Presented at 1991 American Control Conference (ACC)
Boston, MA, June 26-28 1991

ADAPTIVE ROBOTIC VISUAL TRACKING

N. Papanikolopoulos, P. K. Khosla, and T. Kanade

Department of Electrical and Computer Engineering
The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

Current robotic systems lack the flexibility of dynamic interaction with the environment. The use of sensors can make the robotic systems more flexible. Among the different types of sensors. visual sensors play a critical role. This paper addresses some of the issues associated with the use of a visual sensor in the feedback loop. In particular, algorithms are proposed for the solution of the robotic (hand-eye configuration)visual tracking and servoing problem We state the problem of robotic visual tracking as a problem of combining control with computer vision. We propose the use of sum-of-squared differences (SSD) optical flow for the computation of the vector of discrete displacements. These displacements are fed to an adaptive controller (*self-tuning regulator*) that drives the robot in conjunction with a cartesian robotic controller. We have implemented three different adaptive control schemes and the results are presented in this paper.

1. Introduction

One of the most desirable characteristics of a robotic manipulator is its flexibility. Flexible robots can adapt quickly to the evolving requirements of an unknown task, they can recover successfully from hardware failures, and can react properly to sudden changes in the environment. Flexibility and adaptability can be achieved by incorporating sensory information from multiple sources in the feedback loop. This paper addresses the use of vision sensor for dynamically servoing a manipulator for object tracking.

The problem of robotic visual tracking/servoing can be defined as:"move the manipulator (the camera is mounted on the end-effector) in such a way that the projection of a moving or static object is always at the desired location in the image". The solution to this problem can be viewed as a paradigm of the *controlled active vision* framework introduced in [1]. The underlying philosophy of this framework is that controlled and not accidental motion of the camera can enhance the efficiency of the vision algorithms thereby increasing the amount and quality of sensory information.

Research in computer vision has traditionally emphasized the paradigm of image understanding. However, some work has been reported towards the use of vision information for tracking [2, 3, 4, 5, 6]. In addition. some research [7, 8] has been conducted in using vision information in the dynamic feedback loop. While we address the problem of using vision information in the dynamic feedback loop, our paradigm is slightly different. Specifically, we claim that combining vision with control can result in better measurements. 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.

We propose algorithms that address the 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. An advantage of this approach is that the dynamics of the robotic device can be taken into account without changing the basic structure of the system. We introduce algorithms for incorporating color information, sophisticated use of multiple windows. and numerically stable confidence measures in order to improve the accuracy of the vision measurements, In order to circumvent the need to explicitly compute the depth map of the target, adaptive control techniques are proposed. The experimental results show that the proposed system performs satisfactorily even with noisy measurements and adapts well to the changes in the movement of the object.

The organization of this paper is as follows: Section 2 describes the vision techniques (optical flow, confidence measures) used for the computation of the object's motion parameters. The mathematical formulation of the visual tracking problem is described in Section 3. The adaptive control strategies are discussed in Section 4. Section 5 presents the robot control scheme used in the experiments. The experimental results are presented in Section 6. Finally, in Section 7, the paper is summarized. The next Section describes how the vision sensor detects and measures the target's motion.

2. Visual Measurements

An object in an image consists of brightness patterns. As the object moves in 3-D space. the brightness patterns in the image move simultaneously. Horn [9] defines the *optical flow* as "the apparent motion of the brightness patterns". For rigid objects the optical flow corresponds well to the motion field. We assume a pinhole camera model with a frame R, attached to it. We also assume a perspective projection and the focal length to be unity. A point P with coordinates (X_s, Y_s, Z_s) in R_s projects onto a point p in the image plane with image coordinates (x, y) given by:

$$x = X_s / Z_s \quad \text{and} \quad y = Y_s / Z_s. \tag{1}$$

Equation (1) gives the ideal x and y. If we define two scaling factors γ_x, γ_y to account for camera sampling and if (c_x, c_y) is the origin of the image coordinate system F_s then:

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

where x_a and y_a are the actual image coordinates. To keep the notation simple and without any loss of generality. in the mathematical analysis that follows. we use only the relations described by (1). Any displacement of a rigid object can be described by a rotation about an axis through the origin and a translation. If the angle of this rotation is small, the rotation can be characterized by three independent rotations about the X, Y and Z axes. 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 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:

$$u = [\, x\frac{T_z}{Z_s} - \frac{T_x}{Z_s}\,] + [\, xyR_x - (1+x^2)R_y + yR_z\,] \quad (4)$$

$$v = [\, y\frac{T_z}{Z_s} - \frac{T_y}{Z_s}\,] + [\,(1+y^2)R_x - xyR_y - xR_z\,] \quad (5)$$

where $u = \dot{x}$ and $v = \dot{y}$. u and v are also known as the optical flow measurements. Now, instead of assuming a static object and a moving camera, if we were to assume a static camera and a moving object then we would obtain the same result as in **(4)** and **(5)** except for a sign reversal. The computation of u and v has been the focus of much research and many algorithms have been proposed [10, 11]. For accuracy reasons, we use a modified version of the matching based technique [12] also known as the sum-of-squared differences **(SSD)** optical flow. For every point $p_A=(x_A, y_A)$ in image A, we want to find the point $p_B = (x_A+u, y_A+v)$ to which the point p_A moves in image B. It is assumed that the intensity in the neighborhood $\mathcal{L}$ of p_A remains almost constant. that the point p_B is within an area S of p_A, and that velocities are normalized by tune T to get the displacements. Thus. for the point p_A the SSD estimator selects the displacement $d=(u,v)$ that minimizes the SSD mcasurc:

$$e(p_A, d) = \sum_{m,n \in N} [I_A(x_A+m, y_A+n) - I_B(x_A+m+u, y_A+n+v)]^2 \quad (6)$$

where $u, v \in S$, N is an area around the pixel we are interested in, and I_A, I_B are the intensity functions in image A and B, respectively. The different values of the SSD measure create a surface called the *SSD surface.* By using sub-pixel fitting and multi-grid techniques, the accuracy of the **SSD** technique can be improved but at the cost of increasing its computational complexity. The accuracy can also be improved by selecting an appropriate small area N and by having velocity fields with few quantization levels.

The accuracy of the measurements of the displacement vector can also be improved by using multiple windows. The selection of them is discussed in detail in [1]. The next step in our algorithm involves the use of these measurements in the visual tracking process. These measurements should be transformed into control commands to the robotic system. Thus. a mathematical model for this transformation must be developed. In the next Section, we present the mathematical model for the visual tracking problem.

3. Modeling of the Visual Tracking problem

3.1. Visual Tracking of a Single Feature Point

Consider a target that moves in a plane with a feature. located at a point P, that we want to track. The projection of this point on the image plane is the point p. Consider also a neighborhood S_w of p in the image plane. The problem of **2-D** visual **tracking** of a single feature point can be defined as: "find the camera translation (T_x, T_y) with respect to the camera frame that keeps S_w stationary in an area S_o around the origin of the image frame". It is assumed that at initialization of the tracking process. the area S_w is brought to the origin of the image frame, and that the plane of motion is perpendicular to the optical axis of the camera. The problem of visual tracking of a single feature point can also he defined as "find the camera rotation (R_x, R_y) with respect to the camera frame that keeps S_w stationary in an area S_o around the origin of the image frame". The second definition does not require the cornputation of the depth Z_s of the point **P.** Assume that the optical flow of the point p at the instant of time kT is $(u(kT), v(kT))$ where T is the time between two consecutive frames. It can be shown that at time $(k+1)$ T, the optical flow is:

$$u((k+1)T) \approx u(kT) + u_c((k-d)T) \quad (7)$$

$$v((k+1)T) = v(kT) + v_c((k-d)T) \quad (8)$$

where $u_c((k-d)T)$, $v_c((k-d)T)$ are the components of the optical flow induced by the [racking motion of the camera, and d is the delay factor. For the time being. the delay factor is assumed to be zero. Equations (7) and (8) are based on the assumption that the optical flow induced by motion of the feature docs not change in the time interval T. Therefore, T should be as small as possible. To keep the notation simple and without any loss of generality, equations (7) and (8) will be uscd with k and (k+ 1) instead of kT and (k+ 1) T respectively. If the camera tracks the feature point with translation $T_x(k)$ and $T_y(k)$ with respect to the camera frame, then the optical flow that is generated by the motion of the camera with $T_x(k)$ and $T_y(k)$ is:

$$u_c(k) = -\frac{T_x(k)}{Z_s}, \quad v_c(k) = -\frac{T_y(k)}{Z_s} \quad (9)$$

We assume that for **2-D** visual tracking the depth Z_s remains constant. From (7)-(9), the optical flow equations for the translational case of the visual tracking are:

$$u(k+1) \approx u(k) - \frac{T_x(k)}{Z_s} \quad (10)$$

$$v(k+1) \approx v(k) - \frac{T_y(k)}{Z_s} \quad (11)$$

When the tracking motion of the camera is rotation with $R_x(k)$ and $R_y(k)$, the optical flow induced by the moving camera is:

$$u_c(k) = R_x(k)x(k)y(k) - R_y(k)[x^2(k)+1] \quad (12)$$

$$v_c(k) = R_x(k)[y^2(k)+1] - R_y(k)x(k)y(k) \quad (13)$$

It is known that:

$$u(k+1) = \frac{x(k+1)-x(k)}{T}, \quad v(k+1) = \frac{y(k+1)-y(k)}{T} \quad (14)$$

If we substitute $u(k+1)$ and $v(k+1)$ in (7) and (8) with their equivalent expressions from (14), then equations (7) and (8) can he written as:

$$x(k+1) \approx x(k) + Tu_c(k) + Tu(k) \quad (15)$$

$$y(k+1) \approx y(k) + Tv_c(k) + Tv(k) \quad (16)$$

Further, if we model the inaccuracies of the model (neglected accelerations) as white noise, (15) and (16) become:

$$x(k+1) = x(k) + Tu_c(k) + \mathrm{Tu}(k) + v_1(k) \quad (17)$$

$$y(k+1) = y(k) + Tv_c(k) + Tv(k) + v_2(k) \quad (18)$$

where $v_1(k), v_2(k)$ are zero-mean, mutually uncorrclatcd. stationary random variables with variances σ_1^2 and σ_2^2, respectively The above equations can be written in the state-space form as:

$$\mathrm{x}(k+1) = \mathbf{A}\,\mathbf{x}(k) + \mathbf{B}\,\mathbf{u}_c(k) + \mathrm{E}\,\mathrm{d}(k) + \mathbf{H}\,\mathrm{v}(k) \quad (19)$$

where' $\mathbf{A} = \mathbf{H} = \mathbf{I}_2$, $\mathbf{B} = \mathbf{E} = T\,\mathbf{I}_2$, $\mathbf{x}(k) \in R^2$, $\mathbf{u}_c(k) \in R^2$, $\mathbf{d}(k) \in R^2$ and $\mathbf{v}(k) \in R^2$. The vector $\mathbf{x}(k) = (x(k), y(k))^T$ is the state vector, $\mathbf{u}_c(k) = (u_c(k), v_c(k))^T$ is the control input vector. $\mathrm{d}(k) = (u(k), v(k))^T$ is the exogenous disturbances vector, and $\mathrm{v}(k) = (v_1(k), v_2(k))^T$ is the white noise vector. The measurement vector $y(k) = (y_1(k), y_2(k))^T$ is given by:

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

where $\mathbf{w}(k) = (w_1(k), w_2(k))^T$ is a white noise vector ($\mathrm{w}(k) \sim N(0, W)$) and $\mathbf{C} = \mathbf{I}_2$. The measurement vector is computed using the SSD algorithm described in Section **2.** The same modcl can be used for keeping the feature point stationary in an area S_r different from the origin. Assume (r_x, r_y) is the center of this arca S_r. Wc can transform the state variables $x(k)$ and $y(k)$ as $x_N(k) = x(k) - r_x$ and $y_N(k) = y(k) - r_y$, and the previous model still holds. The matrices $\mathbf{A}, \mathbf{B}, \mathrm{C}, \mathrm{E}, \mathrm{H}$ remain unchanged under the transformation. Since the **SSD** algorithm continuously computes the displacement vcctor of the feature point from its desired position, we have the ability to compensate for previous measurement errors that tend to accumulate.

*The symbol I_n denotes the identity matrix of order n

3.2. Visual Tracking of an Object

Consider a target that moves in a plane which is perpendicular to the optical axis of the camera. The projection of the target on the image plane is the area S_w in the image plane. The problem of 2-D visual tracking of a single object can be defined as: "find the camera translation (T_x, T_y) and rotation (R_z) with respect to the camera frame that keeps S_w stationary". It is assumed that the target rotates around an axis Z which at k=0 coincides with the optical axis of the camera. The mathematical model of this problem in state-space form is (a formal derivation is given in [1]):

$$\mathbf{x}(k+1) = A\,\mathbf{x}(k) + \mathbf{B}\,u_c(k) + E\,d(k) + H\,v(k) \tag{21}$$

where $A = H = \mathbf{I}_3$, $\mathbf{B} = \mathbf{E} = T\,\mathbf{I}_3$, $\mathbf{x}(k) \in R^3$, $\mathbf{u}_c(k) \in R^3$, $\mathbf{d}(k) \in R^3$ and $\mathbf{v}(k) \in R^3$. The vector $\mathbf{x}(k) = (x(k), y(k), \theta(k))^T$ is the state vector. $\mathbf{u}_c(k) = (u_c(k), v_c(k), R_z(k))^T$ is the control input vector, $d(k) = (u(k), v(k), \omega(k))^T$ is the exogenous disturbances vector. and $\mathbf{v}(k) = (v_1(k), v_2(k), v_3(k))^T$ is the white noise vector. $x(k)$, $y(k)$, $\theta(k)$ are now the X. Y and roll component of the tracking error, respectively. The measurement vector $y(k) = (y_1(k), y_2(k), y_3(k))^T$ is given by:

$$y(k) = C\,x(k) + w(k) \tag{22}$$

where $\mathbf{w}(k) = (w_1(k), w_2(k), w_3(k))^T$ is a white noise vector ($w(k) \sim N(0, W)$) and $C = \mathbf{I}_3$. The measurement vector is obtained in a slightly different way than in the case of the visual tracking of a single feature point. First, the tracking error of the projections of the two different feature points on the image plane is computed by using the SSD algorithm. Then. an algebraic system of four equations (two tracking error equations per point) is formulated. The solution of this is the X, Y and roll component of the tracking error. If the projections of the two feature points on the image plane are not the same, it is guaranteed that the system of equations has a solution. It is assumed that each one of these features at time $t = 0$ is located at its desired position. If the features at $t = 0$ are not at their desired position, we should use a different vision technique for the recovery of the displacements' vectors. An efficient algorithm for this type of computation is the one that calculates the coordinates of the centroid of the object's projection on the image plane. This technique can be combined with the previously mentioned strategies to keep continuously the target stationary at the desired position and orientation. The optical flow technique can take care of the target tracking after the target has been moved to the desired position and orientation while the centroid calculation permits successful initialization of the tracking process. The control strategies that keep the target stationary in both cases are discussed in detail in the next Section.

4. Design of Adaptive Robotic Visual Controllers

The control techniques that will be presented for the 2-D visual tracking of a moving object can also be used for visual tracking of a single moving feature point. The mathematical models for the two cases (feature. object) that were developed in the previous section have the same structure. The only differences are in the order of the systems and in the way that we obtain the measurement vector. The control objective is to minimize at each instant of time the error vector $\mathbf{e}(k) = (x(k) - 0, y(k) - 0, \theta(k) - 0)^T$ by choosing an appropriate control input vector $\mathbf{u}_c(k)$. In [13], we have presented various control techniques (LQG, PI, Pole Assignment) that are appropriate for the robotic visual tracking problem. Adaptive control techniques can be used for 2-D visual tracking of either a feature or an object when the depth information is not directly available. The adaptive control techniques are used only for the recovery of the $T_x(k)$ and $T_y(k)$. When the object rotates around the optical axis of the camera, all the coefficients of the equation that gives $R_z(k)$ are known. Thus, there is no need for an estimation scheme. These adaptive control techniques are based on the estimated and not the actual values of the system's parameters. A large number of algorithms can be generated, depending on which parameter estimation scheme is used and which control law is chosen. The rest of the section will be devoted to the detailed description of the system model, the estimation schemes, and finally, the selection of the control law.

4.1. System Model

The state-space model in (19) and (20) can be transformed to the following equations in the case of 2-D visual tracking with $T_x(k)$ and $T_y(k)$:

$$x(k+1) = x(k) + b_x T_x(k) + T u(k) \tag{23}$$
$$y(k+1) = y(k) + b_y T_y(k) + T v(k) \tag{24}$$
$$u(k+1) = u(k) + T v_{M1}(k) \tag{25}$$
$$v(k+1) = v(k) + T v_{M2}(k) \tag{26}$$

The coefficients b_x and b_y depend on the values of the depth Z_s and the sampling interval T. If we transform the stochastic state-space model in observer form, we can derive two time-varying SISO ARMAX models (the ARMAX model in this case can be viewed as a compact way of writing down the innovations model):

$$A_i(q^{-1})\,y_i(k) = q^{-d} B_i(q^{-1})\,u_{ci}(k) + C_i(q^{-1})\,w_i(k) \quad k \geq 0,\ i = 1,2 \tag{27}$$

where

$$A_i(q^{-1}) = 1 + a_{i1} q^{-1} + a_{i2} q^{-2} \quad i = 1,2$$
$$B_i(q^{-1}) = b_{i0} + b_{i1} q^{-1} \quad i = 1.2$$
$$C_i(q^{-1}) = 1 + c_{i1} q^{-1} + c_{i2} q^{-2} \quad i = 1.2$$

The noise sequences $w_i(k)$ are assumed to satisfy the assumptions:

$$E\{w_i(k) \mid F^i_{k-1}\} = 0 \quad E\{w_i^2(k) \mid F^i_{k-1}\} = \sigma_i^2 \quad i = 1,2$$

where the symbol $E\{X\}$ denotes the expected value of tlic random variable X and F^i_{k-1} is the sigma algebra generated by the past measurements and the past control inputs up to time $k-1$. The index i corresponds to the two different SISO ARMAX models, the scalar input $u_{ci}(k)$ now represents either $T_x(k)$ or $T_y(k)$, and the scalar $y_i(k)$ corresponds to the measured deviation of the feature point from its desircd position in one of the X or Y directions. It can be shown that $b_{10} = -b_{11} = b_x$, $b_{20} = -b_{21} = b_y$ and $d = 1$. Since image processing calculations require finite time, the introduced computational delays can be represented by the delay factor d. For the time being, it is assumed that $d = 1$. The coefficients of the polynomials $C_i(q^{-1})$ depend on the values of the Kalman gains. and thus. $C_i(q^{-1})$ are time-varying due to the time-varying nature of the Kalman gains. It can be shown [1] that the polynomials $C_i(q^{-1})$ have their roots strictly inside the unit circle. The optimal one-stepahead predictor [14] with $d = 1$ has the fonn:

$$C_i(q^{-1})\,y_i^*(k+1|k) = A_i^*(q^{-1})\,y_i(k) + B_i^*(q^{-1})\,u_{ci}(k) \quad k \geq 0\ i = 1,2 \tag{28}$$

where

$$B_i^*(q^{-1}) = B_i(q^{-1}) \quad i = 1.2$$
$$A_i^*(q^{-1}) = q[C_i(q^{-1}) - A_i(q^{-1})] \quad i = 1.2$$
$$y_i^*(k+1|k) = E\{y_i(k+1) \mid F_k\} = y_i(k+1) - w_i(k+1)\ i = 1.2$$

$y_i^*(k|(k-1))$ denotes the optimal prediction of the output while the $\hat{y}_i(k)$ denotes the the output of the adaptive predictor. Since ihe $y_i^*((k-1)|(k-2))$, $y_i^*((k-2)|(k-3))$,... are not directly available, the previously estimated values $\hat{y}_i(k-1)$, $\hat{y}_i(k-2)$... can be used instead. The introduced error can be neglected because the polynomials $C_i(q^{-1})$ have all their zeros inside the unit circle. This fact guarantees that the $\hat{y}_i(k+1)$ converges to $y_i^*((k+1)|k)$ exponentially. Thus. tlic $\hat{y}_i(k)$ arc given by the formula:

$$\hat{y}_i(k) = \mathbf{f}_i^T(k-1)\,\hat{\mathbf{q}}_i(k-1) \quad i = 1,2 \tag{29}$$

where

$$\mathbf{f}_i^T(k-1) = [y_i(k-1), y_i(k-2), u_{ci}(k-1), u_{ci}(k-2), -\hat{y}_i(k-1), -\hat{y}_i(k-2)] \quad i = 1,2$$
$$\hat{\mathbf{q}}_i^T(k-1) = [\hat{\alpha}_{i1}(k-1), \hat{\alpha}_{i2}(k-1), \hat{b}_{i0}(k-1), \hat{b}_{i1}(k-1), \hat{c}_{i1}(k-1), \hat{c}_{i2}(k-1)] \quad i = 1,2$$

Based on the relations between the coefficients of the $B_i^{*}(q^{-1})$, we can reduce the number of the coefficients that should be estimated. The $\hat{y}_i(k)$ are now given by equation (29) where

$$\mathbf{f}_i^T(k-1) = [y_i(k-1), y_i(k-2), \Delta u_{ci}(k-1), -\hat{y}_i(k-1), -\hat{y}_i(k-2)] \quad i = 1,2$$

$$\hat{\mathbf{q}}_i^T(k-1) = [\hat{\alpha}_{i1}(k-1), \hat{\alpha}_{i2}(k-1), \hat{b}_{i0}(k-1), \hat{c}_{i1}(k-1), \hat{c}_{i2}(k-1)] \quad i = 1,2$$

The $\Delta u_{ci}(k)$ can be computed from the equation $\Delta u_{ci}(k) = u_{ci}(k) - u_{ci}(k-1)$. The next step is the on-line estimation of these parameters.

4.2. Estimation Schemes

Many estimation schemes have been proposed for this type of problem by different researchers [14]. Due to the time-varying nature of the estimated coefficients, a variation of the *extended least squares* (ELS) algorithm is used. It is called *least-squares with exponential data weighting* [14]. This algorithm discards old data exponentially under the assumption that the most recent data contain more information. The parameter vector $\mathbf{q}_i(k)$ is estimated on-line by the equations:

$$\hat{\mathbf{q}}_i(k) = \hat{\mathbf{q}}_i(k-1) + \mathbf{P}_i(k-1)\,\mathbf{f}_i(k-1)\,e_i(k) \quad i = 1,2 \tag{30}$$

$$\mathbf{P}_i(k-1) = \frac{1}{\lambda_i(k-1)}\left[\mathbf{P}_i(k-2) - \frac{\mathbf{P}_i(k-2)\,\mathbf{f}_i(k-1)\,\mathbf{f}_i^T(k-1)\,\mathbf{P}_i(k-2)}{\lambda_i(k-1) + \mathbf{f}_i^T(k-1)\,\mathbf{P}_i(k-2)\,\mathbf{f}_i(k-1)}\right] \quad i = 1,2 \tag{31}$$

$$e_i(k) = y_i(k) - \hat{y}_i(k) \quad i = 1,2 \tag{32}$$

The scalar $\lambda_i(X) = \bar{\lambda}_i$ is the design variable which should be selected to be $0 < \bar{\lambda}_i \leq 1$. In the examples, $\bar{\lambda}_i$ has been chosen to be 0.95. The next step in the implementation of our algorithm is the selection of the control law.

4.3. Selection of the Control Law

The first adaptive controller that we propose is a modified version of the *adaptive minimum variance* controller. The adaptive minimum variance controller is often called *self-runing* regulator [15] (STR). For this type of controller, the input is chosen so as to minimize the mean-square error between the output and the desired value. The corresponding cost function is:

$$J_i(k+d) = E[[y_i(k+d) - y_i^*(k+d)]^2 \mid F_k] \tag{33}$$

For $d = 1$, the input signal $\Delta u_{ci}(k)$ is obtained by the equation:

$$y_i^*(k+1) = \mathbf{f}_i^T(k)\,\hat{\mathbf{q}}_i(k) \quad i = 1,2 \tag{34}$$

The above equation is generated by replacing $\hat{y}_i(k)$ with $y_i^*(k)$ and k with $k+1$ in (29). One important observation is that in our application $y_i^*(k) = 0$ for every instant of time. It can be easily shown that if $y_i^*(k) = 0$ for every instant of time, then $\hat{y}_i(k) = 0$ for every instant of time. Thus, the coefficients of the $C_i(q^{-1})$ can be removed from the problem. This fact simplifies the algorithm and reduces the number of parameters that should be estimated. Thus, in the 2-D visual tracking problem only six parameters must be computed each instant of time. The general ARMAX formulation for the same problem requires the on-line estimation of ten parameters. The minimum variance controller presents some serious problems. The first and most important is that the coefficients b_{i0} can become zero or very small. This will result in large input signals or even unbounded ones. A second important problem is the possible saturation of the input signals. A large change in the values of the system parameters can create large input signals. A practical but partial solution to the problem is to bound the input signals. Another problem of the minimum-variance controller is its bad performance under noisy measurements. Large oscillations appear due to the effort of the controller to compensate fast for large errors. Thus, the application of minimum variance control law for robotic visual tracking is not highly recommended. Since the robotic system cannot track objects whose projections in the image plane move extremely fast, a smooth control performance is important in the case of robotic visual tracking. A smooth control performance can be achieved by using a cost function that includes the input signal (STRWU). The oscillations are reduced in number and in magnitude and the overall performance is better. The major disadvantage of the STRWU control scheme is the large steady-state error. The cost function that includes the input signal is:

$$J_i(k+d) = E[[y_i(k+d) - y_i^*(k+d)]^2 + \rho_i\,u_{ci}^2(k) \mid F_k] \tag{35}$$

For $d = 1$, the modified one-step-ahead predictor [14] is given by:

$$C_i(q^{-1})\,\frac{b_{i0}}{b_{i0}^2 + \rho_i}\left[y_i^{*}(k+1 \mid k) - y_i^*(k+1) + \frac{\rho_i}{b_{i0}}\,u_{ci}(k)\right] = \mathbf{f}_i'^T(k)\,\mathbf{q}_i'(k) + u_{ci}(k) \quad i = 1,2 \tag{36}$$

where $\mathbf{q}_i'(k)$ is the vector that contains the coefficients of the polynomials:

$$\frac{b_{i0}}{b_{i0}^2 + \rho_i}\left\{A_i^{*}(q^{-1}), \frac{\rho_i}{b_{i0}}[C_i(q^{-1}) - 1]\,q + \{B_i^{*}(q^{-1}) - b_{i0}]\,q, -C_i(q^{-1})\right\} \quad i = 1,2$$

and $\mathbf{f}_i'(k)$ is given by the following equation:

$$\mathbf{f}_i'^T(k) = [y_i(k), y_i(k-1), u_{ci}(k-1), u_{ci}(k-2), y_i^*(k+1), y_i^*(k), y_i^*(k-1)] \quad i = 1,2$$

Goodwin [14] has suggested an adaptive control scheme for this type of problem. The proposed algorithm uses the following control law:

$$u_{ci}(k) = -\mathbf{f}_i'^T(k)\,\hat{\mathbf{q}}_i'(k) \quad i = 1,2 \tag{37}$$

where

$$\mathbf{q}_i'(k) = \mathbf{q}_i'(k-1) + \frac{1}{b_{i0}^{*} + \frac{\rho_i}{b_{i0}^{*}}}\,\mathbf{P}_i(k-1)\,\mathbf{f}_i'(k-1)\,e_i(k) \tag{38}$$

$$\mathbf{P}_i(k-1) = \mathbf{P}_i(k-2) - \frac{\mathbf{P}_i(k-2)\,\mathbf{f}_i'(k-1)\,\mathbf{f}_i'^T(k-1)\,\mathbf{P}_i(k-2)}{1 + \mathbf{f}_i'^T(k-1)\,\mathbf{P}_i(k-2)\,\mathbf{f}_i'(k-1)} \tag{39}$$

$$e_i(k) = \frac{\rho_i}{b_{i0}^{*}}\,u_{ci}(k) + y_i(k) - y_i^*(k) \tag{40}$$

The symbol b_{i0}^{*} denotes the initial estimate of the coefficient b_{i0}. The above algorithm minimizes the cost function (35) with a modified p, that we call ρ_i':

$$\rho_i' = \frac{\rho_i\,b_{i0}}{b_{i0}^{*}}$$

Due to the fact that $y_i^*(k) = 0\ \forall\ k > 0$, the three last coefficients of the $\mathbf{q}_i'(k)$ need not be computed. Thus, eight parameters should be estimated in total. The computational complexity is higher than the STR control scheme but the performance is improved. The problem with this approach is that it creates a steady-state error (SSE). To reduce the SSE, one should introduce an integrator in the system. The last controller (STRWDU) which is implemented is designed to provide integral action. This can be accomplished by weighting the control signal change. This is in agreement with the structural and operational characteristics of a robotic system. A robotic system cannot track objects that have large changes in their image projections during the sampling interval T. In addition, there are some upper limits in the robotic tracking ability. The cost function that includes the control signal change is:

$$J_i(k+d) = E\{[y_i(k+d) - y_i^*(k+d)]^2 + \rho_i\,\Delta u_{ci}^2(k) \mid F_k\} \tag{41}$$

Based on the relations between the coefficients of $B_i^{*}(q^{-1})$, we can rewrite the modified one-step-ahead predictor in (36) as:

$$C_i(q^{-1})\frac{b_{i0}}{b_{i0}^2+\rho_i}\left[y_i^{m}(k+1|k)-y_i^{*}(k+1)+\frac{\rho_i}{b_{i0}}\Delta u_{ci}(k)\right]=$$
$$=\mathbf{f}_i'^{T}(k)\,\mathbf{q}_i'(k)+A\,u_{ci}(k)\quad i=1,2 \tag{42}$$

where $\mathbf{q}_i'(k)$ is the vector that contains the coefficients of the polynomials.

$$\frac{b_{i0}}{b_{i0}^2+\rho_i}\left\{A_i^{m}(q^{-1}),\frac{\rho_i}{b_{i0}}[C_i(q^{-1})-1]\,q,-C_i(q^{-1})\right\}\quad i=1,2$$

and the new $\mathbf{f}_i'$ (k) is given by:

$$\mathbf{f}_i'^{T}(k)=[y_i(k),y_i(k-1),\Delta u_{ci}(k-1),\Delta u_{ci}(k-2),$$
$$y_i^{*}(k+1),y_i^{*}(k),y_i^{*}(k-1)]\quad i=1,2$$

The new control law is:

$$A\,u_{ci}(k)=-\mathbf{f}_i'^{T}(k)\,\hat{\mathbf{q}}_i'(k)\quad i=1,2 \tag{43}$$

The estimation scheme is almost the same as the one in equations (38)-(40). The only change is in equation (40) where the $u_{ci}(k)$ should be changed to $A\,u_{ci}(k)$. In total, eight parameters should be estimated on-line. This controller seems the most appropriate for the specific control problem that we have to solve.

5. Robot Control

After the computation of $u_{ci}(k)$ signals with respect to the camera frame R_s, we transform them to the end-effector frame R_e with the use of the transformation ^eT_s. The transformed signals are fed to the robot controller. We experimented with a cartesian PD scheme with gravity compensation. The selection of the appropriate robot control method is essential to the success of our algorithms because small oscillations can create blurring in the acquired images. Blurring reduces the accuracy of the visual measurements. and as a result the system cannot accurately track the moving object. The mathematical model of the robot's dynamics is:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}+\mathbf{c}(\mathbf{q},\dot{\mathbf{q}})+\mathbf{g}(\mathbf{q})=\boldsymbol{\tau} \tag{44}$$

where q is the vector of the joint variables of the robotic arm, D is the inertial acceleration related matrix, c is the nonlinear Coriolis and centrifugal torque vector. g is the gravitational torque vector and τ is the generalized torque vector. The model is nonlinear and coupled. This control scheme assumes that all velocities in the dynamics equations are zero. This implies that $\dot{\mathbf{q}}=\mathbf{J}=\mathbf{c}(\mathbf{q},\dot{\mathbf{q}})=0$. $\mathbf{J}(\mathbf{q})$ is the manipulator's Jacobian. Thus, the actuator torque vector τ is given by:

$$\boldsymbol{\tau}=\mathbf{J}^T(\mathbf{q})\,\mathbf{F}+\mathbf{g}(\mathbf{q}) \tag{45}$$
$$\mathbf{F}=\mathbf{K}_p\,\Delta\mathbf{x}+\mathbf{K}_v\,\Delta\dot{\mathbf{x}}+\ddot{\mathbf{x}}_{des} \tag{46}$$
$$\Delta\dot{\mathbf{x}}=\dot{\mathbf{x}}_{des}-\mathbf{J}(\mathbf{q})\,\dot{\mathbf{q}} \tag{47}$$

where $\mathbf{F}$ is the generalized force vector, $\Delta\mathbf{x}^T=(\Delta x:,\ \Delta\mathbf{x}_o^T)$ is the position and orientation error vector. and $\mathbf{K}_p$ and $\mathbf{K}_v$ are gain matrices. The subscript *des* denotes the desired quantities. The next Section describes the experimental results.

6. Experiments

A number of experiments were performed on the CMU DDArm II robotic system. A description of the hardware configuration of the CMU DDArm II can he found in [13]. The camera is mounted on the end-effector. The focal length of the camera is 7.5mm while the objects are moving on a plane (average depth Z_s=680mm). The center of mass of eaich one of these objects moves across the line $Y=0.74X+0.77$ (Y and X in meters). The real images are 510x492 and are quantized to 256 gray levels. The objects used in the tracking examples are books, pencils. and generally. items with distinct features. The user, by using the mouse. proposes to the system some of the object's features that he is interested in. Then. the system evaluates on-line the quality of the measurements. based on the confidence measures described in a previous Section. Currently, four features are used and the size of the attached windows is 10x10. The experimental results are plotted in Fig. 1-3 where the dotdashed trajectories correspond to the trajectories of the center of mass of the moving objects. The *Mes_P* vector represents the position of the end-effector with respect to the world frame. For the STR controller, the covariance matrix $P_i(k)$ is initialized to be $P_{i0}=I$, and the initial value of the vector $\hat{\mathbf{q}}_i^T(0)$ is $\hat{\mathbf{q}}_i^T(0)=[0.0,0.0,-1.0]$. For both the STRWU and STRWDU controllers. the covariance matrix P_i(k) is initialized to be $P_{i0}=I$, and the initial value of the vector $\hat{\mathbf{q}}_i'^T(k)$ is $\hat{\mathbf{q}}_i'^T(0)=[-1.0,1.0,1.0,1.0]$. The value of the scalar ρ_i is 0.2. The experimental results lead to some interesting observations. The simple PD produces oscillations around the desired trajectory. The reason why we do not use the computed torque scheme is that it requires the inversion of the Jacobian. Thus, the DDArm II can easily become unstable (whenever two of the joints are aligned). The observed oscillations are due to the fact that the robotic controller (PD with gravity compensation) does not take into consideration the robot dynamics. The results are presented in Fig. 1-3. The knowledge of the depth Z_s is assumed to be inaccurate. The adaptive minimum variance controller (STR) is implemented with bounded control signal change $\Delta u_{ci}(k)$. This controller. as depicted in Fig. 1, has the worse performance. The reason is that the large variations in the control signal create blurring in the images, and thus. large errors in the visual measurements. The STRWU and STRWDU controllers have almost comparable performances. The STRWU controller presents n steady-state error while the STRWDU regulator seems to hnve the smoothest performance.

7. Conclusions

In this paper, the robotic visual tracking (hand-eye configuration) problem is addressed. We claim that we should look to the problem by combining vision and control techniques together. The potential of the proposed approach has been demonstrated by presenting experimental results from the application of our framework to the problem of robotic visual tracking of arbitrary 3-D objects traveling at unknown velocities in a 2-D space. We first presented a mathematical formulation of the problem. This model is a major contribution of this work and is bnscd on measurements of the vector of discrete displacements which are obtained by the sum-of-squared differences (SSD) optical flow This model can be extended to 3-D by including in the calculations a larger number of feature points. Other contributions of this work are n sophisticated measurement scheme using multiple windows and efficient confidence measures that improve the accuracy of the visual measurements. The next step was to show the effectiveness of the idea of combination of control with vision as a solution to the robotic visual tracking problem. Adaptive control schemes were introduced for the case of inaccurate knowledge of some of the system's parameters. Three different adaptive controllers were implemented on our experimental testbed, the DDArm II system. Experimental results show that the methods are quite accurate, robust and promising. One important observation is that all the experiments were done in real-time.

8. Acknowledgements

This research was supported by the Defense Advnnccd Research Projects Agency, through ARPA Order Number DAAA-21-89C-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either cxprcssed or implied. of the funding agencies. We should also thank Innovision Inc. for providing us with the image processing equipment.

References

1. N. Papanikolopoulos, P. Khosla, and T. Kanade, "Robotic visual tracking: Theory and experiments", Tech. report , Carnegie Mellon University, The Robotics Institute, 1990.

2. D. Tsakiris, "Visual tracking strategies", Master's thesis, Department of Electrical Engineering, University of Maryland, 1988.

3 R.C. Luo. R.E. Mullen Jr.. and D.E. Wessel, "An adaptive robotic tracking system using optical flow", *Pror. of the IEEE Int. Conf. on* Roborrcs and *Automation*, 1988. pp. 568-573.

4. R. Goldenberg. W.C. Lau. A. She, and A.M. Waxman, "Progress on the prototype PIPE". Proc. of the IEEE *Int. Conf. on Roborics* and *Automation*, 1987. pp. 1267-1274.

5. P.K. Allen, "Real-time motion racking using spatio-temporal filters", Proc. DARPA *Image Understanding* Workshop. 1989, pp. 695-701.

6. A.E. Hunt and A.C. Sanderson. "Vision-based predictive tracking of a moving target". Tech. report CMU-RI-TR-82-15, Carnegie Mellon University. The Robotics Institute. January 1982.

7 J.T.Feddema, C.S.G. Lee, and O.R. Mitchell. "Automatic selection of image features for visual servoing of a robot manipulator". Proc. *of the IEEE Inr. Conf. on Roborics and Automation*, May 1989. pp. 832-837.

8. L.E. Weiss, A.C. Sanderson. and C.P. Neuman, "Dynamic sensor-hascd control of robots with visual feedback". IEEE *Journal of Robotics and Automation*, Vol. RA-3. No. 5, October 1987, pp. 404-417

9. B.K.P. Horn, *Robot* vision. MIT Press. Cambridge, 1986.

10. B.K.P Horn and B.G. Schunck, "Determining optical flow", *Artificial Intelligence*. Vol. 17, 1981, pp. 185-204.

11. D.J Heeger. "Depth and flow from motion energy", Science. Vol. , No. 86, 1986, pp. 657-663.

12. P. Anandan, "Measuring visual motion from image sequences", Tech. report COINS-TR-87-21, COINS Department, University of Massachusetts, 1987.

13. N. Papanikolopoulos. P. Khosla, and T. Kanade. Vision and *control* techniques *for robotic visual tracking*, Accepted to the 1991 IEEE Int. Conf. on Robotics and Automation., 1991.

14. G.C. Goodwin and K.S. Sin, Adaptive filtering. prediction and control. Prentice-Hall, Inc., Englewood Cliffs. New Jersey 07632, Information and Systems Science Series. Vol. I. 1984.

15. K.J. Astrom, U. Borisson. L. Ljung and B. Wittenmark, "Theory and application of self-tuning regulators". *Automatica*, Vol. 13, No. 5, 1977. pp. 457-476.

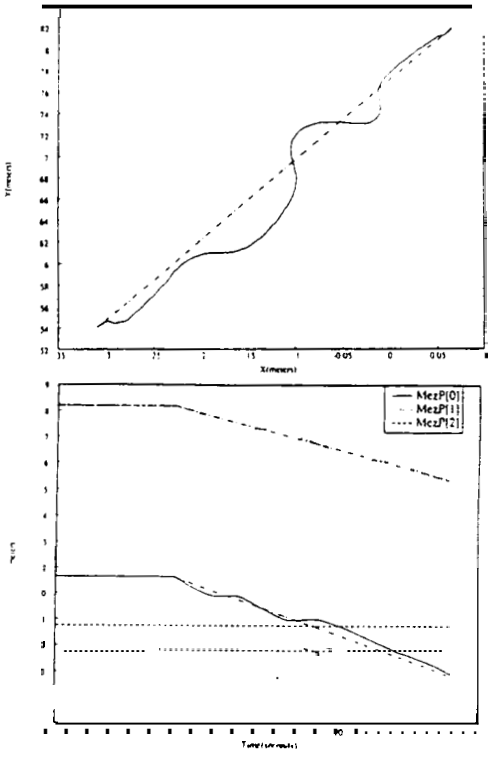

Figure 1: STR adaptive controller in conjunction with a cartesian PD robotic conirollcr with gravity compensation (Experimental).

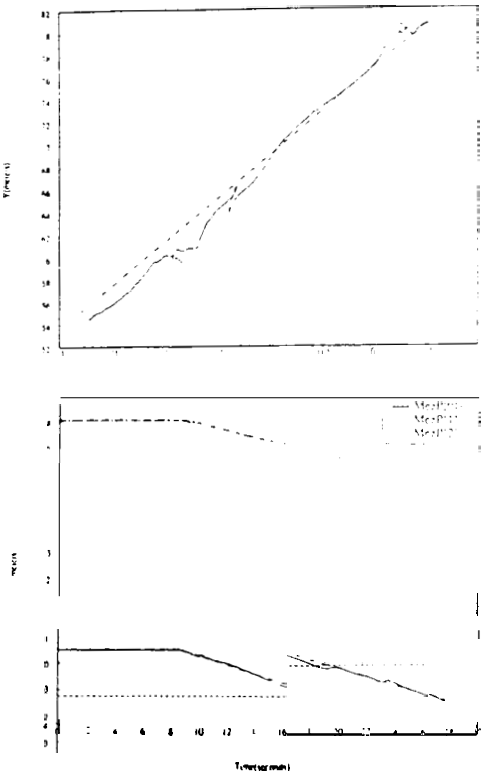

Figure 2: STRWU adaptive controller in conjunction with a cartesian PD robotic controller with gravity compensation (Experimental).

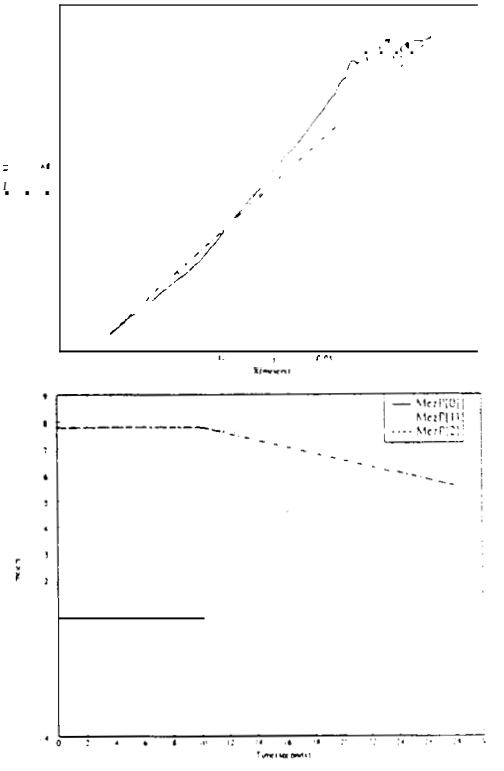

Figure 3: STRWDU adaptive controller in conjunction with a cartesian PD robotic controller with gravity compensation (Experimental).