Cue Integration for Visual Servoing

Danica Kragić and Henrik I. Christensen, *Member, IEEE*

***Abstract*—The robustness and reliability of vision algorithms is, nowadays, the key issue in robotic research and industrial applications. To control a robot in a closed-loop fashion, different tracking systems have been reported in the literature. A common approach to increased robustness of a tracking system is the use of different models (CAD model of the object, motion model) known *a priori*. Our hypothesis is that fusion of multiple features facilitates robust detection and tracking of objects in scenes of realistic complexity. A particular application is the estimation of a robot's end-effector position in a sequence of images.**

The research investigates the following two different approaches to cue integration: 1) voting and 2) fuzzy logic-based fusion. The two approaches have been tested in association with scenes of varying complexity. Experimental results clearly demonstrate that fusion of cues results in a tracking system with a robust performance. The robustness is in particular evident for scenes with multiple moving objects and partial occlusion of the tracked object.

***Index Terms*—Fuzzy logic, tracking, visual cues, visual servoing, voting.**

I. Introduction

VISUAL servoing is composed of two intertwined processes: tracking and control. The major components of a visual servo system are shown in Fig. 1. Tracking is responsible for maintaining an estimate of the target's position, while servo control reduces the error between the current and the desired position of the target. Each of these processes can be studied independently, but the actual implementation must consider the interaction between them to achieve robust performance. Tracking of a target can be divided into the following three subtasks: 1) detection of the target; 2) matching across images in an image stream; and 3) estimation of the motion of the target (see Fig. 2). Once the position and the velocity of the target are known they can be fed into a control loop. The tracking can either be performed in image (i.e., image coordinates of the tracked features are estimated) or in world coordinates (a model of the target/camera parameters are used to retrieve the three-dimensional (3-D) pose of the tracked features). *Image-based* servo control uses image coordinates of the features directly in the control loop. If the control is performed in the 3-D Cartesian space we talk about *position-based* servo control [22]. In this work, image-based tracking and servoing are studied.

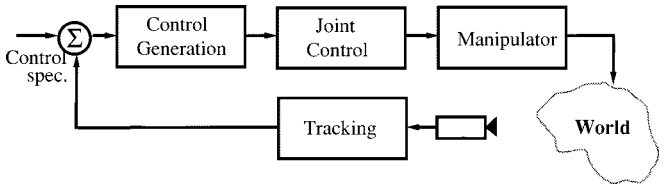

Fig. 1. The major components of a visual servo system.

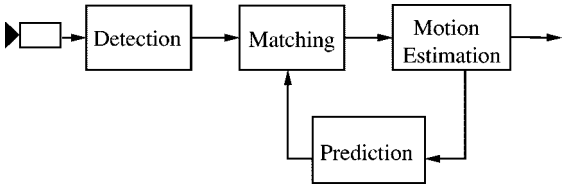

Fig. 2. The major processes involved in visual tracking.

The reported literature on visual tracking and servoing is extensive. Some examples include the work by Dickmanns et al. [1] and Kollnig and Nagel [2]. Both of these approaches have adopted specific models [in terms of the environment and/or the objects of interest (cars)]. In terms of manipulation many approaches exploit markers or *a priori* known features like lines and corners to simplify detection and tracking. Examples of such work include Hager [3], Allen [4], and Papanikolopoulos and Khosla [5].

A notorious problem in visual servoing is robustness. Lack of robustness is primarily due to the following three problems: 1) figure ground segmentation (detection of the target); 2) matching across images (in particular in the presence of large inter-frame motion); and 3) inadequate modeling of motion (to enable prediction of the target in new images). Robust target detection is often achieved through the use of artificial markers that are easy to segment. An alternative approach is the use of CAD models, for example, as demonstrated by Kollnig and Nagel [2] and Hirzinger et al. [6]. Such an approach is particularly relevant for tracking of well-known/well-defined objects that can be modeled *a priori*. However, for general objects of complex shapes, an approach to increased robustness may be an integration of multiple visual cues. A lot of work has been reported on fusion of visual cues [7]–[9]. Most of the reported techniques are model-based [8], where a specific model of the imaging process and feature extraction is used as the basis for fusion of cues. Visual servoing requires techniques that are suited for real-time implementation. To achieve this, one often has to resort to 'simple' visual cues. In addition, fast execution and redundancy enable simplified tracking and allow handling of temporary loss of individual features. In this

Manuscript received January 18, 2000; revised November 17, 2000. This paper was recommended for publication by Associate Editor H. Zhuang and Editor V. Lumelsky upon evaluation of the reviewers' comments. This work was supported by the Swedish Foundation for Strategic Research through the Centre for Autonomous Systems.

D. Kragić is with the Computational Vision and Active Perception Laboratory, Numerical Analysis and Computer Science Department, The Royal Institute of Technology (KTH), Stockholm, Sweden (e-mail: danik@nada.kth.se).

H. I. Christensen is with the Centre for Autonomous Systems at the Royal Institute of Technology (KTH), Stockholm, Sweden (e-mail: hic@nada.kth.se).

Publisher Item Identifier S 1042-296X(01)03162-7.

paper, two different methods for real-time fusion of visual cues are investigated: integration using voting and integration using fuzzy logic.

Initially, two methods for integration of visual cues are proposed. It is then outlined how these methods are used to design a tracking system for hand–eye coordination. The system has been experimentally tested and evaluated. Finally, conclusions regarding the proposed methodology and experimental results are presented.

II. Cue Integration Methods

Our environment offers a rich set of cues originating from the scene structure and events. Recent work on robot navigation and mobile manipulation employs more than one sensor for task execution. Such multisensor or multicue integration techniques perform very well in complicated and dynamic environments where the uncertainty of individual sensors/cues vary during the task execution [31]. The information obtained by one or multiple sensors is both complementary and redundant hence it supports the robustness of algorithms in unforeseen situations. Machine vision is a typical modality that benefits from this redundancy.

A probabilistic framework has been widely used in fusion of multiple cues [8], [10], [11]. For example, as described in [12] and [13], these methods have primarily been used for pattern recognition and scene reconstruction. This approach requires a model that fits to the data and prior distributions of possible results. The problem is that the validity of the model is often limited and it might be difficult or impossible to verify the correctness of the model at run-time.

An alternative to the use of strong models is model-free approach to cue integration [9], [14], [15]. This approach exploits the incidental agreement of multiple cues where methods like voting and fuzzy logic can be used for deciding on agreement. This approach has been widely used in cases where a precise mathematical model of the controlled process is not available [15]–[18]. The following section introduces voting and fuzzy logic-based cue integration which are used to design a tracking system. The objective of the system is to track a portion of the image occupied by the end-effector of a robotic manipulator. The center of this region is then used to control a pan-tilt unit to keep the end-effector in the center of the image.

A. Fusion Using Voting Schema

Voting has been widely used in machine vision in various forms, the most popular probably being the Hough transform [19]. In computer science, it has been used for reliable computing, where multiple sources were combined according to simple approval rules [9], [15]. Voting enables increased reliability of an integrated system consisting of a number of modules/cues where the reliability of each individual module varies significantly over time.

The main advantage of voting mechanisms is that they can operate "model-free" with respect to the individual cues. In probabilistic fusion, a model of the form $p(\text{cue}|\text{object})$ encodes the relationship between visual cues and particular objects/patterns. In voting, a very simple or no model is used for fusion. In principle, each estimator may be a simple classifier that votes for a particular attribute or against it (binary voting).

A common estimation/classification space or a (*voting domain*), Θ is usually introduced in voting where each cue estimator v_i is a mapping

$$v_i\colon \Theta \rightarrow [0;\ 1]. \tag{1}$$

The voting domain may for example be the 3-D space, the image plane or alternatively, the control space such as the joint space (R^6) of a manipulator. In terms of voting, there are several possible methods for selection/integration. If each of the n cue estimators (v_i) produce a binary vote for a single class (present/not present) a set of thresholding schemes can be used

Unanimity: $\sum v_i(\theta) = n$
Byzantine: $\sum v_i(\theta) > 2/3n$
Majority: $\sum v_i(\theta) > n/2$

where θ represents a particular class. If each cue estimator is allowed to vote for multiple classes, and the maximum vote is used to designate the final classification, the voting scheme is denoted consensus voting and the winning class θ' is chosen according to

$$\theta' = \arg\max_{\theta\in\Theta} \delta(\theta) \tag{2}$$

where δ is a combination method, which for example could be a simple addition of the votes or a weighting function that takes the relative reliability of the cues into account, i.e.,

$$\delta(\theta) = \sum_{i=1}^{n} w_i * v_i(\theta). \tag{3}$$

A general class of voting schemes, known as *weighted consensus voting*, is defined by the following definition.

Definition 1 (m-out-of-n voting): An m-out-of-n voting scheme, $V\colon \Theta \rightarrow [0,\ 1]$, where n is the number of cue estimators, is defined in the following way:

$$V(\theta) = \begin{cases} \Lambda(c_1(\theta),\ \ldots,\ c_n(\theta)), & \text{if } \sum_{i=1}^{n} v_i(\theta) \geq m \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where

$$v_i(\theta) = \begin{cases} 1, & \text{if } c_i(\theta) > 0 \\ 0, & \text{otherwise} \end{cases} \qquad \text{for } i = 1,\ \ldots,\ n \tag{5}$$

is the voting function and $\Lambda\colon [0;\ 1]^n \rightarrow [0;\ 1]$ is a function for combining the confidence for each estimator. □

In our implementation, a cue estimator can give a vote for a given class θ if the output of the estimator is >0. If m or more cues vote for a given class θ, the value is estimated using the fusion method Λ. If more than m cues are compatible, a weighted estimate is produced by the structure function Λ. The motivation for not using simple averaging is that the different cues might have different levels of uncertainty associated which can be taken into account by the fusion operator Λ.

Each cue estimator provides a binary estimate/answer. If we assume the probability of correct classification is p, then the estimator can be modeled as a Bernoulli random variable. The

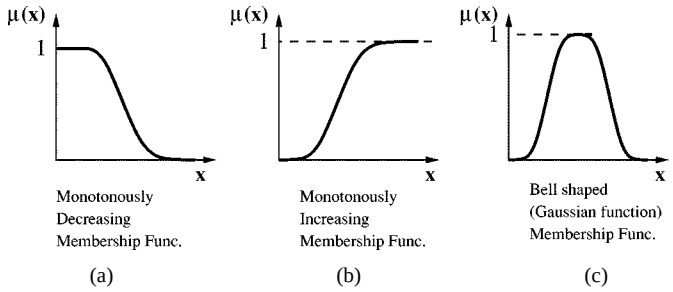

Fig. 3. Examples of membership functions—every fuzzy set can be represented by its membership function.

combined/fused estimate is a combination of $m-out-n$ votes which is a regular combination. The fused estimate ($V(\theta)$) will thus have a binomial distribution $B(n, m, p)$. The assumption of uniform detection probability is in general unrealistic, but it may be used for an overall design. A more detailed analysis can be used for design of the fusion mechanism.

We have implemented weighted consensus voting where two-out-of-five cues are needed for the fusion. The response of ith cue is weighted by a factor w_i where $\sum_{i=1}^{5} w_i = 1$. To enable the flexibility, the weights are changed dynamically depending on the performance measure of the individual cue. For each cue, the image position of the tracked region is estimated. As the performance measure, we use the difference between the position as estimated by individual cue and the position estimated by each of the integration techniques. The cues are weighted proportionally to this measure.

B. Fusion Using Fuzzy Logic

The main concept of fuzzy set theory is the notion of a fuzzy set $\mathbf{F}$, which is completely determined by the set of ordered pairs

$$\mathbf{F} = \{(x,\, \mu_F(x)) \mid x \in \Theta\} \tag{6}$$

where Θ denotes the universe of discourse for the set $\mathbf{F}$ and x is an element of Θ. The membership function μ_F (Fig. 3) gives a membership value $\mu_F(x)$ for each element x: μ: $\Theta \rightarrow [0;\ 1]$.

In our implementation, the membership function was a monotonously increasing function [see Fig. 3(b)]. The membership function for a cue and a particular image position was generated with a look-up table where the value of membership depended on a number of neighborhood pixels in a 5×5 kernel that gave a response to that particular cue (for 0 pixels occupied $\mu(x) = 0$, for 25 pixels occupied $\mu(x) = 1$). As the composition operator, we use the min–max operator defined by Zadeh [20]

$$\mu_{R_1 \circ R_2 \circ \cdots \circ R_n} = \max\{\min(\mu_{R_1},\, \mu_{R_2},\, \ldots \mu_{R_n})\} \tag{7}$$

where μ_{R_i} is the output of ith cue estimator.

III. Visual Cues

The following section presents visual cues used in the integration process (see Fig. 4).

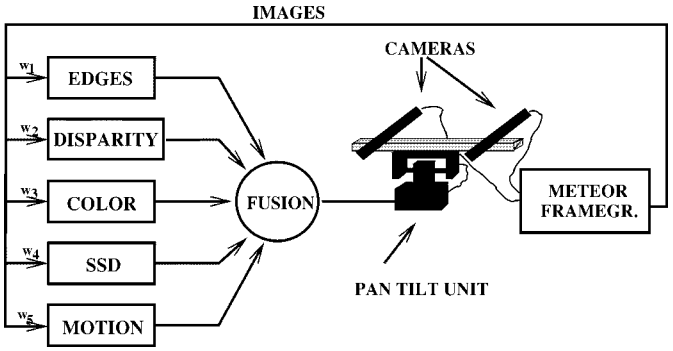

Fig. 4. Fusion mechanism.

A. Color-Based Segmentation

Color detection is based on the *hue* (H) and *saturation* (S) components of the color histogram values

$$H = \arccos\left[\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{((R-G)^2+(R-B)(G-B))}}\right]$$
$$S = 1 - \frac{3}{(R+G+B)}\min(R,\, G,\, B). \tag{8}$$

Color training was performed off-line, i.e., the *a priori* known color is used to compute its 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 belong to the tracked object.

B. Sum of Squared Differences (SSD)

An experimental study of different correlation methods, performed by Burt et al. [23] showed that a direct correlation method and SSD perform nearly as well as the more complicated methods (mean-normalized correlation, variance-normalized correlation, etc.). An abundance of efforts has been devoted to using different optimization techniques for speeding up the correlation [25], [24]. Motion estimation in general involves SSD minimization where the minimization process is performed by using a discrete search over an image region assuming *intensity constancy* between successive image frames

$$\mathbf{I}(\mathbf{x},\, t+1) = \mathbf{I}(\mathbf{x} - \mathbf{v}(\mathbf{x},\, \mathbf{p}),\, t) \tag{9}$$

where $\mathbf{x} = (x, y)$ is spatial image position of a point, $\mathbf{I}$ is the image intensity, $\mathbf{v}(\mathbf{x},\, \mathbf{p})$ denotes the image velocity at that point, and $\mathbf{p}$ is the number of parameters of the velocity model. Motion parameters are estimated by minimizing the residual

$$\epsilon = \iint [\mathbf{I}(\mathbf{x},\, t+1) - \mathbf{I}(\mathbf{x} - \mathbf{v}(\mathbf{x},\, \mathbf{p}),\, t)]^2 w(\mathbf{x})\, d\mathbf{x} \tag{10}$$

where the summation is performed along the feature window (region of interest) and $w(\mathbf{x})$ is a weighting function that is, in the simplest case, $w(\mathbf{x}) = 1$. In our implementation, we used a Gaussian-like function to reduce windowing effects. To speed up the process, we used optimization techniques as proposed

in [25]: loop short-circuiting, heuristic best place search beginning, and spiral image traversal pattern.

C. Edge Tracking

The main objective of our experiments is to track a robot's end-effector. We use a parallel jaw gripper and the most simple model is to consider the jaws as pairs of parallel lines. Therefore, a simple edge operator (Sobel) was used to detect clusters of parallel lines.

D. Blob Motion

In our implementation, image differencing is performed as the absolute difference of the intensity component (I) in consecutive images

$$M^{l,r}(\mathbf{X}) = \mathbf{F}(|I^{l,r}(\mathbf{x}, t) - I^{l,r}(\mathbf{x}, t-1)| - \Gamma) \qquad (11)$$

where Γ is a fixed threshold and $\mathbf{F}$ is the Heavyside function [32]. The scene is segmented into static and moving regions since only objects having a nonzero temporal difference change position between frames. Since the motion cue responds to all moving image regions we compensate for the egomotion of the camera head itself before computing the motion cue.

E. Disparity

The fundamental problem of disparity computation is finding the corresponding elements between two or more images. This can be done as proposed in [26], by matching small regions of one image to another based on intrinsic features of the region. These methods can be further classified depending on whether they match discrete features between images or correlate small area patches [27]. In our implementation we use combined corner and edge detector proposed by Harris and Stephens [28]. They proposed the following *corner response function* $\mathbf{R}$:

$$\mathbf{R} = \mathbf{Det}(\mathbf{M}) - k[\mathbf{Tr}(\mathbf{M})]^2 \qquad (12)$$

where k is a constant, $\mathbf{Det}$ is the determinant, and $\mathbf{Tr}$ is the trace of matrix $\mathbf{M}$, which is a 2×2 symmetric matrix of spatial derivatives

$$\begin{bmatrix} \Sigma I_x^2 & \Sigma I_x I_y \\ \Sigma I_x I_y & \Sigma I_y^2 \end{bmatrix} \qquad (13)$$

$\mathbf{R}$ is positive for a corner region, negative for an edge region, and small for a flat region [29]. We have used the knowledge of the epipolar geometry during the matching process together with uniqueness and ordering constraints. To get rid of the outliers, a maximum-likelihood cost function is used to find the most probable combination of disparities along a scan line.

IV. Visual Servo Control

The most commonly used approach in image-based visual servo control is to control the motion of the robot to servo the image plane features toward to a set of desired locations [3], or to achieve a defined relationship of feature characteristics [21], [22]. In general, for k feature points, the relationship between

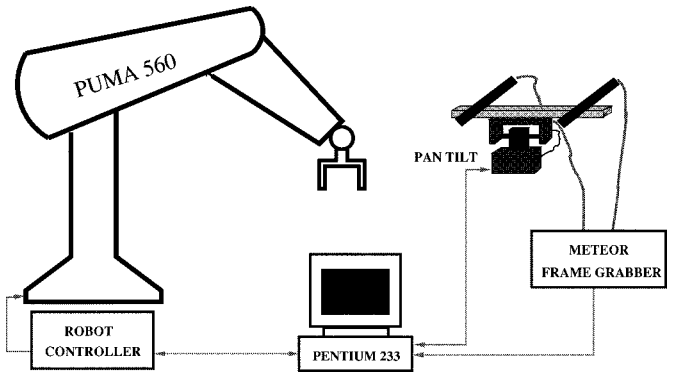

Fig. 5. Robotic workcell. In the present configuration, no specialized vision hardware was used.

feature velocities in image plane and velocities in camera frame are expressed by the following equation:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \vdots \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} \frac{f}{Z_1} & 0 & -\frac{x_1}{Z_1} & -\frac{x_1 y_1}{f} & \frac{f^2+x_1^2}{f} & -y_1 \\ 0 & \frac{f}{Z_1} & -\frac{y_1}{Z_1} & \frac{-f^2-y_1^2}{f} & \frac{x_1 y_1}{f} & x_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{f}{Z_k} & 0 & -\frac{x_k}{Z_k} & -\frac{x_k y_k}{f} & \frac{f^2+x_k^2}{f} & -y_k \\ 0 & \frac{f}{Z_k} & -\frac{y_k}{Z_k} & \frac{-f^2-y_k^2}{f} & \frac{x_k y_k}{f} & x_k \end{bmatrix} \cdot \begin{bmatrix} \mathbf{V_C} \\ \mathbf{\Omega_C} \end{bmatrix}. \qquad (14)$$

In our experiments, we assume a gripper that moves in Cartesian space. Translational and rotational velocities of the gripper frame origin are $\mathbf{V_O}$ and $\mathbf{\Omega_O}$, respectively. The relationship between two velocity screws: one expressed in the camera frame and second expressed in the robot frame is represented by

$$\begin{bmatrix} \mathbf{V_C} \\ \mathbf{\Omega_C} \end{bmatrix} = \Theta \begin{bmatrix} \mathbf{V_O} \\ \mathbf{\Omega_O} \end{bmatrix} \qquad (15)$$

where

$$\Theta = \begin{bmatrix} \mathbf{R} & -\mathbf{R}S(-\mathbf{R}^T\vec{t}) \\ 0 & \mathbf{R} \end{bmatrix} \qquad (16)$$

where S is a skew-symmetric matrix, $\mathbf{R}$ and $\vec{t}$ are the rotation matrix and translation vector associated with the gripper-to-camera homogeneous transformation. Combining (14) and (15), we obtain

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \mathbf{V_O} \\ \mathbf{\Omega_O} \end{bmatrix} \qquad (17)$$

where $\mathbf{J}$ is image Jacobian. The task is to move the robot in such a way that the image distance between the current position of a

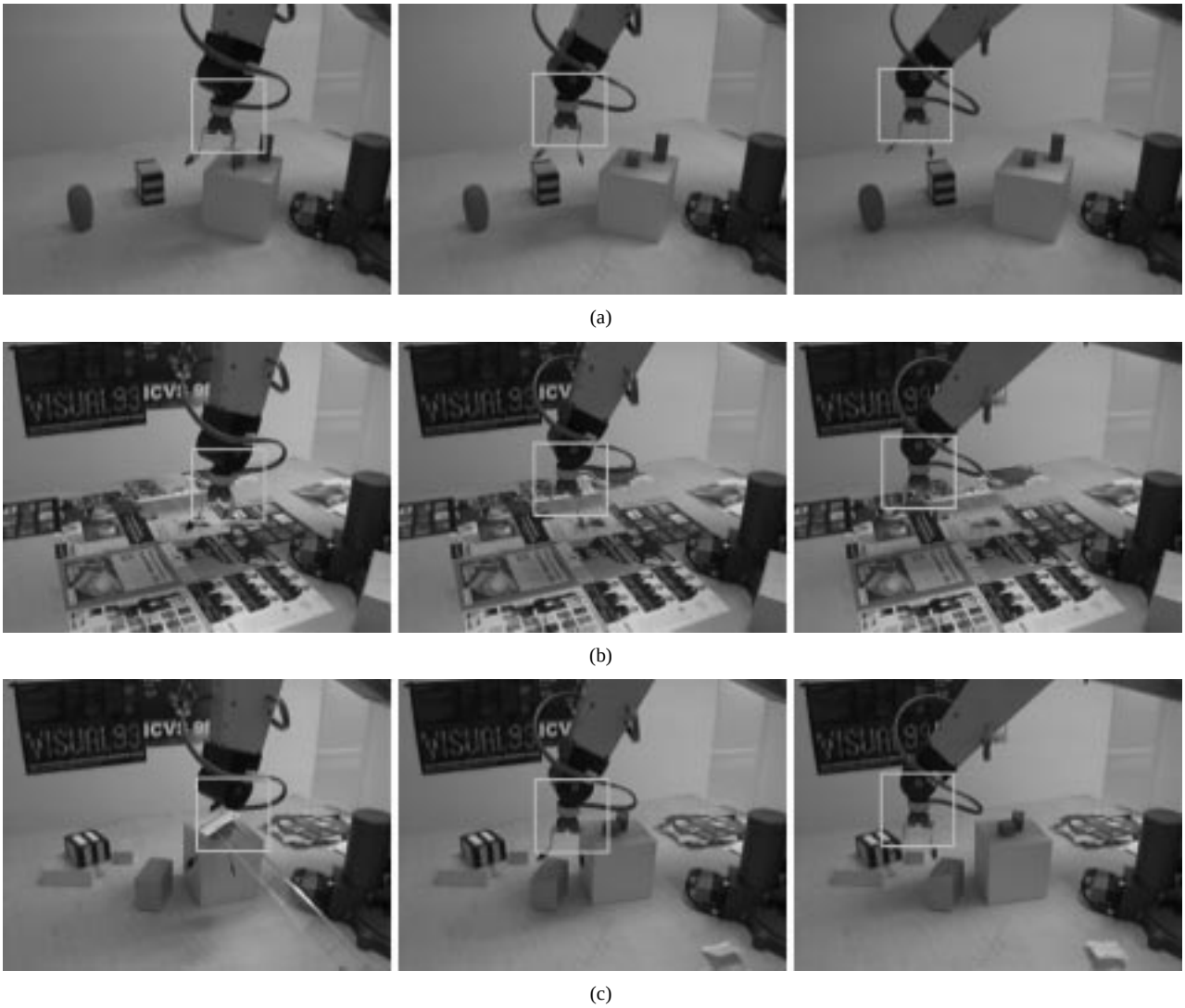

Fig. 6. Three configurations used for visual tracking experiments: (a) the configuration used in Experiment VI-A.1; (b) the configuration used in Experiment VI-A.2; (c) the configuration used in Experiment VI-A.3.

point $\vec{p}$ and the goal position $\vec{p}^*$ is minimized. Considering the image velocity of each point to be proportional to the difference vector between the current and the goal position, we can write

$$\dot{p} = K(\vec{p}^* - \vec{p}) \tag{18}$$

where K is a scalar that controls the convergence rate of the servoing. From (17) and (18) follows

$$\begin{bmatrix} \mathbf{V_O} \\ \mathbf{\Omega_O} \end{bmatrix} = K(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T(\vec{p}^* - \vec{p}). \tag{19}$$

Image Jacobian is usually not a square matrix and therefore the pseudoinverse (left or right) is used in order to compute the velocity screw of the manipulator [22].

Our tracking system is used to obtain continuous visual feedback in two different settings as follows: 1) in a tracking task where the objective was to keep the target in the center of the image, and 2) in a simple *pick-and-place* task.

V. System Setup

We used a pair of color CCD cameras arranged in a stereo setting as shown in Fig. 5. The cameras (mounted on a pan-tilt unit) view a robot manipulator (PUMA560) and its workspace from a distance of about 2 m. The optical axes of the cameras are parallel to each other with a baseline of 20 cm. The size of the original image is 320×240 pixels and the focal length of the cameras is 6 mm. The implemented system is running on a regular 233-MHz Pentium.

VI. Experimental Evaluation

The aim of the experiments was to investigate the hypothesis that fusion of multiple visual cues can lead to increased overall reliability of the tracking system compared to a system that employs just one of the cues.[1] For that purpose we have tested each cue separately as well as the proposed integration techniques in scenes with different level of clutter and complexity.

[1]This hypothesis has been proven in a different formulation in [33].

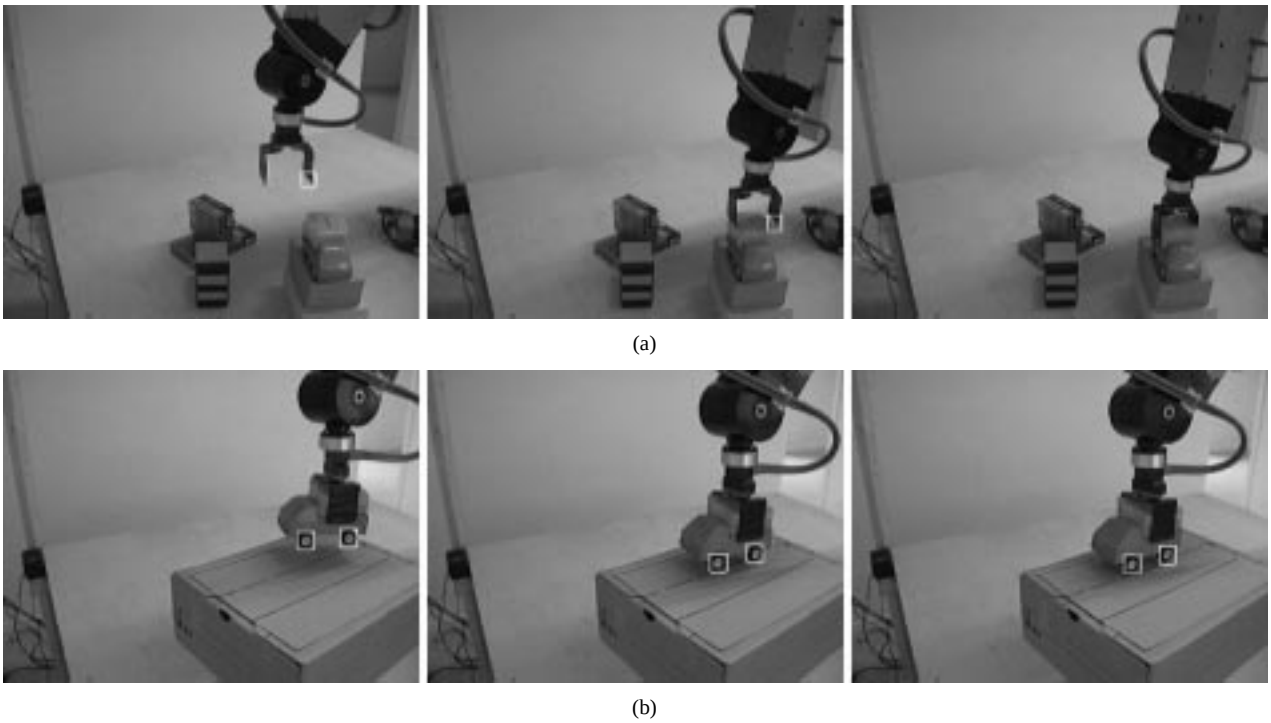

Fig. 7. Experimental setting. In the present configuration, no specialized vision hardware was used.

The tracking system was evaluated in two different settings as follows.

1) **Visual Tracking**: The camera system was used to actively track a robot end-effector with respect to various levels of background clutter and partial occlusion. Three different scenarios can be seen in Fig. 6. The objective of this experiment was to evaluate the performance of individual cues and proposed integration methods.
2) **Visual Servoing**: The objective of this experiment was to test the performance of the developed visual tracking system in a visual servo system. Voting-based integration was used for a pick-and-place servo task (see Fig. 7).

A. Visual Tracking

A control algorithm was designed in order to keep the end-effector in the center of the image as presented in the Appendix. A series of tracking experiments has been carried out. The results are presented for a sequence of 30 frames. In order to obtain the comparison between the different approaches, we first recorded few sequences of manipulator motion. Each sequence was then sampled so that every tenth frame was used. The size of the tracked region was 60×60 pixels.

We evaluated the performance of each cue estimator as well as the result obtained by voting and fuzzy logic-based integration. To present the quality of tracking we used the measure of the *relative error*. To express the relative error we used the distance (in pixels) between the ground truth position values (that were obtained off-line) and the position values obtained with cue estimators and the integration modules, respectively. To estimate the ground truth, we used the tool-center-point (TCP) to be our reference position. Three images from each sequence are shown in Fig. 6. The tracking performance for each of the cue estimators and the integrated cues are shown in Fig. 8. The results are presented and summarized in Tables I–III through the relative error using the mean and the standard deviation of the distance error.

Experiment 1: The manipulator was moving on a straight path for about 100 cm [Fig. 6(a)]. The distance of the manipulator relative to the camera was increasing from about 75 to 110 cm. The rotation of the sixth joint was 20°. The background was not cluttered but all the objects in the scene had the same color properties as the end-effector. The objective was to test the color segmentation and SSD in order to evaluate whether the mentioned cues perform better in a regular scene with a few challenges.

It is evident that most cues have a reasonable performance [Fig. 8(a)]. The relative error is presented in Table I. The disparity cue is the only cue with significant deviations (due to the lack of texture on the target object). From the error graph for X values, the uniform performance is apparent. From the error graph for Y values, we observe that the higher weight for the correlation cue implies that the voting scheme relies heavily on this particular cue, i.e., the correlation and voting-based fusion provide similar results. Overall tracking with a standard deviation of 2 pixels is very good. The bias trend in errors for the X values is due to the motion in depth. The fuzzy integration shows a larger bias, but as most of the cues are in agreement, the variation over the sequence is still not significant.

Experiment 2: A complex background will introduce strong differences between image intensities in consecutive image frames. This kind of setting will mostly affect the SSD since the background variations strongly influence the appearance of the template. The motion of the manipulator was as in the previous experiment.

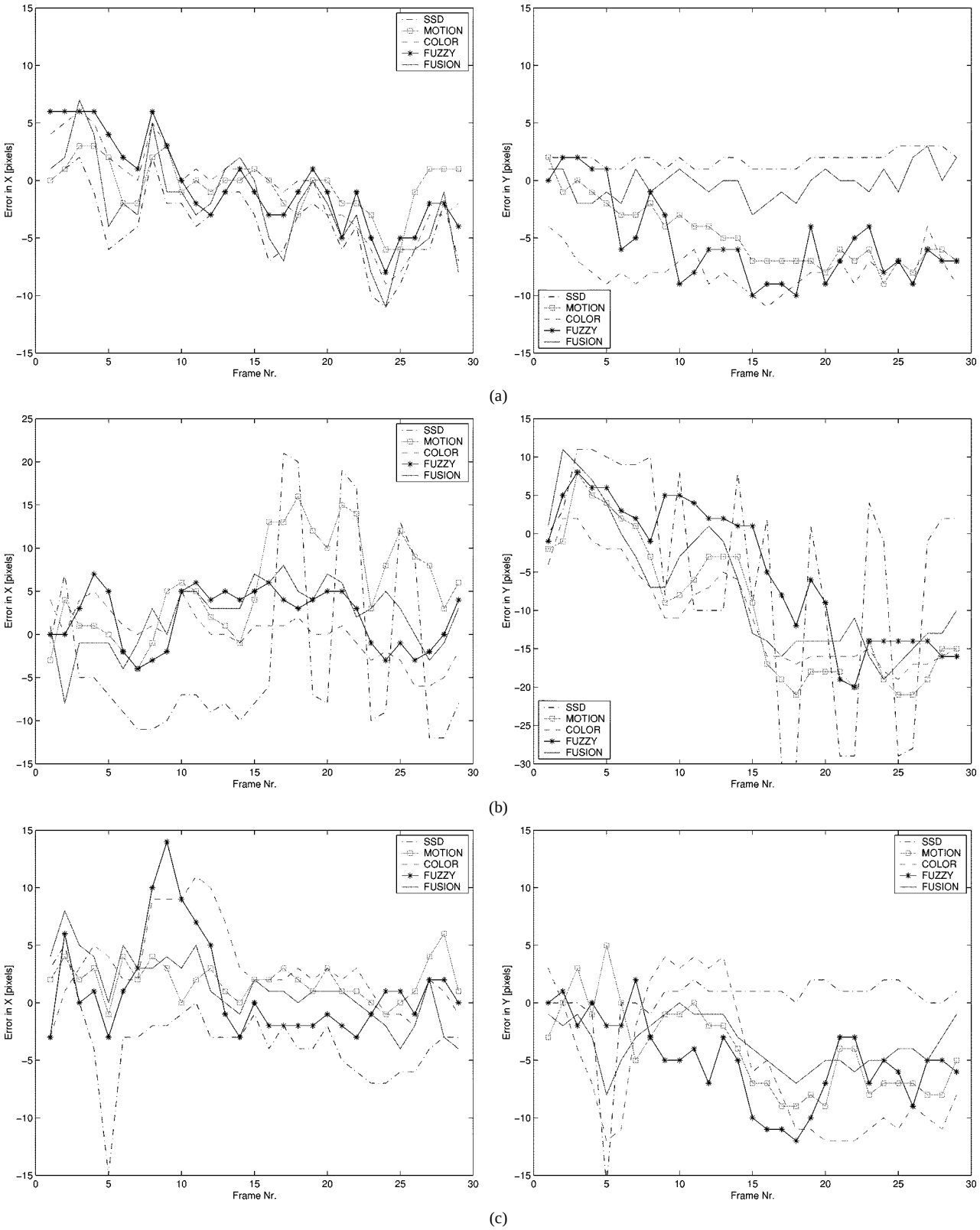

Fig. 8. Distance error in the X-horizontal (left image) and Y-vertical (right image) direction for the modules during (a) Experiment VI-A.1; (b) Experiment VI-A.2; and (c) Experiment VI-A.3. For visual clarity, the results from the edge and disparity cues are not presented. ("Frame Nr" represents the frame number. Section VI-A explains how image sequences were obtained.)

We introduced a significant background clutter in the scene [Fig. 6(b)]. The distance errors in X-horizontal and Y-vertical direction are presented in Table II and the performance is shown in Fig. 8(b). It is evident from the graphs that the correlation method is likely to fail in the cases where the variations in template appearance are significant. Although very good and fast cue, in the cases where the environment is challenging, this cue is not reliable enough. Significant deviations in the response of

TABLE I
MEAN DISTANCE ERROR AND STANDARD DEVIATION FOR EXPERIMENT IV-A.1

Module	Mean	STD
Color	**8.7411**	**1.5944**
Motion	**5.5291**	**2.3179**
Disp.	**15.9268**	**3.3047**
Edges	**5.0062**	**2.5509**
SSD	**4.7063**	**2.5613**
Voting	**4.0131**	**2.6590**
Fuzzy	**7.2744**	**2.0035**

TABLE II
MEAN DISTANCE ERROR AND STANDARD DEVIATION FOR EXPERIMENT IV-A.2

Module	Mean	STD
Color	**16.6770**	**3.6571**
Motion	**11.6329**	**4.1148**
Disp.	**26.5804**	**10.2971**
Edges	**12.7066**	**4.5489**
SSD	**17.4272**	**3.3322**
Voting	**5.5483**	**2.1347**
Fuzzy	**10.7648**	**4.0567**

TABLE III
MEAN DISTANCE ERROR AND STANDARD DEVIATION FOR EXPERIMENT IV-A.3

Module	Mean	STD
Color	**8.8805**	**3.1845**
Motion	**5.5900**	**2.5342**
Disp.	**13.7512**	**3.7664**
Edges	**6.5618**	**2.6815**
SSD	**4.2967**	**3.8108**
Voting	**4.9133**	**1.6249**
Fuzzy	**6.7763**	**3.4917**

this particular cue, reduces the weights assigned to this cue. A dynamic change of weights makes voting superior to fuzzy fusion which does not offer added robustness in this situation. This experiments shows an obvious need for a cue integration.

Experiment 3: The third experiment includes multiple moving objects and partial occlusion of the target [Fig. 6(c)]. The independent motion will confuse the simple motion detector, as it is used without a windowing function (obviously, it is possible to introduce validation gates to reduce this problem). The occlusion will at the same time result in failure for the correlation tracker, which again demonstrates the need for integration to ensure continuous tracking. The distance errors are presented in Table III and the performance in Fig. 8(c) where the error graphs clearly illustrate how correlation and color cue fail. It is evident here that this failure is propagated almost unattenuated through the fuzzy tracker, which indicates that fuzzy approach used here is inappropriate for the type of cue integration pursued in this work. For implemented weighted consensus voting, the consensus of most cues allows the system to cope with both types of error situations which is reflected in the limited variation over the image sequence and the best overall performance.

B. Visual Servoing

Based on the previous experimental evaluation, we chose the voting-based integration to design a tracking system which provided the visual feedback in a number of robotic tasks. Here, we present two tasks as follows.

1) **Pick-and-place task**: The task is to visually servo the robot to pick up an object (can) on the table [see Fig. 7(a)]. In the first frame, we manually initiate a point in the image (a point on the end-effector) and the region around this point is continuously tracked in left and right camera images during the servoing sequence. The size of the region was 5×5 pixels. The final position of the tracked point is known *a priori*. The error between the current and desired point position in the image is used to compute the velocity screw of the robot using (19). Since only three degrees of the robot are controlled, it is enough to track one point in each image in order to compute the image Jacobian [22].
2) **Positioning task**: The task is to visually servo the robot in order to place the wheels of the car parallel to the road [see Fig. 7(b)]. Again, we manually initiate the two points in each image which are then continuously tracked during the servoing sequence. The control is done in a similar fashion as in the previous example.

VII. CONCLUSION

Visual servoing has a tremendous potential for a wide variety of applications, but to gain acceptance a key factor is robust operation in realistic settings. This has been a major obstacle for widespread use of vision. Fusion of information to achieve robustness is a well-known technique. Most earlier approaches have used a Bayesian approach to integration as it provides a nice theoretical framework for analysis and design. Unfortunately, the Bayesian approach requires good models in terms of conditional probabilities and prior information that sometimes is hard to provide.

This paper has discussed methods for integration, which are based on weak or model-free approaches to integration. In particular, voting and fuzzy logic have been studied. The basic methods have been outlined and it has been described how the methods may be used for integration of visual cues for image-based tracking. In this context, the image space has been used for fusion of visual cues. For real-time visual servoing, a number of relatively simple cues have been used for experimentation to evaluate the potential utility of cues that individually are inadequate in realistic settings. The results show that integration using weak methods enables a significant increase in robustness. The methods clearly demonstrate that voting in particular provides performance that is better than any of the individual cues. The evaluation has included laboratory settings that pose a limited challenge and settings that have explicitly been designed to provide a low signal-to-noise ratio so as to generate unreliable cue estimates.

The presented methodology address the general problem of multicue figure ground segmentation. This problem only makes sense in the context of tasks and for well-defined objects. A problem that has not been addressed in this paper is model selection to enable segmentation, which for typical systems will require use of attention methods for selection and indexing. In addition, the presented method has assumed that a blob-like recognition scheme is adequate for tracking and servoing. In many tasks, more detailed models must be deployed to enable accurate pose control. Both of these problems are being addressed as part of our current research effort.

APPENDIX

A. Control Algorithm

Under perspective projection, the velocity of a point T in the camera frame can be associated to the velocity of a point $\dot{x}$ in the image plane as [22]

$$\dot{x} = \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} \frac{f}{Z} & 0 & \frac{-x}{Z} & \frac{-xy}{f} & \frac{f^2+x^2}{f} & -y \\ 0 & \frac{f}{Z} & \frac{-y}{Z} & \frac{-f^2-y^2}{f} & \frac{xy}{f} & x \end{bmatrix} T \quad (20)$$

where T represents velocity screw

$$T = [\, V_X \quad V_Y \quad V_Z \quad \omega_X \quad \omega_Y \quad \omega_Z \,]^T \quad (21)$$

and f is the focal length of the camera. The objective of our experiment was to control the pan-tilt unit in order to keep the target in the center of the image. We relate the differential change in the image coordinates with the differential change of pan and tilt angles in the following way:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} \frac{-xy}{f} & \frac{f^2+x^2}{f} \\ \frac{-f^2-y^2}{f} & \frac{xy}{f} \end{bmatrix} \begin{bmatrix} \omega_X \\ \omega_Y \end{bmatrix}. \quad (22)$$

The error signal is defined as the difference between the target current position in the image and the center of the image (x_0, y_0)

$$e = [\, x - x_0 \quad y - y_0 \,]^T \quad (23)$$

where x and y the image coordinates of the target. We used a proportional control to recenter the target in the image, defined as

$$\Delta q = [\, \Delta\alpha \quad \Delta\beta \,]^T = K_p e \quad (24)$$

where K_p is a constant.

REFERENCES

[1] E. D. Dickmanns, B. Mysliwetz, and T. Christians, "An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles," *IEEE Trans. Syst., Man, Cybern.*, vol. 37, pp. 1273–1284, Nov./Dec. 1990.

[2] H. Kollnig and H. Nagel, "3D pose estimation by directly matching polyhedral models to gray value gradients," *Int. J. Comput. Vis.*, vol. 23, no. 3, pp. 282–302, 1997.

[3] G. Hager, "Calibration-free visual control using projective invariance," in *Proc. Int. Conf. Computer Vision*, 1995, pp. 1009–1015.

[4] P. Allen, "Automated tracking and grasping of a moving object with a robotic hand–eye system," *IEEE Trans. Robot. Automat.*, vol. 9, p. 152, 1993.

[5] N. P. Papanikolopoulous and P. K. Khosla, "Adaptive robotic visual tracking: Theory and experiments," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 429–445, Mar. 1993.

[6] G. Hirzinger, M. Fischer, B. Brunner, R. Koeppe, M. Otter, M. Gerbenstein, and I. Schäfer, "Advanced in robotics: The DLR experience," *Int. J. Robot. Res,*, vol. 18, pp. 1064–1087, Nov. 1999.

[7] J. Clark and A. Yuille, *Data Fusion for Sensory Information Processing Systems*. Norwell, MA: Kluwer, 1990.

[8] J. Aloimonos and D. Shulman, *Integration of Visual Modules*. New York: Academic, 1989.

[9] P. Pirjanian, H. I. Christensen, and J. Fayman, "Application of voting to fusion of purposive modules: An experimental investigation," *Robot. Auton. Syst.*, vol. 23, no. 4, pp. 253–266, 1998.

[10] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA: MIT Press, 1987.

[11] L. Davis and A. Rosenfeld, "Cooperating processes for low-level vision: A survey," *Artif. Intell.*, vol. 17, pp. 245–263, 1981.

[12] I. Bloch, "Information combination operators for data fusion: A comparative review with classification," *IEEE Trans. Syst., Man, Cybern. A*, 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 behavior and performance," *IEEE Trans. Syst., Man, Cybern. A*, vol. 27, no. 5, pp. 553–568, 1997.

[14] C. Bräutigam, "A model-free voting approach to cue intagration," Ph.D. dissertation, Computational Vision and Active Perception Lab. (CVAP), Royal Inst. of Technol., Stockholm, Sweden, 1998.

[15] B. Parhami, "Voting algorithms," *IEEE Trans. Rel.*, vol. 43, no. 3, pp. 617–629, 1994.

[16] E. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man–Machine Studies*, vol. 7, pp. 1–13, 1975.

[17] Y. Li and C. Lau, "Development of fuzzy algorithms for servo systems," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1988.

[18] H. Borotsching and A. Pinz, "A new concept for active fusion in image understanding applying fuzzy set theory," in *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, 1996.

[19] J. Illingworth and J. Kittler, "A survey of the Hough transform," *Comput. Vis., Graph. Image Process.*, vol. 44, 1988.

[20] L. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 1, pp. 28–44, 1973.

[21] F. Chaumette, P. Rives, and B. Espiau, "Positioning a robot with respect to an object, tracking it and estimating its velocity by visual servoing," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1991.

[22] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 651–670, Oct. 1996.

[23] P. Burt, C. Yen, and X. Xu, "Local correlation measures for motion analysis—A comparative study," in *Proc. IEEE Conf. Pattern Recognition and Image Processing*, 1982, pp. 269–274.

[24] S. Nassif and D. Capson, "Real-time template matching using cooperative windows," in *Proc. IEEE Canadian Conf. Engineering Innovation: Voyage of Discovery*, vol. 2, 1997, pp. 391–394.

[25] C. Smith, S. Brandt, and N. Papanikolopoulos, "Eye-in-hand robotic tasks in uncalibrated environments," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 903–914, Dec. 1996.

[26] K. Konolige, "Small vision systems: Hardware and implementation," in *Proc. 8th Int. Symp. Robotics Research*, Oct. 1997.

[27] S. Barnard and M. Fischler, "Computational stereo," *Comput. Surv.*, vol. 14, no. 4, pp. 553–572, 1982.

[28] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conf.*, 1998.

[29] X. Zhang, "Detection of moving corners in dynamic images," in *Proc. Int. Symp. Industrial Electronics*, 1994, pp. 36–41.

[30] G. Hager and K. Toyama, "The XVision System: A general-purpose substrate for portable real-time vision applications," *Comput. Vis. Image Understanding*, vol. 69, no. 1, pp. 23–37, 1996.

[31] M. Knapek, R. S. Oropeza, and D. J. Kriegman, "Selecting promising landmarks," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 4, San Francisco, CA, Apr. 2000, pp. 3771–3777.

[32] J. Triesch and C. Von der Malsburg, "Self-organized integration of adaptive visual cues for face tracking," in *Proc. 4th IEEE Int. Conf. Automatic Face and Gesture Recognition*, 2000, pp. 102–107.
[33] K. Hashimoto, A. Aoki, and T. Noritsugu, "Visual servoing with redundant features," in *Proc. Conf. Decision and Control*, 1996, pp. 2483–2483.

Danica Kragić received the M.Sc. degree in mechanical engineering from Technical University of Rijeka, Croatia, in 1995. She is currently working toward the Ph.D. degree in computer science at the Computer Vision and Active Perception Laboratory and Centre for Autonomous Systems at the Royal Institute of Technology, Stockholm, Sweden. Her research interests include real-time vision, visual servoing, and data fusion.

Henrik I. Christensen (M'87) received the M.Sc. and Ph.D. degrees from Aalborg University in 1987 and 1989, respectively.

He is a Chaired Professor of Computer Science at the Royal Institute of Technology, Stockholm, Sweden. He is also the Director of the Centre for Autonomous Systems. He has also held appointments at Aalborg University, University of Pennsylvania, and Oak Ridge National Laboratory. He has published more than 130 papers on robotics, vision, and integration. His primary research interest is system integration for vision and mobile robotics.