Proceedings of the 1999 IEEE/RSJ
International Conference on
Intelligent Robots and Systems

Integration of visual cues for active tracking of an end-effector

D. Kragić
Computational Vision and Active Perception
Numerical Analysis and Computing Science
Royal Institute of Technology
SE-100 44 Stockholm, SWEDEN

H. I. Christensen
Centre for Autonomous Systems
Numerical Analysis and Computing Science
Royal Institute of Technology
SE-100 44 Stockholm, SWEDEN

Abstract

In this paper we describe and test how information from multiple sources can be combined into a robust visual servoing system. The main objective is integration of visual cues to provide smooth pursuit in a cluttered environment using a minimum or no calibration. For that purpose, voting schema and fuzzy logic command fusion are investigated. It is shown that the integration permits detection and rejection of measurements outliers.

1 Introduction

Closed loop control enables increased robustness in robotic manipulation. Vision is a particularly powerful sensory modality for feedback control. When applied in closed loop control it is referred to as *visual servoing*. One of the most common tasks in visual servoing is to maintain a desired visual pose of a moving target, i.e. an end–effector of a robotic arm.

Both monocular and stereo vision approaches [1, 2, 3, 4] have been facilitated to solve this task. Object (end–effector) tracking is a necessary precursor to many tasks in visual servoing. It can, for example, be used for positioning the gripper with respect to known or unknown object in the workspace. However, almost all of the existing techniques facilitate either a model based information that requires an off-line system initialization or special markers/fiducial points on a robotic arm.

The performance of feature-based visual servoing depends on the robustness and uniqueness of the features used. The feature selection problem has been discussed extensively in the literature [5, 2, 6]. However, most tasks require sophisticated image processing to extract the target and in that case we can limit processing to small windows.

To enable vision to be used in real–world applications an added degree of robustness must be introduced to facilitate use of natural features for visual tracking. In a natural environment no single cue is likely to be robust for an extended period of time and there is thus a need for fusion of multiple cues. Rather than solving a task by building several modules that have to perform good in unforeseen situations, we are interested in constructing reliable cues or behaviors from a multitude of less reliable ones [7, 8]. The cues can contribute to the control of the system in a number of different ways. They can vote for alternative commands and their votes can then be fused. Another approach is that one cue can trigger the next cue until the final objective is achieved. Either way, a particular command is generated that is used to control the system.

The idea we propose is a development of a modular system that consists of simple and in–expensive visual cues that are used for the control. By using the information from multiple cues we will be able to make manipulation tasks more robust and to perform them in a cluttered environment without tailoring the process in any special way.

The paper is organized as follows. In Section 2, visual cues used for the integration are presented. Those cues can be viewed as redundant behaviors since they have an identical task objective. In Section 3, we present techniques for fusion of visual cues, while in Section 4 we describe the control algorithm. Initial experimental results are shown in Section 5. Finally, a summary and issues for future research are presented in Section 6.

2 Visual Cues

We combine disparity, motion, color, edges and normalized cross correlation to achieve the robustness demanded by real world applications.

0-7803-5184-3/99/$10.00 © 1999 IEEE

2.1 Stereo Model

For a pinhole camera model and a parallel axis stereo system where we assume that the image planes are rectified and internal camera parameters same, we can express the coordinates of the left image plane in the coordinate system of the right image plane as [9]:

$$\begin{bmatrix} \alpha & 0 & X_0 - X_L \\ 0 & \beta & Y_0 - Y_L \\ \alpha & 0 & X_0 - X_R \\ 0 & \beta & Y_0 - Y_R \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ B\alpha \\ 0 \end{pmatrix}$$

The following notation has been used:

- $X_{R,L}, Y_{R,L}$ are the image coordinates of a point for the right and left image plane
- X_0, Y_0 represent image center
- X, Y, Z are world coordinates of a point
- α and β are internal camera parameters
- B is the baseline distance

The system is solved using a least square method.

2.2 Motion Cue

The absolute difference image of the intensity component (I) of consecutive images is computed as:

$$M^{l,r}(\mathbf{X}) = \mathbf{H}(|I^{l,r}(\mathbf{x},t) - I^{l,r}(\mathbf{x},t-1)| - \Gamma)$$

where Γ is a fixed threshold and $\mathbf{H}$ is the Heavyside function. To remove isolated pixels and highlight those that have a number of direct neighbors, a median filter is used. We segment the scene into static and moving regions since only objects having a non-zero temporal difference change position between frames. However, the motion cue responds not only to all moving regions but also to the strong changes in the illumination. In addition, we have to compensate for the egomotion of the camera head itself before computing the motion cue. Egomotion estimation is based on encoder readings of the pan-tilt unit and inverse kinematics.

2.3 Color Cue

We represent our images in the HSV space since this representation is less sensitive to variations in illumination. Therefore, the color detection of the robot's end-effector is based on the *hue* (H) and *saturation* (S) components of the color histogram values.

Saturation is a measure of the lack of whiteness in the color, while the *hue* is defined as the angle from the red color axis.

$$H = acos\left[\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right]$$

$$S = 1 - \frac{3}{(R+G+B)} min(R,G,B)$$

$$V = \frac{1}{3}(R+G+B)$$

To achieve real-time performance the color to be recognized has been selected *a priori*. Color training is done off-line, i.e. the known color is used to compute its color distribution in the H-S plane. In the segmentation stage all pixels whose hue and saturation values fall within the set defined during off-line training and whose brightness value is higher than a threshold are assumed to be object of interest.

2.4 Normalized Cross Correlation

The idea is to track the object of interest in the part of the image by searching for the region in the image that looks like the desired object defined by some mask or sub-image (template). The image template describes color, texture and material that the object is made from. This kind of modeling includes assumptions about ambient lightning and background color that are not object's features and, therefore, will effect the performance of the cue.

Template matching is done by normalized cross correlation (NCC). The region providing the maximal similarity measure is selected as the location of the object in the image. To decrease the computation time the template matching algorithm is initialized in the region where the object of interest was found in the previous frame.

In addition, the loop short–circuiting, the heuristic best place search beginning and spiral image traversal pattern as described by [10] are used to optimize the search.

2.5 Disparity Map

For a particular feature in one image, there are usually several matching candidates in the other image. For computing the disparity map we used grey level values correlation based stereo technique. It is usually necessary to use additional information or constraints to assist in obtaining the correct match [11]. We have used the epipolar constraint, uniqueness and ordering constraint. We implemented dynamic programming that is based on matching of windows of pixel intensities, instead of using windows of pixel intensities of gradient values. A maximum likehood cost function is used to find the the most probable combination of disparities along a scan–line. This cost function takes into account the ordering and uniqueness constraints. The complexity of finding a unique minima is greatly reduced by dynamic programming. The size of the correlation mask was 9x9 pixels.

2.6 Edges

Since at this stage we are mostly interested in clusters of edges, we use a simple gradient based edge detector operator.

3 Implementation

Each of the cue employed in this experiment will perform different in different situations and degrade the performance in various ways. Many of suggested methods were implemented in pattern recognition [12, 13] and the dominating method in computer vision has been Bayesian estimation [14, 15]. Most of these methods employ a model based (phenomological) approach that relate the cues to the external word. The models have a limited application domain and it is difficult to design models that can operate in rich environments.

Our approach is a model–free approach to fusion [16]. First, the data from individual modules has to be transformed into a common representation and then combined so that the final output can be used in a control algorithm. We denote visual modules as cue estimators.

Each cue estimator, presented in Section 2, is a function that operates on a certain region (region of interest) and delivers the binary answer [0, 1] whether or not a certain pixel satisfies the conditions of the given function. In the case of voting schema, the voting space, Θ, is the image plane.

As presented in (Fig. 1), in the case of the fuzzy fusion, we integrate the information from n–sample histograms. Here, n is the number of cue estimators. Each cue estimator delivers a histogram where the values on the apscisa present the pixel number in the x–horizontal image direction and the ordinata presents the sum of the pixel values from different cue estimators for a certain x in the y–vertical image direction.

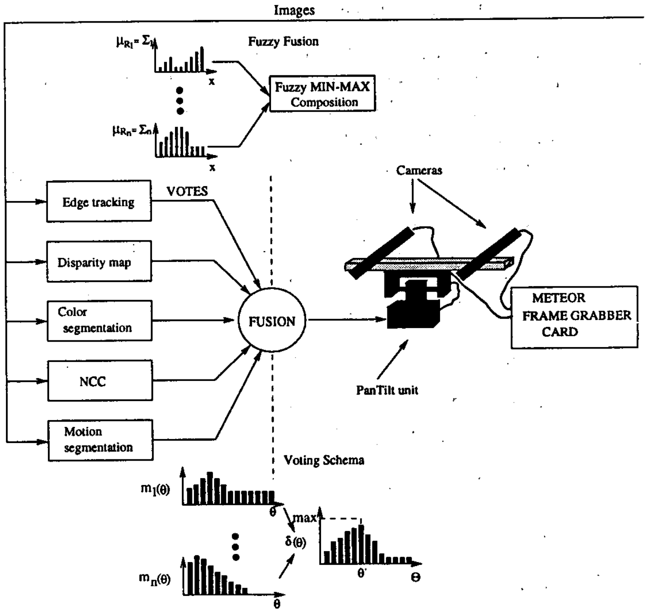

Figure 1: Schematic overview of the system.

3.1 Fusion Using Voting Schemas

The basic idea in voting is to combine (binary) decisions from several cue estimators to improve the probability of making a correct decision [7]. There are several different classes of voting schemas [16] including majority, mean, and plurality voting.

We have implemented a weighted plurality voting, which chooses the action that has received the maximum number of votes. This schema can be expressed as follows:

$$\delta(\theta) = \sum_{i=1}^{n} w_i m_i(\theta)$$

where δ represents approval voting scheme for a n number of cue estimators $m_1, m_2, ...m_n$ and w is a weighting factor. θ is a part of the image that is currently observed (ROI). The most appropriate action is selected according to:

$$\theta' = \max\{\delta(\theta) \mid \theta \in \Theta\}$$

where Θ is the voting space (image plane in our case).

3.2 Fuzzy Command Fusion

Fuzzy systems belong to the class of knowledge based systems that aim to implement human know-how or heuristic rules in the form of a computer pro-

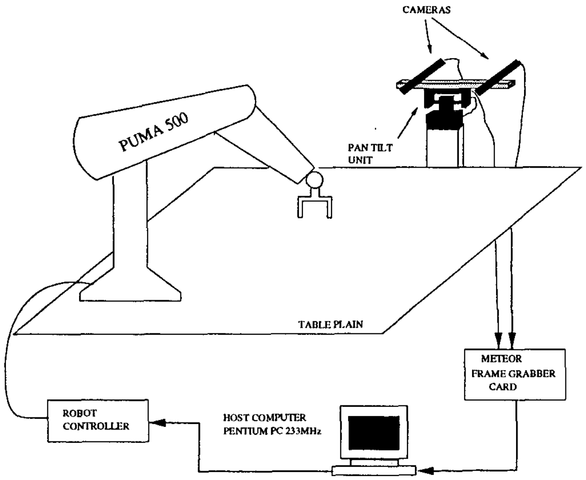

Figure 2: Robot Configuration

gram [17]. Fuzzy logic provides a mathematical formalism for this goal. Fuzzy controllers model human experience in the form of linguistic *if-then* rules; a fuzzy inference engine computes the control actions for each given situation.

In this approach we will consider that sample histograms as fuzzy relations defined on different product spaces [17] and we will combine them by the operation "composition",as defined by Zadeh [18]:

$$\mu_{R_1 o R_2 o \ldots o R_n} = \max\{\min(\mu_{R_1}, \mu_{R_2}, \ldots \mu_{R_n})\}$$

where μ_{R_i} are the outputs of cue estimators.

4 Active Tracking and Control

The discipline of active vision was developed in the 80's by [19, 20, 21]. The objective is that a vision system should not spend time on obtaining the maximum information from individual images but rather concentrate on particular information related to the task at hand [22]. Our camera head has two degrees of freedom: the pan and the tilt angles. Control of the two angular velocities (i.e. velocities of the pan and tilt angle) can be expressed as [2]:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} \frac{xy}{f} & \frac{-f^2-x^2}{f} \\ \frac{f^2+y^2}{f} & \frac{-xy}{f} \end{bmatrix} \begin{bmatrix} \omega_{xc} \\ \omega_{yc} \end{bmatrix}$$

Here, f is the focal length of the camera's lens, subscript c denotes the relation to the camera frame and x and y are the image coordinates. The error signal is defined as a difference between the target image position and some reference position which is in our case the center of the image. We are using a P–controller to recenter the target in the image.

5 Experiments

The aim of the experiments was to investigate the hypothesis that fusion of information from multiple sources can lead to improved reliability. We have also investigated which of two different techniques for fusion gives better results for our task. A series of the experiments were performed and we present the results from two of them. In the conducted experiments, we tested different weighting factors for each individual cue.

5.1 Experimental Setup

In this project an external camera system is employed with a pair of color CCD cameras arranged for stereo vision, see (Fig. 2). The cameras view a robot manipulator and its workspace from a distance of about 2m. The camera pair is mounted on the pan-tilt unit with 2DOF and together they make up the "stereo head". The size of the original image was 320x240 pixels. In the experiments presented here, the object being tracked is a PUMA560 robotic arm. The movement of the arm was under external control.

5.2 Experimental Results

The arm moved on a straight path for about 100cm. The distance of the arm relative to the camera was increasing from about 75 to 100cm. The rotation of the end–effector was about 20°. Three images from the two presented experiments can be seen in (Fig. 7) and (Fig. 8). During arm movement, visual cues were extracted as presented in Section 2. Those results are fused as explained in Section 3 and then fed to the control algorithm as presented in Section 4. The results are presented with the measure of *relative error* in Table 1 and Table 2 separately for the X–horizontal and Y–vertical image components to determine which of these components gives stronger contribution to the overall result.

Relative error is the difference in pixels between the position of the tracked object obtained by a cue estimator and the ground truth value. The ground truth value, i.e. the position of the end-effector in the image, is analyzed manually on the sequence of recorded images.

5.3 Experiment 1:

This experiment demonstrates the performance of the implemented system for a regular scene. Most of the cues had resonable performance, see Table 1.

Module	$\bar{X}$	STD X	$\bar{Y}$	STD Y	$\overline{(X,Y)}$	STD
Color	-0.44	3.84	-7.86	1.64	**8.74**	**1.59**
Motion	-0.44	2.29	-4.82	2.72	**5.52**	**2.31**
Disp.	7.03	4.46	13.79	2.32	**15.92**	**3.30**
Edges	-1.65	2.39	-4.10	2.54	**5.00**	**2.55**
NCC	-3.75	3.31	1.82	0.65	**4.70**	**2.56**
Voting	-2.17	4.09	-0.20	1.42	**4.01**	**2.65**
Fuzzy	-0.34	3.93	-5.34	3.69	**7.27**	**2.00**

Table 1: Relative error presented by the mean distance error and standard deviation for the Experiment 1.Main results are highlighted.

Module	$\bar{X}$	STD X	$\bar{Y}$	STD Y	$\overline{(X,Y)}$	STD
Color	3.10	3.63	-5.55	6.05	**8.88**	**3.18**
Motion	1.89	1.65	-4.17	3.78	**5.59**	**2.53**
Disp.	11.37	4.63	-6.51	3.23	**13.75**	**3.76**
Edges	1.31	2.20	-5.65	3.46	**6.56**	**2.68**
NCC	-3.27	3.48	0.17	3.21	**4.29**	**3.81**
Voting	1.27	2.87	-3.55	2.14	**4.91**	**1.62**
Fuzzy	1.24	4.32	-5.00	3.68	**6.77**	**3.49**

Table 2: Relative error presented by the mean distance error and standard deviation for the Experiment 2.Main results are highlighted.

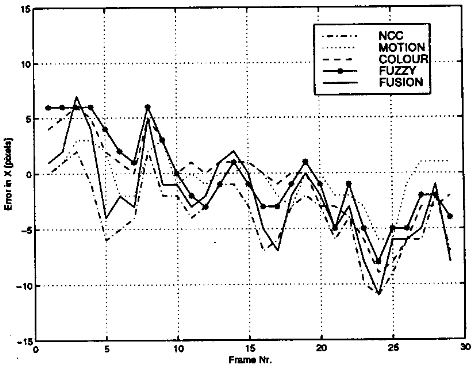

Figure 3: Distance error in the $X-$horizontal direction for the Experiment 1. For the clarity reasons, the results from the edge and disparity module are not presented.

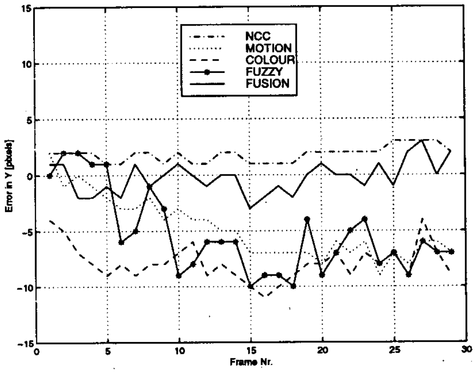

Figure 4: Distance error in the $Y-$vertical direction for the Experiment 1. For the clarity reasons, the results from the edge and disparity module are not presented.

Due to the lack of texture on the target object, the disparity cue deviated significantly. As we can see in (Fig. 5.3) and (Fig. 5.3), higher weight on the correlation implies that the voting schema heavily relied on this particular cue. In the case of the fuzzy integration, all the cues had the same weighting factor which resulted with unstable performance.

5.4 Experiment 2:

In this experiment, we introduced multiple moving targets and occlusion. The aim was to test how the integration will perform in the case where the NCC or the motion cue fail to detect the target or give an incorect response. As it can be seen in Table 1, the NCC gave better results regarding the overall mean error. However, see (Fig.5) and (Fig.6), both NCC and motion cue completely fail to detect target in certain frames. This means that it is impossible to rely just on one of this cues in the case of continious tracking. Considering the performance of the fuzzy integration, we realized that fuzzy approach to cue integration was not of the desired quality for this kind of task.

Presented results illustrate that the integration of multiple cues facilitates continious tracking performance in the case of clutter and multiple moving targets. Our future work will consider a better reasoning on a choice for a particular cue used in the integra-

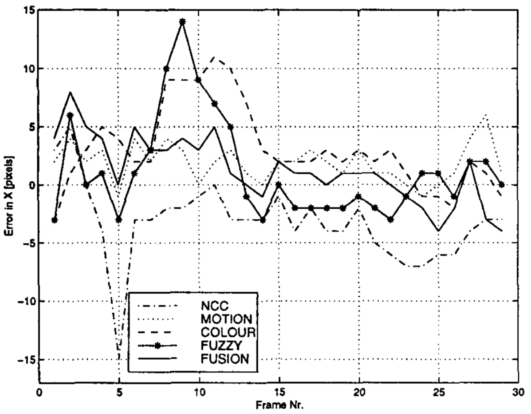

Figure 5: Distance error in the X–horizontal direction for the Experiment 2. For the clarity reasons, the results from the edge and disparity module are not presented.

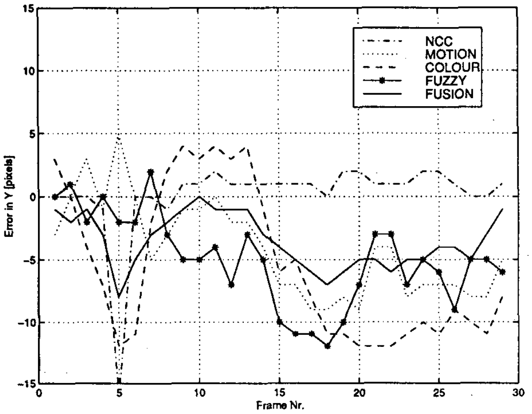

Figure 6: Distance error in the Y–vertical direction for the Experiment 2. For the clarity reasons, the results from the edge and disparity module are not presented.

tion.

6 Conclusion and Future Work

We have presented a vision-based techniques for tracking a robot end–effector. We have experimentally shown that reliable and robust tracking can be performed by integration of individual cues. In our approach there is no need for building detailed models of the environment or using the *a priori* information about the object to be tracked.

We proposed two different integration techniques: fusion using voting schemas and fuzzy command fusion. Experimental results showed that the voting schemas performed better in presented scenarios. However, more appropriate fuzzy inferencing techniques should be investigated as they might improve the tracking performance.

In the future we aim to perform the visual servoing tasks without using special markers on the end–effector and without need to manually initialize the tracking region. We will investigate the performance of the system in positioning the end–effector with an object in the environment. In addition, further experiments will investigate how the initial guess of the position of the end–effector influence the the performance of each individual module as well as the performance of the proposed integration techniques.

7 Acknowledgement

This research has been sponsored by the Swedish Foundation for Strategic Researh through the Centre for Autonomous Systems. The funding is gratefully acknowledged. The implementation of visual modules has benefitted from insightful discussions with Mårten Björkman.

References

[1] R. Horaud, F. Dornaika, and B. Espiau, "Visually guided object grasping," *IEEE TRA*, vol. 14, no. 4, pp. 525-532, 1998.

[2] F. Chaumette, P. Rives, and B. Espiau, "Positioning a robot with respect to an object, tracking it and estimating its velocity by visual servoing," *IEEE ICRA*, 1991.

[3] P. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE TRA*, vol. 9, no. 2, pp. 152–165, 1993.

[4] G. Hager, "A modular system for robust positioning using feedback from stereo vision," *IEEE TRA*, vol. 13, no. 4, pp. 582–595, 1998.

[5] J. Feddema, C. Lee, and O.R.Michell, "Automatic selection of image features for visual servoing of a robot manipulator," *IEEE ICRA*, pp. 832–837, 1987.

[6] N.Hollinghurst, *Uncalibrated Stereo and Hand-Eye Coordination*. PhD thesis, Department of Engineering, University of Cambridge, 1997.

[7] P. Pirjanian, H. Christensen, and J. Fayman, "Application of voting to fusion of purposive modules: An experimental investigation," *Robotics and Automation Systems*, vol. 23, no. 4, pp. 253–266, 1998.

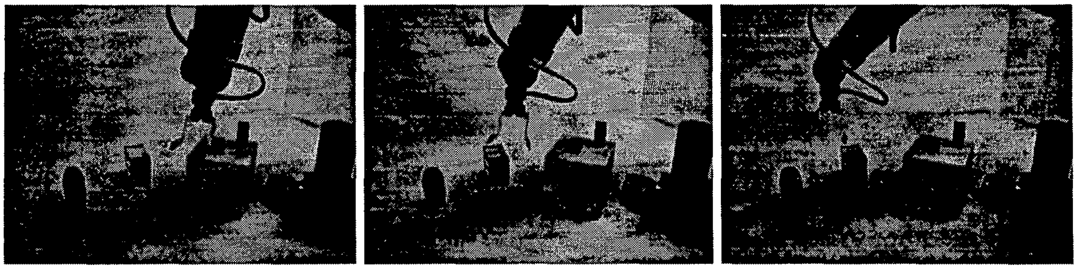

Figure 7: Frames number 10, 20 and 30 of the configuration in the Experiment 1.

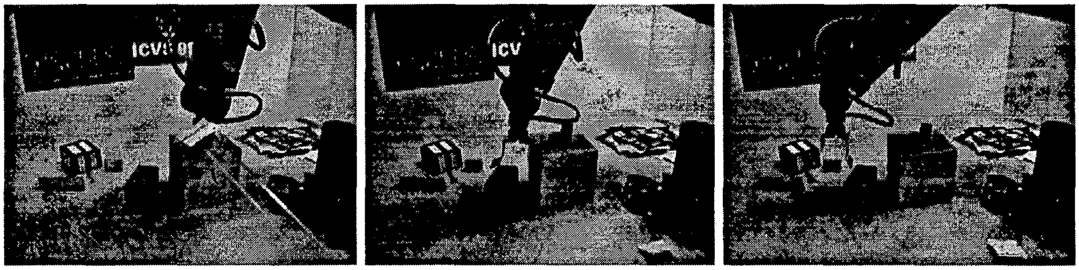

Figure 8: Frames number 10, 20 and 30 of the configuration in the Experiment 2.

[8] P.Pirjanian, *Multiple Objective Action Selection And Behaviour Fusion Using Voting.* PhD thesis, Department of Medical Informatics And Image Analysis, Aalborg University, 1998.

[9] D. Kragić and H. Christensen, "Using a redundant coarsely calibrated vision system for 3d grasping," *Proc. CIMCA'99*, 1999.

[10] C. Smith, S. Brandt, and N. Papanikolopoulos, "Eye–in–hand robotic tasks in uncalibrated environments," *IEEE TRA*, vol. 13, no. 6, pp. 903–914, 1996.

[11] C. Eveland, K. Konolige, and R. Bolles, "Background modelling for segmentation of video–rate stereo sequences," *Proc. CVPR*, June 1998.

[12] I. Bloch, "Information combination operators for data fusion: A comparative review with classification," *IEEE Trans. on Systems, Man and Cybernetics, Part A:Systems and Humans*, vol. 26, no. 1, pp. 42–52, 1996.

[13] L. Lam and C. Suen, "Application of majority voting to pattern recognition: An analysis of its behaviour and performance," *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 27, no. 5, pp. 553–568, 1997.

[14] A. Blake and A. Zisserman, *Visual reconstruction.* Cambridge, Massachusetts: MIT Press, 1987.

[15] J. Aloimonos and D. Shulman, *Integration of Visual Modules.* Academic Press, Inc, 1989.

[16] B. Parhami, "Voting algorithms," *IEEE Transactions on Reliability*, vol. 43, no. 3, pp. 617–629, 1994.

[17] J. Godjevac, *Neuro-Fuzzy Controllers; an Application in Robot Learning.* PPUR-EPFL, 1997.

[18] L. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 3, no. 1, pp. 28–44, 1973.

[19] R. Bajcsy, "Active perception," *IEEE Proc.*, vol. 76, pp. 996–1006, Aug. 1988.

[20] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *IJCV*, vol. 1, Jan. 1988.

[21] D. Ballard, "Animate vision," *Artificial Intelligence*, vol. 48, pp. 1–27, Feb. 1991.

[22] J. Crowley and H. Christensen, "Integration and control of active visual pocesses," *IROS*, Aug. 1995.