CONTROLLED ACTIVE VISION AND VISION-BASED CONTROL OF ROBOTIC MANIPULATORS

Nikolaos P. Papanikolopoulos

Department of Computer Science
4-192 EE/CSci Building
200 Union Street S.E.
University of Minnesota
Minneapolis, MN 55455
npapas@cs.umn.edu

ABSTRACT

This paper presents the application of the *controlled active vision* framework to several problems of eye-in-hand robotic systems such as the derivation of depth maps from controlled motion; the vision-guided, automatic grasping of static or slowly moving objects; the active calibration of the robot-camera system; the problem of automatically detecting moving objects of interest; and the computation of the relative pose of the target with respect to the camera. All the algorithms are experimentally verified on the Minnesota Robotic Visual Tracker (a flexible eye-in-hand robotic system). We have applied these techniques to transportation applications (e.g., pedestrian detection and tracking, vision-based vehicle following), inspection, and assembly (e.g., vision-guided manipulation of moving objects). It should be stated that this work is not only limited to vision sensors. Our algorithms can be used in active sensing systems that include a variety of diverse sensors (e.g., force, tactile, sonar).

1. INTRODUCTION

In order to be effective, robotic agents in uncalibrated environments must operate in a flexible and robust manner. The computation of unknown parameters (e.g., the velocity of objects and the depth of object feature points) is essential information for the accurate execution of many robotic tasks, such as manipulation, inspection, and exploration. The determination of such parameters has traditionally relied upon the accurate knowledge of other related environmental parameters. For instance, traditional approaches to the problem of depth recovery have assumed that extremely accurate measurements of the camera parameters and the camera system geometry are provided a priori, making these methods useful in only a limited number of situations. Similarly, previous approaches to robotic visual tracking assumed known and accurate measures of camera parameters, camera positioning, manipulator positioning, target depth, target orientation, and environmental conditions. This type of detailed information is not always available or, when it is available, not always accurate. Inaccuracies are introduced by positioning, path constraints, changes in the robotic system, and changes in the operational environment. In addition, camera calibration and determining camera parameters can be computationally expensive, time consuming, and error prone. In particular, depth derivation and tracking techniques that rely upon stereo vision systems require careful geometry measurements and the solution of the correspondence problem, making the computational overhead prohibitive for real- or near-real-time systems. Furthermore, many structure-from-motion algorithms use simple accidental motion of the camera that does not guarantee the best possible identifiability of the depth parameter. To be effective in uncalibrated environments, the robotic agent must perform under a variety of situations when only simple estimates of parameters (e.g., depth, focal length, pixel size, etc.) are used and with little or no a priori knowledge about the target, the camera, or the environment.

One solution to these problems can be found under the *controlled active vision* framework [1]. Instead of relying heavily on a priori information, this framework provides the flexibility necessary to operate under dynamic conditions when many environmental and target-related factors are unknown and possibly changing. The *controlled active vision* framework is based upon adaptive controllers that utilize Sum-of-Squared Differences (SSD) optical flow measurements as inputs to the control loop. The SSD algorithm is used to measure the displacements of feature points in a sequence of images where the displacements may be induced by manipulator motion, target motion, or both. The measured displacements are compared to predicted displacements that are derived using the current parameter estimates in the adaptive controller. The errors from these comparisons are then used, in conjunction with previous measured displacements, to update parameter estimates and to produce the next control input. The control input is derived such that the amount of the error in the next iteration will be minimized given environmental- and manipulator-specific constraints. This type of adaptive control technique is useful under a variety of situations, including the application areas we have selected: depth recovery, vision-based grasping, calibration, automatic detection of objects of interest, and robotic visual tracking. To reduce the influence of workspace-, camera-, and manipulator- specific inaccuracies, an adaptive controller is utilized to provide accurate and reliable in-

formation regarding the depth of an object's feature points. This information may then be used to guide operations such as tracking, inspection, and manipulation. All the proposed algorithms are implemented on the Minnesota Robotic Visual Tracker (MRVT), an active vision testbed that integrates a traditional robotic manipulator (a Puma 560) with a state-of-the-art computer vision system to provide a unique and flexible sensor based robotic system.

2. PREVIOUS WORK

In accomplishing visual servoing the camera may either be mounted on the manipulator (if the camera is mounted on the manipulator's end-effector, the configuration is called eye-in-hand) or it may be statically located. In both these instances, the traditional approach to visual servoing has been to decouple the problems of obtaining information about the target (from the camera image) and manipulator control [2, 3, 4]. This approach is known as *look-and-move* approach. Since initial work in this area was typically hampered by the available computational power, it was obvious to see in physical experiments that a manipulator was looking and then moving (hence the name *look-and-move*). Recent advances in computers have made enough computational power available so that *look-and-move* is not very obvious in physical experiments even though the basic philosophy of separating the computer vision and manipulator control problems still persists. It is also worth mentioning that in visual servoing emphasis has been typically placed either on the computer vision processing component or the manipulator component.

Several research efforts have focused on using computer vision information in the dynamic feedback loop [5, 6, 7, 3, 8, 9, 10]. Weiss *et al.* [10] have proposed a model reference adaptive control scheme for robotic visual servoing. In this work servoing is performed with the goal of reducing the error between desired image attributes (center-of-mass, first or second moment of the image) and the current image attributes. The verification of the proposed algorithms has been limited to simulations. Allen [2] has proposed an approach that uses image-differencing techniques in order to track and grab a moving object. Distributed Kalman filter techniques as a solution to the visual tracking problem have been proposed by Brown *et al.* [11]. Koivo and Houshangi [12] have proposed an adaptive scheme for visually servoing a manipulator based on the information obtained by a static sensor. Feddema and Lee [7] have proposed a MIMO adaptive controller for eye-in-hand visual tracking. Their work has been used as the basis for our approach. Several other researchers [13] have proposed strategies for vision-based exploration. Finally, B. Ghosh [14] has addressed several vision-based robotic issues with the aid of a new "Realization Theory" for perspective systems.

The proposed work describes a methodology for integrating sensing (in this case a vision sensor) with control. Furthermore, our approach assumes only a partial knowledge of the mapping between the target and the camera. The adaptation mechanism is used to determine the mapping between the target and the camera. It is not used to determine the unknown dynamics of the manipulator. We assume that a manipulator controller has already been designed and is now available to us. So while we address the problem of using vision information in the dynamic feedback loop, our paradigm is different. Specifically, we claim that combining computer vision with control can result in better measurements (better tracking can improve low-level vision processing). It is in this context that we view our current work, which shows that noisy measurements from a vision sensor when combined with an appropriate control law can lead to an acceptable performance of a visual servoing algorithm. Our approach is dynamic because it incorporates the target's dynamics and kinematics in the system model as compared to the classical *look-and-move* approach which is static. In addition, the camera model and the noise characteristics of the computer vision measurements are included in the design of the robotic visual tracking strategy.

Using the *controlled active vision* framework, we have proposed algorithms [15, 16, 17] that address real-time robotic visual tracking of moving objects. To achieve this objective, computer vision techniques for detection of motion are combined with appropriate control strategies to compute the actuating signal for driving the manipulator. The problem is formulated from the system's theory point of view. We have introduced sophisticated use of multiple windows and numerically stable confidence measures in order to improve the accuracy of the computer vision measurements. The selection of the controller is based on the computer vision technique that is used for the computation of the displacement vector. For example, a large number of windows provides accurate measurements and thus, a simple controller is adequate. On the other hand, a small number of windows provides noisy measurements and therefore a stochastic controller must be used. In order to circumvent the need for explicitly computing the depth map of the target, adaptive control techniques are proposed. This is an important contribution of our work since our algorithms do not require accurate knowledge of the camera model and the environment. Moreover, there is no need for continuous calibration of the eye-in-hand system (at least for tracking and servoing tasks).

3. IMPACT OF THE CONTROLLED ACTIVE VISION FRAMEWORK

Vision-based control and active vision can have a significant impact on space applications, intelligent highways, manufacturing, and nuclear waste clean-up efforts. Vision-based control can enhance the performance of industrial robots in assembly lines, aid in better alignment of an object with the camera in automatic inspection systems, improve the automatic assembly of electronic devices (surface mount technology), assist in the realization of vehicle following (platooning), make possible autonomous satellite docking and recovery, and improve the efficiency of outdoor navigation techniques.

It is important to mention that there is an absence of a framework that covers all the issues that are introduced by integrating the vision sensor and more generally any sensor in the feedback loop of a robotic device. We think that there is a significant waste due to the fact that there is a trend to build systems that only address the use of specific sensing modules in the feedback loop. Small changes in the hardware or the software of

a specific sensing module require significant re-design of the whole system, thereby increasing the cost and the development time. We claim that the *controlled active vision* framework addresses many sensor-based control issues and provides a unified way of looking at problems of this type. It is not only limited to robotic visual servoing [18].

The idea behind all this is simple: "move the sensor in a controlled way in order to derive the best possible knowledge about the aspects of the environment relevant to the task, and then try to accomplish your task." The gains from this research can be summarized to significant reduction of hardware (e.g., we have shown that by efficiently using the motion of the eye-in-hand system we can achieve monocular full 3-D robotic visual tracking instead of using stereo), drastic reductions in software (you do not need significant amount of code for calibrating the eye-in-hand system since adaptive algorithms are used), and improved safety by introducing robust mechanisms for integrating the human operator in the feedback loop. Moreover, our algorithms do not require significant investment in image-processing hardware and can be immediately applied to any available robotic device. For example, the transfer of our algorithms from a direct-drive arm to a PUMA560 took two days. Since the algorithms are designed in order to use the existing robot controllers, there is no need for modifying the internal electronics of the industrial robots used.

4. DETAILS ABOUT THE CONTROLLED ACTIVE VISION FRAMEWORK

Sensory information enhances the robot's capability by continuously updating the robot's view (or model) of the world and the task. The completeness and accuracy of this view depends on the existence of a framework for the integration of sensory information with the other components of a robotic system. The proposed structure of such a framework is described in [1]. Its basic components are:

1. **Vision Sensor:** The vision sensor in this case is the CCD camera that is used in the experiments.
2. **Target:** The target is charactarized by a number of parameters that describe its structure and motion (Shape, Motion).
3. **Estimator:** The estimator continuously updates the values of the parameters of motion and of the robot's dynamical model based on the motion measurements and their associated variances.
4. **Manipulator Controller:** This module issues the control commands by taking into consideration the output of the estimator and the dynamical model of the robotic device.

To date, researchers have looked separately at the components of the above system. Some [12, 10] have dealt with the dynamics of the model (component 4) and some [19, 2, 20] with the vision algorithms (components 1, 2, and 3). Our approach is that we should look simultaneously at all the components of the above mentioned system. One cannot control a robotic device by ignoring the model of the camera and the available computer vision algorithms. Both, the robotic device and the computer vision algorithms play a significant role in the selection of the appropriate control scheme. For example, simple control (PI) strategies are adequate for accurate motion measurements while stochastic controllers are needed for noisy measurements. In addition, accurate camera modeling introduces a number of coefficients that have to be computed, either on-line or off-line, if satisfactory tracking performance is desired. On the other hand, the robotic manipulator is a device with limited tracking capabilities. If these constraints are ignored, the servoing/tracking algorithm will provide infeasible control commands. We view visual tracking and servoing and generally, active vision from a framework that pays attention to both the computer vision algorithms and the control techniques. This requires that new control algorithms be developed to confront the noise in the measurements and the nonlinear dynamic model of the robot, and that new computer vision techniques for motion detection be developed. Finally, we claim that combining control with computer vision can enhance the quality of the visual measurements and create new, flexible, and reliable robotic tracking devices.

The above framework also allows this system to be used for recovery of the object's shape information. Controlled movement of the manipulator can provide us with an estimation of the depth of a point. As an extension to this, controlled movement can be used for active calibration of the camera. In other words, we can use this technique to estimate the coefficients of the camera model.

The integration of computer vision with control has significant importance since it addresses both, theoretical and experimental research issues. On the theoretical side, it deals with issues such as sensory feedback, vision-based control, nonlinear control, interaction between vision and control, representation of motion, detection of motion, noise reduction, active vision, and selection of appropriate features for visual tracking. On the experimental side, it addresses issues such as system integration and real-time performance.

In addition to robotic visual servoing, the potential of the *controlled active vision* framework has been tested on the following tasks:

- **Automatic detection of servoing targets:** We use a *figure/ground* approach in order to detect objects of interest.
- **Vision-based robotic grasping:** We employ adaptive control and computer vision techniques in order to approach and grasp a static or slowly moving object.
- **Active calibration:** The idea is similar to the ones mentioned before. Based on the motion of the robot-camera system , we try to estimate the intrinsic parameters of the camera and the relative pose of the object with respect to the camera. This algorithm can be useful to the vision-based robotic tasks that require calibration of the eye-in-hand system.
- **Computation of shape through the controlled motion of the eye-in-hand system:** We derive depth maps by effectively using the controlled motion of the camera.

5. MATHEMATICAL FORMULATION OF THE PROBLEM

This section covers the mathematical model of our approach (the different tasks require slight modifications of the model). In particular, we assume a pinhole camera model with a frame $\{R_s\}$ attached to it. Consider a static target with a feature, located at a point $\mathbf{P}$ with coordinates (X_s, Y_s, Z_s) in $\{R_s\}$. The projection of this point on the image plane is the point $\mathbf{p}$ with image coordinates (x, y) given by

$$x = \frac{fX_s}{Z_s\gamma_x} \text{ and } y = \frac{fY_s}{Z_s\gamma_y} \tag{1}$$

where f is the focal length of the camera and γ_x, γ_y are the dimensions (mm/pixel) of the camera's pixels. In addition, it is assumed that $Z_s \gg f$. If (c_x, c_y) is the origin of the image coordinate system $\{F_s\}$ then

$$x_a = x + c_x \text{ and } y_a = y + c_y \tag{2}$$

where x_a and y_a are the actual image coordinates in $\{F_a\}$. Let us assume that the camera moves in a static environment with a translational velocity $\mathbf{T}=(T_x, T_y, T_z)^T$ and with an angular velocity $\mathbf{R}=(R_x, R_y, R_z)^T$ with respect to the camera frame $\{R_s\}$. The velocity of point $\mathbf{P}$ with respect to the $\{R_s\}$ frame is

$$\frac{d\mathbf{P}}{dt} = -\mathbf{T} - \mathbf{R} \times \mathbf{P}. \tag{3}$$

By taking the time derivatives of the expressions for x and y and using (1) and (3) we obtain:

$$u=x\frac{T_z}{Z_s} - \frac{fT_x}{Z_s\gamma_x} + \frac{xy\gamma_y}{f}R_x - (\frac{f}{\gamma_x} + \frac{x^2\gamma_x}{f})R_y + \frac{y\gamma_y}{\gamma_x}R_z \tag{4}$$

$$v=y\frac{T_z}{Z_s} - \frac{fT_y}{Z_s\gamma_y} + (\frac{f}{\gamma_y} + \frac{y^2\gamma_y}{f})R_x - \frac{xy\gamma_x}{f}R_y - \frac{x\gamma_x}{\gamma_y}R_z \tag{5}$$

where $u = \dot{x}$ and $v = \dot{y}$. The terms u and v are also known as the optical flow measurements. If we assume $\gamma_x = \gamma_y = f = 1$, equations (4) and (5) become:

$$u = x\frac{T_z}{Z_s} - \frac{T_x}{Z_s} + xyR_x - (1 + x^2)R_y + yR_z \tag{6}$$

$$v = y\frac{T_z}{Z_s} - \frac{T_y}{Z_s} + (1 + y^2)R_x - xyR_y - xR_z. \tag{7}$$

To keep the notation simple and without any loss of generality, in the mathematical analysis that follows, we use only the relations described by (6) and (7). Assume that the optical flow of the point $\mathbf{p}$ at time kT is $(u(kT), v(kT))$ where T is the time between two consecutive frames. It can be shown [4] that at time kT, the optical flow is:

$$u(kT)=\mu_x u_0(kT) + u_c(kT) \tag{8}$$

$$v(kT)=\mu_y v_0(kT) + v_c(kT) \tag{9}$$

where $u_c(kT)$ and $v_c(kT)$ are the components of the optical flow induced at the time instant kT by the servoing motion of the camera, and $u_0(kT)$ and $v_0(kT)$ are the components of the optical flow induced at the time instant kT by the possible motion of the target. The coefficients μ_x and μ_y are defined as:

$$\mu_x = \mu_y = \begin{cases} 1 & \text{Moving Target} \\ 0 & \text{Static Target} \end{cases} \tag{10}$$

Equations (8) and (9) will henceforth be used with k instead of kT. Equations (8) and (9) do not include any computational delays that are associated with the computation and the realization of the servoing motion of the camera. If we include these delays in the model, equations (8) and (9) are transformed to:

$$u(k)=\mu_x u_0(k) + u_c(k-d+1) = \mu_x u_0(k) + q^{-d+1}u_c(k) \tag{11}$$

$$v(k)=\mu_y v_0(k) + v_c(k-d+1) = \mu_y v_0(k) + q^{-d+1}v_c(k) \tag{12}$$

where d is the delay factor ($d \in \{1, 2, ...\}$) and q^{-1} is the backward shift operator [21]. For the time being, it is assumed that $d = 1$. From (6) and (7), $u_c(k)$ and $v_c(k)$ are given by:

$$u_c(k)=x(k)\frac{T_z(k)}{Z_s(k)} - \frac{T_x(k)}{Z_s(k)} + x(k)y(k)R_x(k) - [1 + x^2(k)]R_y(k) + y(k)R_z(k) \tag{13}$$

$$v_c(k)=y(k)\frac{T_z(k)}{Z_s(k)} - \frac{T_y(k)}{Z_s(k)} + [1 + y^2(k)]R_x(k) - x(k)y(k)R_y(k) - x(k)R_z(k). \tag{14}$$

We substitute $u(k)$ and $v(k)$ in (11) and (12) with an approximation to the optical flow, obtained by dividing the x and the y disparities ($(x(k+1) - x(k))$ and $(y(k+1) - y(k))$, respectively) by the time interval T. As a result, equations (11) and (12) can be written as:

$$x(k+1)=x(k) + Tu_c(k-d+1) + \mu_x Tu_0(k) + v_x(k) \tag{15}$$

$$y(k+1)=y(k) + Tv_c(k-d+1) + \mu_y Tv_0(k) + v_y(k) \tag{16}$$

where the white noise terms $v_x(k)$ and $v_y(k)$ are included to model the inaccuracies of the model (neglected accelerations, inaccurate robot control, etc.). $v_x(k)$ and $v_y(k)$ are white noise terms with variances σ_x^2 and σ_y^2, respectively.

For every feature point we obtain two equations that relate the new feature coordinates to the previous coordinates in terms of the sampling time (T) and the optical flow. Equations (15) and (16) can be represented compactly in matrix-vector form (also known as state-space form) as (the subscript $\mathbf{f}$ denotes the state-space description of a specific feature):

$$\mathbf{x_f}(k+1)=\mathbf{A_f}(k)\mathbf{x_f}(k) + \mathbf{B_f}(k-d+1)\mathbf{u_{con}}(k-d+1) + \mathbf{E_f}(k)\mathbf{u_f}(k) + \mathbf{H_f}(k)\mathbf{v_f}(k) \tag{17}$$

where $\mathbf{A_f}(k) = \mathbf{H_f}(k) = \mathbf{I_2}$, $\mathbf{E_f}(k) = T\ \text{diag}\{\mu_x, \mu_y\}$, $\mathbf{x_f}(k) \in R^2$, $\mathbf{u_{con}}(k) \in R^6$, $\mathbf{u_f}(k) \in R^2$, and $\mathbf{v_f}(k) \in R^2$. The matrix $\mathbf{B_f}(k) \in R^{2x6}$ is:

$$\mathbf{B_f}(k) = T \begin{bmatrix} \frac{-1}{Z_s(k)} & 0 & \frac{x(k)}{Z_s(k)} & x(k)y(k) & -(1 + x^2(k)) & y(k) \\ 0 & \frac{-1}{Z_s(k)} & \frac{y(k)}{Z_s(k)} & (1 + y^2(k)) & -x(k)y(k) & -x(k) \end{bmatrix}$$

The vector $\mathbf{x_f}(k) = (x(k), y(k))^T$ is the state vector, $\mathbf{u_{con}}(k) = (T_x(k), T_y(k), T_z(k), R_x(k), R_y(k), R_z(k))^T$ is the control input vector, $\mathbf{u_f}(k) = (u_0(k), v_0(k))^T$ is the disturbance vector, and $\mathbf{v_f}(k) = (v_x(k), v_y(k))^T$ is the white noise vector. The measurement vector $\mathbf{y_f}(k) = (y_x(k), y_y(k))^T$ for this feature is given by:

$$\mathbf{y_f}(k) = \mathbf{C_f x_f}(k) + \mathbf{w_f}(k) \tag{18}$$

where $\mathbf{w_f}(k) = (w_x(k), w_y(k))^T$ is a white noise vector ($\mathbf{w_f}(k) \sim N(\mathbf{0}, \mathbf{W})$) and $\mathbf{C_f} = \mathbf{I_2}$. The elements of the covariance matrix $\mathbf{W}$ are set to some constant values. Plausible estimates of these elements can be computed from the image. The measurement vector is computed using the Sum-of-Squared Differences (SSD) algorithm which is described in [16].

The state-space model for $N(N \geq 3)$ feature points can be written as:

$$\begin{aligned} x(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k-d+1)\mathbf{u_{con}}(k-d+1) \\ + \mathbf{E}(k)\mathbf{d}(k) + \mathbf{H}(k)\mathbf{v}(k) \end{aligned} \tag{19}$$

where $\mathbf{A}(k)$=$\mathbf{H}(k)$=$\mathbf{I_{2N}}$, $\mathbf{E}(k)$=$Tdiag\{\mu_x, \mu_y, ..., \mu_x, \mu_y\}$, $\mathbf{x}(k) \in R^{2N}$, $\mathbf{d}(k) \in R^{2N}$, and $\mathbf{v}(k) \in R^{2N}$. The matrix $\mathbf{B}(k) \in R^{2NX6}$ is:

$$\mathbf{B}(k) = \begin{bmatrix} \mathbf{B_f}^{(1)}(k) \\ ... \\ \mathbf{B_f}^{(N)}(k) \end{bmatrix}.$$

The superscript (j) denotes each one of the feature points ($(j) \in \{(1), ..., (N)\}$). The vector $\mathbf{x}(k) = (x^{(1)}(k), y^{(1)}(k), ..., x^{(N)}(k), y^{(N)}(k))^T$ is the new state vector, and $\mathbf{v}(k) = (v_x^{(1)}(k), v_y^{(1)}(k), ..., v_x^{(N)}(k), v_y^{(N)}(k))^T$ is the new white noise vector. The new measurement vector $\mathbf{y}(k) = (y_x^{(1)}(k), y_y^{(1)}(k), ..., y_x^{(N)}(k), y_y^{(N)}(k))^T$ for $N(N \geq 3)$ features is given by:

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{w}(k) \tag{20}$$

where $\mathbf{w}(k) = (w_x^{(1)}(k), w_y^{(1)}(k), ..., w_x^{(N)}(k), w_y^{(N)}(k))^T$ is the new white noise vector ($\mathbf{w}(k) \sim N(\mathbf{0}, \mathbf{W})$) and $\mathbf{C_f} = \mathbf{I_{2N}}$.

In the case that $\mu_x = \mu_y = 1$, we can combine equations (19)-(20) into a MIMO (Multi-Input Multi-Output) model (a similar model can be derived for the case $\mu_x = \mu_y = 0$):

$$\begin{aligned} (1 - 2q^{-1} + q^{-2})\mathbf{y}(k) = \mathbf{B}(k-d)\mathbf{u_{con}}(k-d) \\ -\mathbf{B}(k-d-1)\mathbf{u_{con}}(k-d-1) + \mathbf{n}(k) \end{aligned} \tag{21}$$

where $\mathbf{n}(k)$ is the white noise vector. The new white noise vector $\mathbf{n}(k)$ corresponds to the measurement noise, to the modeling errors, and to the noise introduced by inaccurate robot control. If we assume $\mathbf{B}(k-d) \approx \mathbf{B}(k-d-1)$ then (21) can be rewritten as a MIMO ARX (AutoRegressive with auXiliary input) model. This model consists of $2N$ MISO (Multi-Input Single-Output) ARX models. In addition, the new model's equation is:

$$(1-2q^{-1}+q^{-2})\mathbf{y}(k) = \mathbf{B}(k-d)\Delta\mathbf{u_{con}}(k-d)+\mathbf{n}(k) \tag{22}$$

where $\Delta\mathbf{u_{con}}(k-d)$ is defined as:

$$\Delta\mathbf{u_{con}}(k-d) = \mathbf{u_{con}}(k-d) - \mathbf{u_{con}}(k-d-1). \tag{23}$$

The control law in this case is

$$\begin{aligned} \mathbf{u_{con}}(k) = -[\hat{\mathbf{B}}^{\mathbf{T}}(k)\mathbf{Q}\hat{\mathbf{B}}(k) + \mathbf{L} + \mathbf{L_d}]^{-1}[\hat{\mathbf{B}}^{\mathbf{T}}(k)\mathbf{Q} \\ \{(d+1)\mathbf{y}(k) - \mathbf{y}^*(k+d) - d\mathbf{y}(k-1) \\ -d\hat{\mathbf{B}}(k-d)\mathbf{u_{con}}(k-d) + \sum_{m=1}^{m=d-1} \hat{\mathbf{B}}(k-m) \\ \mathbf{u_{con}}(k-m)\} - \mathbf{L_d}\mathbf{u_{con}}(k-1)] \end{aligned} \tag{24}$$

where $\hat{\mathbf{B}}(k)$ is the estimated value of the matrix $\mathbf{B}(k)$ and $\mathbf{y}^*(k)$ denotes the desired location of the features. This control law corresponds to the objective function ($\mathbf{Q}$, $\mathbf{L}$, and $\mathbf{L_d}$ are weighting matrices)

$$\begin{aligned} J(k+d) = E\{[\mathbf{y}(k+d) - \mathbf{y}^*(k+d)]^T\mathbf{Q} \\ [\mathbf{y}(k+d) - \mathbf{y}^*(k+d)] + \mathbf{u_{con}}^T(k)\mathbf{L} \\ \mathbf{u_{con}}(k) + \Delta\mathbf{u_{con}}^T(k)\mathbf{L_d} \\ \Delta\mathbf{u_{con}}(k)|F_k\} \end{aligned} \tag{25}$$

where the symbol $E\{X\}$ denotes the expected value of the random variable X and F_k is the sigma algebra generated by the past measurements and the past control inputs up to time k. The selection of the weighting matrices $\mathbf{Q}$, $\mathbf{L}$, and $\mathbf{L_d}$ allows one to place more or less emphasis on the control input, the control input change, and the servoing error when attempting to satisfy the control objective. Similar objective functions can be designed for a variety of tasks (e.g., dynamic repositioning, derivation of depth maps). The matrix $\hat{\mathbf{B}}(k)$ is dependent on the estimated values of the features' depth $\hat{Z}_s^{(j)}(k)$ ($(j) \in \{(1), ..., (N)\}$) and the coordinates of the features' image projections. In particular, the matrix $\hat{\mathbf{B}}(k)$ is defined as follows:

$$\hat{\mathbf{B}}(k) = \begin{bmatrix} \hat{\mathbf{B}}_\mathbf{f}^{(1)}(k) \\ ... \\ \hat{\mathbf{B}}_\mathbf{f}^{(N)}(k) \end{bmatrix}$$

where $\hat{\mathbf{B}}_\mathbf{f}^{(j)}(k)$ is given by:

$$\hat{\mathbf{B}}_\mathbf{f}^{(j)}(k) = T$$

$$\begin{bmatrix} \frac{-1}{\hat{z}_s^{(j)}(k)} & 0 & \frac{x^{(j)}(k)}{\hat{z}_s^{(j)}(k)} & x^{(j)}(k)y^{(j)}(k) & -1-(x^{(j)}(k))^2 & y^{(j)}(k) \\ 0 & \frac{-1}{\hat{z}_s^{(j)}(k)} & \frac{y^{(j)}(k)}{\hat{z}_s^{(j)}(k)} & 1+(y^{(j)}(k))^2 & -x^{(j)}(k)y^{(j)}(k) & -x^{(j)}(k) \end{bmatrix}.$$

The depth $\hat{Z}_s^{(j)}(k)$ for each individual feature is estimated by using the techniques described in [21, 22, 23]. The computed rotational and translational displacements are fed to a cartesian robot controller that provides the actuating torques. The use of this scheme allows us to successfully track a 3-D moving target in a poorly calibrated environment (the errors in the initial estimates of the depth related parameters and the focal length are 100% of the actual parameters' values). The algorithms are implemented on the Minnesota Robotic Visual Tracker (MRVT) system. The MRVT is

a multi-architectural system which consists of two main parts: the Robot/Control Subsystem (RCS) and the Vision Processing Subsystem (VPS). The RCS consists of a PUMA 560 manipulator, its associated Unimate Computer/Controller, and a VME-based Single Board Computer (SBC). The manipulator's trajectory is controlled via the Unimate controller's Alter line and requires path control updates once every 28 msec. Those updates are provided by an Ironics 68030 VME SBC running Carnegie Mellon University's CHIMERA real-time robotic environment. A Sun SparcStation 330 serves as the CHIMERA host and shares its VME bus with the Ironics SBC via BIT-3 VME-to-VME bus extenders. The VPS receives input from a Panasonic GP-KS102 miniature camera that is mounted parallel to the end-effector of the PUMA and provides a video signal to a Datacube system for processing. The Datacube is the main component of the VPS and consists of a Motorola MVME-147 SBC running OS-9, a Datacube MaxVideo20 video processor, a Datacube Max860 vector processor, and a BIT-3 VME-to-VME bus extender. The bus extender allows the VPS and the RCS to communicate via shared memory, eliminating the need for expensive serial communication. The VPS performs the computer vision algorithms, calculates the desired control input, and supplies the input vector via shared memory to the Ironics processor for inclusion as an input into the control software. The video processing and calculations required to produce the desired control input are performed under a pipeline programming model using Datacube's Imageflow libraries. We conducted multiple runs for the tracking of objects that exhibited unknown two- and three-dimensional motion with a coarse estimate of the depth of the objects. The results from these experimental runs can be found in [24].

6. INDIVIDUAL TASKS

6.1. Automatic Detection of Servoing Targets

While many researchers have focused on the control or computer vision aspects of the robotic visual servoing/tracking problem, few efforts have been reported in the area of automatic detection of moving objects of interest. We propose a scheme that uses a *figure/ground* approach. This approach is one of a family of image-differencing algorithms. When detection is considered, it is helpful to view an image as a set of pixels that belong to one of two categories: *figure* or *ground*. *Figure* pixels are those that are believed to belong to one of several objects of interest, while *ground* pixels belong to the surrounding environment. The proposed scheme maintains a *ground* image that represents the past history of the environment. For each pixel in the current image, a comparison is made to the corresponding pixel in the ground image. If the pixels differ by more than a threshold intensity amount, then the pixel is considered to be part of a binary *figure* image. We plan to improve the computational performance of the system with the use of *spontaneous* and *continuous* domains. Domains are areas in the image where objects of interest are expected to appear. Therefore, we may have to search only an extremely small part of the image.

6.2. Vision-based Robotic Grasping

We propose a flexible system based upon the *controlled active vision* framework that robustly grasps objects in the manipulator's workspace. The system operates in an uncalibrated space with an uncalibrated camera. The object of interest is not required to appear in a specific location, orientation, or depth, nor is it required to remain motionless during the grasp. Additionally, we decompose the motion into coarse and fine segments by using two different classes of object features during operation. We use the idea of "coarse" and "fine" points during the operation of the system to guide the manipulator's movements. Consider approaching a building that you wish to enter. At long distances, you use the building as a whole to guide your approach. This is similar to the use of coarse points in our system to guide the early, coarse movements. Once you are near enough to the building to identify the entrance, the entrance itself becomes the guiding feature, while the entirety of the building is ignored. This is similar to the use of fine points in our system. When the object dominates the field of view of the camera, fine points are used to guide the manipulator motion. Coarse points are selected while the object is at relatively far distances from the end-effector. The system automatically aligns the gripper with the object and forces the optical axis of the camera to pass through the centroid of the object. It then drives the manipulator toward the object while maintaining proper gripper and optical axis alignment. When the object is in motion, these alignment constraints result in the tracking of the object by the manipulator. When the coarse points approach the boundaries of the image plane, fine points are selected. These are used to drive the end-effector the remaining distance to the object and to signal when to grasp the object using a pneumatic, two-fingered hand. Proper orientation is maintained throughout by visual information derived from either the coarse points or the fine points, depending on the type of points being used to guide the manipulator.

The selection of features happens automatically based on the confidence measures reported in [16]. These confidence measures are also used on-line for the evaluation of the visual measurements that are provided by the Sum-of-Squared Differences (SSD) optical flow algorithm. Finally, experimental results from this approach can be found in [25].

6.3. Active Calibration

The controlled motion of the eye-in-hand system can be used in order to compute the pose of the target with respect to the camera frame. In addition, we are able to compute the piercing point of the optical axis (parameters c_x and c_y) and produce accurate estimations for the scaling (parameters f/γ_x and f/γ_y) and distortion factors. The whole problem can be modified in order to use the motion of the camera in a way that certain ill-conditioned sub-problems are simplified. The traditional approach is a static one and tries to compute several camera parameters from static views of the world. This problem which is closely related with the sensor-placement task has an impact on several areas such as satellite docking.

The computation of the pose can be modeled as a

MIMO (Multi-Input Multi-Output) estimation and control problem. Based on our current knowledge about the camera parameters and the camera position, we design a recursive estimation scheme. Moreover, the next position of the camera is computed by a control law that is based on the estimated and not the actual values of the system's parameters. Several experimental results from our approach are reported in [26]. The average errors for the scaling factors are 5%, for the image center parameters are 26%, and for the extrinsic parameters are 8%.

6.4. Computation of Shape Through the Controlled Motion of the Eye-in-hand System

In several inspection tasks, the shape of the object under inspection plays a significant role. Moreover, the shape of the object may indicate appropriate places for grasping it. We use a controlled motion of the eye-in-hand system in order to derive a depth map of the target. This approach helps us recognize its shape or moreover find deformations of its initial structure. Contrary to previous approaches [20, 19], we propose that we should move the eye-in-hand system in a controlled way if we want to have increased accuracy. In other words, our scheme creates a task function for each patch or pixel on the image. Then, we move the eye-in-hand system in a way that we guarantee that the depth parameter is identifiable. It should be mentioned that the design of the best possible trajectory (a set of $\mathbf{y}^*(k)$'s for every patch or pixel) is an interesting and difficult problem. While we are controlling the system in order to follow the desired trajectory for identification, we are estimating the depth of the patch or pixel with respect to the camera frame. As a result, a simple self-tuning controller can be designed for this task. In addition, since the computational delays are significant, the estimation scheme should be modified in order to include them. Otherwise, the control commands turn to be invalid. The control law that produces these image trajectories for each individual patch or pixel corresponds to the following objective function

$$J(k+d) = E\{[\mathbf{y}(k+d) - \mathbf{y}^*(k+d)]^T \mathbf{Q}[\mathbf{y}(k+d) - \mathbf{y}^*(k+d)] + \mathbf{u_{con}}^T(k)\mathbf{L}\mathbf{u_{con}}(k)|F_k\}. \quad (26)$$

The control law in this case is

$$\mathbf{u_{con}}(k) = -[\hat{\mathbf{B}}^{\mathbf{T}}(k)\mathbf{Q}\hat{\mathbf{B}}(k) + \mathbf{L}]^{-1}\hat{\mathbf{B}}^{\mathbf{T}}(k)\mathbf{Q}\{[\mathbf{y}(k) - \mathbf{y}^*(k+d)] + \sum_{m=1}^{m=d-1} \hat{\mathbf{B}}(k-m)\mathbf{u_{con}}(k-m)\}. \quad (27)$$

Since the identification of the depth often requires simple motions, we generally use simplified forms of the previous control law. As a result, the computation of the depth maps can be performed almost in real-time. Several experimental results can be found in [27]. The average error in the estimation of the depth is 2%. Matthies [19] has proposed an accidental motion in order to derive the structure of an object. We claim that small accidental motions of the camera may fail to produce displacements that will help the estimator to converge.

7. CONCLUSIONS

This paper discusses possible applications of the *controlled active vision* framework in the design of flexible and effective eye-in-hand robotic systems. The *controlled active vision* framework states that a controlled instead of an accidental motion of the camera can maximize the performance of any active vision algorithm. The techniques proposed in this paper provide ways for efficient operation of eye-in-hand robotic systems in uncalibrated environments. For example, we propose a scheme for vision-assisted grasping of static or slowly moving objects. For the computation of depth maps, we propose a technique that is based on the automatic selection of features and the design of specific trajectories on the image plane for each individual feature. Unlike similar approaches, this approach aids the design of trajectories that provide maximum identifiability of the depth parameter. During the execution of the specific trajectory, the depth parameter is computed with a simple estimation scheme that takes into consideration the previous movements of the camera and the computational delays. We have also studied the problems of the robotic visual tracking, the controlled active calibration of the robot-camera system, the computation of the relative pose of the target with respect to the camera frame, and the automatic detection of objects of interest. Our framework can be easily adapted to other applications, as demonstrated by the preliminary results of two transportation related applications (pedestrian tracking and vision-based vehicle-following [28]). All of the work presented in this paper has been implemented on the MRVT developed at the University of Minnesota, demonstrating the flexibility of the algorithms that have been developed.

8. ACKNOWLEDGEMENT

This work has been supported by the Department of Energy (Sandia National Laboratories) through Contracts #AC-3752D and #AL-3021, the National Science Foundation through Contracts #CDA-9222922 and #IRI-9410003, the McKnight Land-Grant Professorship Program, the Minnesota Department of Transportation through Contracts #71789-72983-169 and #71789-72447-159, and the Center for Transportation Studies at the University of Minnesota through Contract #USDOT/DTRS 93-G-0017-01.

9. REFERENCES

[1] N.P. Papanikolopoulos. *Controlled active vision.* PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, August 1992.

[2] P.K. Allen, B. Yoshimi, and A. Timcenko. Real-time visual servoing. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 851–856, April 1991.

[3] R.C. Luo, R.E. Mullen Jr., and D.E. Wessel. An adaptive robotic tracking system using optical flow. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 568–573, 1988.

[4] D. Tsakiris. Visual tracking strategies. Master's thesis, Department of Electrical Engineering, University of Maryland, 1988.

[5] A. Castano and S. Hutchinson. Hybrid vision/position servo control of a robotic manipulator. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1264–1269, May 1992.

[6] F. Chaumette and P. Rives. Vision-based-control for robotic tasks. In *Proc. of the IEEE International Worshop on Intelligent Motion Control*, pages 395–400, 20-22 August 1990.

[7] J.T. Feddema and C.S.G. Lee. Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera. *IEEE Trans. on Systems, Man, and Cybernetics*, 20(5):1172–1183, 1990.

[8] K. Hashimoto, T. Ebine, and H. Kimura. Dynamic visual feedback control for a hand-eye manipulator. In *Proc. of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '92)*, pages 1863–1868, July 7-10 1992.

[9] M.M. Trivedi, C. Chen, and S.B. Marapane. A vision system for robotic inspection and manipulation. *Computer*, 22(6):91–97, June 1989.

[10] L.E. Weiss, A.C. Sanderson, and C.P. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, 3(5):404–417, October 1987.

[11] C.M. Brown. Centralized and decentralized kalman filter techniques for tracking, navigation, and control. In *Proc. DARPA Image Understanding Workshop*, pages 651–675, 1989.

[12] A.J. Koivo and N. Houshangi. Real-time vision feedback for servoing of a robotic manipulator with self-tuning controller. *IEEE Trans. Systems, Man and Cybernetics*, 21(1):134–142, 1991.

[13] K.N. Kutulakos and C.R. Dyer. Recovering shape by purposive viewpoint adjustment. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 16–22, 1992.

[14] B.K. Ghosh. Image based estimation problems in system theory: motion and shape estimation of a planar textured surface undergoing a rigid flow. In *Proc. of the American Control Conference*, pages 1322–1326, 1993.

[15] N.P. Papanikolopoulos and P.K. Khosla. Adaptive robotic visual tracking: theory and experiments. *IEEE Trans. on Automatic Control*, 38(3):429–445, March 1993.

[16] N.P. Papanikolopoulos, P.K. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision. *IEEE Trans. on Robotics and Automation*, 9(1):14–35, February 1993.

[17] N.P. Papanikolopoulos and P.K. Khosla. Feature based robotic visual tracking of 3-d translational motion. In *Proc. of the 30th IEEE CDC, Brighton, UK*, pages 1877–1882, December 1991.

[18] M. Kume and P.K. Khosla. Using the controlled active vision paradigm in order to actively measure the radioactivity of a certain object. Technical Report CMU-RI-92-30, Carnegie Mellon University, Robotics Institute, 1992.

[19] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.

[20] J. Heel. Dynamic motion vision. In *Proc. DARPA Image Understanding Workshop*, pages 702–713, 1989.

[21] G.C. Goodwin and K.S. Sin. *Adaptive filtering, prediction and control*, volume 1 of *Information and Systems Science Series*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1984.

[22] P.S. Maybeck. *Stochastic models, estimation, and control*. Academic Press, London, 1979.

[23] N.P. Papanikolopoulos, B. Nelson, and P.K. Khosla. Full 3-d tracking using the controlled active vision paradigm. In *Proc. of the 1992 IEEE International Symposium on Intelligent Control (ISIC-92)*, pages 267–274, August 11-13 1992.

[24] S. Brandt, C.E. Smith, and N.P. Papanikolopoulos. The minnesota robotic visual tracker: a flexible testbed for vision-guided robotic research. In *Proc. of the 1994 IEEE Int. Conf. on Systems, Man, and Cybernetics*, pages 1363–1368, October 1994.

[25] C.E. Smith and N.P. Papanikolopoulos. Visually-guided, automatic grasping of static objects. Technical Report 94-059, Army High Performance Computing Research Center, University of Minnesota, 1994.

[26] R.H. Nelson. Evaluation of an active vision based method for focal length and image center determination. Master's thesis, Department of Computer Science, University of Minnesota, 1994.

[27] C.E. Smith and N.P. Papanikolopoulos. Computation of shape through controlled active exploration. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2516–2521, May 8-13 1994.

[28] C.E. Smith, S. Brandt, and N.P. Papanikolopoulos. Vision sensing for intelligent vehicle and highway systems. In *Proc. of the 1994 IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pages 784–791, October 1994.