pp. 1273–1278, 1995.
15. Lee, K.-J. and Bien, Z., *A Model-Based Machine Vision System Using Fuzzy Logic*, International Journal of Approximate Reasoning, Vol. 16, pp. 119–135, 1997.

16. Lamdan, Y., Schwartz, J. T. and Wolfson, H. J., *Object Recognition by Affine Invariant Matching*, Proceedings of Computer Vision and Pattern Recognition, pp. 335–344, 1988.
17. Li, W., Jiang, X. and Wang, Y., *Road Recognition for Vision Navigation of an Autonomous Vehicle by Fuzzy Reasoning*, Fuzzy Sets and Systems, Vol. 93, pp. 275–280, 1998.
18. Mundy, J. L., Huang, C., Liu, J., Hoffman, W., Forsyth, D. A., Rothwell, C. A., Zisserman, A., Utcke, S. and Bournez, O., *MORSE: A 3D Object Recognition System Based on Geometric Invariants*, Proceedings of ARPA Image Understanding Workshop, Monterey, California, pp. 1393–1402, 1994.
19. Mundy, J. L. and Zisserman, A., *Introduction - Towards a New Framework for Vision*, Geometric invariance in computer vision, Cambridge, Mass.: MIT Press, 1992.
20. Miller, W. T., *Real-Time Application of Neural Networks for Sensor-Based Control of Robots with Vision*, IEEE Transactions on System, Man and Cybernetics, Vol. 19, pp. 825–831, 1989.
21. Mundy, J. L. and Zisserman, A., *Geometric invariance in computer vision*, Cambridge, Mass.: MIT Press, 1992.
22. Nayar, S. K., Murase, H. and Nene, S. A., *Learning, Positioning, and Tracking Visual Appearance*, Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3237–3244, 1994.
23. Pal, S. K., *Uncertainty Management in Space Station Autonomous Research: Pattern Recognition Perspective*, Information Sciences, Vol. 72, pp. 1–63, 1993.
24. Rothwell, C. A., *Object recognition through invariant indexing*, Oxford University Press, 1995.
25. Ray, K. S. and Ghoshal, J., *Neuro-Fuzzy Reasoning for Occluded Object Recognition*, Fuzzy Sets and Systems, Vol. 94, pp. 1–28, 1998.
26. Sanger, T., *An Optimality Principle for Unsupervised Learning*, Advances in Neural Information Processing Systems, Morgan Kaufmann, San Mateo, CA, 1989.
27. Walker, E. L., *Perspectives on Fuzzy Systems in Computer Vision*, Proceedings of the Annual Conference of the North American Fuzzy Information Processing Society, pp. 296–300, 1998.
28. Wei, G.-Q. Wei, Hirzinger, G. and Brunner, B., *Sensorimotion Coordination and Sensor Fusion by Neural Networks*, Proceedings of IEEE International Conference on Neural Networks, San Francisco, USA, pp. 150–155, 1993.
29. Zadeh, L. A., *Fuzzy sets*, Information and Control, Vol. 8, pp. 338–353, 1965.
30. Zhang, J. and Knoll, A., *Constructing Fuzzy Controllers with B-Spline Models – Principles and Applications*, International Journal of Intelligent Systems, Vol. 13, pp. 257–285, 1997.
31. Zhang, J., Schmidt, R. and Knoll, A., *Appearance-Based Visual Learning in a Neuro-Fuzzy Model for Fine-Positioning of Manipulators*, Proceedings of the IEEE International Conference on Robotics and Automation, Detroit, USA, 1999.
32. Zhang, J. and Knoll, A., *Situated Neuro-Fuzzy Control for Vision-Based Robot Localisation*, Journal of Robotics and Autonomous Systems, Elsevier Science, Vol. 28, pp. 71–82, 1999.