1992

Shared and Traded Telerobotic Visual Control

N. P. Papanikolopoulos

Pradeep Khosla
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/isr
SHARED AND TRADED TelerOBOTIC VISUAL CONTROL

N. P. Papanikolopoulos and P. Khosla
Department of Electrical and Computer Engineering
The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract
This paper addresses the problem of integrating the human operator with autonomous robotic visual tracking and servoing modules. A CCD camera is mounted on the end-effector of a robot and the task is to serve around a static or moving rigid target. In manual control mode, the human operator, with the help of a joystick and a monitor, commands robot motions in order to compensate for tracking errors. In shared control mode, the human operator and the autonomous visual tracking modules command motions along orthogonal sets of degrees of freedom. In autonomous control mode, the autonomous visual tracking modules are in full control of the servoing functions. Finally, in traded control mode, the control can be transferred from the autonomous visual modules to the human operator and vice versa. This paper presents an experimental setup where all three different schemes have been tested. Experimental results of all modes of operation are presented and the related issues are discussed. In certain degrees of freedom the autonomous modules perform better than the human operator. On the other hand, the human operator can compensate fast for failures in tracking while the autonomous modules fail. Their failure is due to difficulties in encoding an efficient contingency plan.

1. Introduction
In unstructured and/or hazardous environments such as space underwater, and radioactive sites, autonomous robots are needed to reliably cope with complex and diverse tasks. Autonomous capabilities are developed through the use of sensors such as force, vision, tactile, etc., that provide information about the state of the task, the robot, and the environment and through sophisticated algorithms that acquire, interpret, and use sensory information to drive the robot system. Recent advances in sensors, algorithms, and computer technology have enabled the integration of sensors in the feedback loop for real-time task execution. In spite of these advances, present day robotic devices do not possess desirable characteristics such as flexibility, reliability, adaptability, to name a few, and pure autonomy remains an unsolved goal.

The goal of using autonomous robots for applications in unstructured and/or hazardous environments is to remove the human from the task site but not necessarily to remove the human/operator from the task itself. In fact due to safety considerations, it is very important that the human be able to remotely take control of the task and the robot system. For example, unpredictable events and complex evolving environments require the intervention of the human operator. The integration of the human operator in a robotic system results in a system that combines man and machine capabilities and such a system is often called a telerobotic system. During the last decade, research efforts [1, 2, 3, 4] have focused on building telerobotic systems which provide a wide range of control modes, from pure autonomous control to full teleoperation. Most of these efforts integrate the human operator with force/position control schemes. Other sensors, such as CCD cameras, are simply used to provide visual information to the human operator through a video display. In contrast, we view a telerobotic system as a general multi-sensor environment where the human operator and the autonomous sensory modules can cooperate and exchange control of the decision-making process. The exchange of control depends on the specific events which occur during the execution of a task.

In this paper, we address the problem of the integration of a human operator with autonomous visual servoing and tracking modules. In particular, we examine ways that the human operator can cooperate with these modules and improve their performance. Our previous work in visual servoing and tracking emphasized the design of vision and control algorithms for tracking of targets in unknown environments [5, 6, 7]. The camera was mounted on the robot manipulator (eye-in-hand configuration) and the objective was to keep the projection of the target at some desired position on the image plane. Our work differs significantly from previous work in visual servoing and recursive depth estimation [8, 9] and recursive depth estimation [10, 11]. The differences are in the use of a single moving camera (binocular viewing requires the mounting and the calibration of two cameras which is difficult), the ability to compensate for uncertainties in the relative distance of the moving target from the camera frame, and for unknown camera parameters, and the generality of our framework (we have demonstrated full 3-D robotic visual tracking [12]).

We present examples of visual servoing around a static or moving target. The autonomous modules are based on adaptive control techniques especially appropriate for operation in an unknown environment. We assume that we do not know the exact depth map of the targets and that we have a limited knowledge of the camera model and the image noise. In certain cases, the autonomous modules perform better than the human operator, especially when an inexperienced user operates the joystick and interprets the visual data. Another important observation is the way the delays in the human reaction influence the stability of our algorithms. Large delays make the tracking unsuccessful while small delays can be included in the mathematical model, and these, an appropriate control scheme can be designed.

The paper is organized as follows: Section 2 presents the mathematical modeling of our problem. The different strategies for the solution of the robotic visual tracking and servoing problem are discussed in Section 3. In Section 4, we present ways for integration of the human operator with the autonomous modules. The architecture of our experimental testbed, CMU DD Arm II, is given in Section 5. The experimental results are presented in Section 6. Finally, in Section 7, the paper is summarized.

2. System Modeling
This Section describes the mathematical modeling of our problem. Due to the fact that the majority of the tracking and servoing examples can be decomposed into the simultaneous tracking of several features, only the tracking and servoing model for one feature is presented in this Section.

We assume a pinhole camera model with a frame (R) attached to θ. In addition, we also assume a perspective projection. Consider a static
Equations (8) and (9) will henceforth be used with \( k \) instead of \( t \). Equations (8) and (9) do not include any computational delays that are associated with the computation and the realization of the servomotion of the camera. If we include these delays in the model, equations (8) and (9) are transformed to:

\[
\begin{align*}
\mu_x(k) &= \mu_x(k) + x_c(k) \cdot (k - d + 1) + \mu_x(k) + q \cdot d^{-1} \cdot \mu_x(k) \\
\mu_y(k) &= \mu_y(k) + y_c(k) \cdot (k - d + 1) + \mu_y(k) + q \cdot d^{-1} \cdot \mu_y(k)
\end{align*}
\]
3.3-D Visual Servoing Around a Static/Moving Target

This section examines the control strategies that realize the servoing and tracking motion. The estimation scheme used to estimate the unknown parameters of the model and the ways we can integrate the human operator in the feedback loop. First, we present the control structure for servoing around a static target, and then we present the structure for tracking a target that moves with 3-D translation motion. It is assumed that the depth \( z(k) \) and the depth-related parameters (in equations (19) and (20)) are unknown. In order words, we initialize our system with some estimate of the depth and the depth-related parameters. The values of these parameters can be different from the actual values by a factor of 2 to 3. This fact allows for the use of our algorithms in poorly calibrated environments like underwater space and nuclear sites. The depth and the depth-related parameters are estimated on-line by using the motion of the camera-robot system and the possible motion of the target.

3.1 Visual Servoing Around a Static Target

The first example is visual servoing around a static target. First, we present the autonomous visual servoing modules that perform this task, and then, in Section 6, we compare the results with the results that a human operator can achieve. The control objective is to move the features' projections on the image plane to some selected positions. The repositioning of the projections is realized by an appropriate motion of the system robot-camera. In [9], it is proved that at least three feature points are needed, especially when a certain pose is not required. If a certain pose is required, more than three feature points are needed. We assume that the target is stationary and therefore \( u(k) = 0 \). The measurement vector \( y(k) \) is composed of the positions of four feature points at every instant \( k \). A simple control law can be derived by the minimization of a cost function \( J \) which includes the control signal:

\[
J(k+1) = E[\|y(k+1) - y^*(k+1)\|^2] + u^T(k) P_{u}(k) u(k)
\]

where the expectation \( E(\cdot) \) denotes the expected value of the random variable \( X \) and \( P_{u}(k) \) denotes data (past measurements and control inputs) up to time \( k \). The vector \( y^*(k) \) represents the desired positions of the projections of the features' image points in the plane image. In our experiments, the vector \( y^*(k) \) is known \textit{a priori} and is constant over time. The control law is:

\[
u(k) = -\hat{B}_u(k) Q \hat{B}_u(k) + L^{(1)} \hat{B}_u(k) L^{(2)} Q (y(k) - y^*(k+1))
\]

where \( \hat{B}_u(k) \) is the estimated value of the matrix \( B_u(k) \). The design parameters in this control law are the elements of the matrices \( Q \). The matrix \( \hat{B}_u(k) \) is dependent on the estimated values of the features' depth \( z_u(k) \) and the coordinates of the features' image projections. In particular, the matrix \( \hat{B}_u(k) \) is defined as follows:

\[
\hat{B}_u(k) = \begin{bmatrix}
\hat{b}_u^{(1)}(k) \\
\hat{b}_u^{(2)}(k) \\
\hat{b}_u^{(3)}(k)
\end{bmatrix}
\]

where \( \hat{B}_u^{(i)}(k) \) is given by:

\[
\begin{bmatrix}
\hat{b}_u^{(1)}(k) \\
\hat{b}_u^{(2)}(k) \\
\hat{b}_u^{(3)}(k)
\end{bmatrix} = \begin{bmatrix}
\frac{1}{z_u(k)} x_u(k) y_u(k) \\
\frac{1}{z_u(k)} e^{(1)}(k) \\
\frac{1}{z_u(k)} e^{(2)}(k)
\end{bmatrix}
\]

The estimation of the features' depth \( z_u(k) \) with respect to the camera frame can be done in multiple ways. We define the inverse of the depth \( z_u(k) \) as \( \xi_u^{(i)}(k) \). Then, equations (19)-(21) of each feature point can be rewritten as:

\[
y_P(k) = A_p^{(i)} (k) y_P(k-1) + A_p^{(i)} (k-1) B_p^{(i)} (k-1) T(k-1) + B_p^{(i)} (k-1) R(k-1) + n_p^{(i)} (k)
\]

when \( B_p^{(i)} (k) \) and \( B_p^{(i)} (k) \) are given by:

\[
B_p^{(i)} (k) = T\begin{bmatrix}
-1 & 0 & x_i(k) \\
0 & -1 & y_i(k)
\end{bmatrix}
\]

and the vector \( n_p^{(i)} (k) \) is a mean noise vector with mean 0 and covariance \( \Sigma_p^{(i)} (k) \). By defining \( u_p^{(i)} (k) \) and \( u_p^{(i)} (k) \) as \( u_p^{(i)} (k) = B_p^{(i)} (k) T(k) \) and \( u_p^{(i)} (k) = B_p^{(i)} (k) R(k) \), respectively, equation (24) is transformed into:

\[
y_{p}^{(i)} (k) = \xi_u^{(i)} (k-1) y_P^{(i)} (k-1) + \xi_u^{(i)} (k-1) u_p^{(i)} (k-1) + u_p^{(i)} (k-1) + n_p^{(i)} (k)
\]

A last transformation of equation (25) is done by using the vector \( A \), which is defined as:

\[
A \gamma_p^{(i)} (k) = y_{p}^{(i)} (k) - y_{p}^{(i)} (k-1) - u_p^{(i)} (k-1)
\]

The new form of equation (25) is:

\[
\Delta \gamma_p^{(i)} (k) = \xi_u^{(i)} (k-1) u_p^{(i)} (k-1) + n_p^{(i)} (k)
\]

The vectors \( \Delta \gamma_p^{(i)} (k) \) and \( u_p^{(i)} (k-1) \) are known in every instant of time, while the scalar \( \xi_u^{(i)} (k) \) is the inverse of the unknown depth \( z_u^{(i)} (k) \). It is continuously estimated. It is assumed that an initial estimate \( \hat{\xi}_u^{(i)} (0) \) of \( \xi_u^{(i)} (0) \) is given and \( \xi_u^{(i)} (0) = E(\xi_u^{(i)} (0) - \hat{\xi}_u^{(i)} (0)) \) is a positive scalar \( \rho_\xi \) that can be interpreted as a measure of the confidence that we have in the initial estimate \( \hat{\xi}_u^{(i)} (0) \). Accurate knowledge of the scalar \( \xi_u^{(i)} (k) \) corresponds to a small covariance scalar \( \rho_\xi \). In our examples, \( \Sigma_u^{(i)} (k) \) is a constant pre-defined matrix. In addition, for simplicity in notation, \( h(k) \) is used instead of \( u_p^{(i)} (k) \).

The estimation equations are (the superscript (−) denotes the predicted value of a variable while the superscript (+) denotes its updated value):

\[
\begin{align*}
-\hat{\xi}_u^{(i)} (k) &= \hat{\xi}_u^{(i)} (k-1) + \\
\rho_\xi (k) &= \rho_\xi (k-1) + e^{(i)} (k-1)
\end{align*}
\]

\[
\begin{align*}
\rho_\xi (k) &= \rho_\xi (k-1) + e^{(i)} (k-1) + h(k-1) R(k-1)
\end{align*}
\]

\[
\begin{align*}
\hat{k}^{(i)} (k) &= \rho_\xi (k) h^{(i)} (k-1)
\end{align*}
\]

\[
\begin{align*}
\hat{\xi}_u^{(i)} (k) &= \hat{\xi}_u^{(i)} (k) + \hat{k}(k) \Delta \gamma_p^{(i)} (k) - \xi_u^{(i)} (k-1)
\end{align*}
\]

where \( \hat{\xi}_u^{(i)} (k) \) is a covariance scalar which corresponds to the noise that characterizes the transition between the states. The depth of the parameter \( \xi_u^{(i)} (k) \) is a time-varying variable since the camera translates along its optical axis and rotates along the X and Y axes. The estimation scheme of equations (27)-(31) can compensate for the time-varying nature of \( \xi_u^{(i)} (k) \) because it is designed under the assumption that the estimated variable undergoes a random change. One problem is to keep the covariance scalar \( \rho_\xi (k) \) finite. Solutions for this type of problem can be found in [13]. In addition, we implemented estimation techniques such as exponential data weighting and covariance resetting [11] which deal with time-varying parameters.
32. Visual Servoing Around a Target with 3-D Translational Motion

This Section presents our strategies for visual tracking of a moving target. Let us assume that the target moves with 3-D translational motion and the objective is to keep the target's projection on the image plane stationary while the target moves. Therefore, we must only compute \( T_x(k), T_y(k), \) and \( T_z(k) \). Due to the fact that the target is moving, we assume \( a_x = a_y = a_z = 1 \). To simplify the modeling of the noise, white noise terms are assumed to accompany the terms \( u_x(k) \) and \( v_x(k) \). Therefore, we transform equations (17) and (18) \( n_w(k), \)

\[ n_w(k) \]

be the noise terms. The next equations for a single feature point \( R(k) = R_x(k) = R_y(k) = 0 \):

\[ x(k+1) = x(k) + T_x(k) + T u_x(k) + (T^2/2)n_w(k) \]

(32)

\[ y(k+1) = y(k) + T y(k) + T v_y(k) + (T^2/2)n_w(k) \]

(33)

\[ u_x(k+1) = u_x(k) + T n_w(k) \]

(34)

\[ v_y(k+1) = v_y(k) + T n_w(k) \]

(35)

where \( S_x(k) \) and \( S_y(k) \) are defined as:

\[ S_x(k) = T_x(k) + y(k) T_y(k) \]

\[ S_y(k) = T_y(k) + x(k) T_x(k) \]

and coefficients \( b_1, b_2, \) and \( b_3, b_4, \) depend on the values of the depth \( Z \), and of the sampling interval \( T \). For \( M \) feature points, we have \( M \) sets of the equations (32)-(35).

The next step in our modeling is to remove the terms \( u_x(k) \) and \( v_y(k) \) from our formulation. In order to achieve this, we derive two time-varying SISO (Single-Input Single Output) ARMAX (AutoRegressive Moving-Average) model with \emph{external input} models (the ARMAX [13] model in this case can be viewed as a compact way of writing down the innovations model). The ARMAX models are derived by subtraction and addition of the expressions for \( x(k), x(k-1) \) and \( x(k-2), \) and \( y(k), y(k-1) \) and \( y(k-2), \) respectively. For each feature, the corresponding two time-varying SISO ARMAX model is:

\[ A_{2i} (q^{-1}) y_i(k) = B_1 (q^{-1}) u_i(k) + C_2 (q^{-1}) w_i(k) \]

(36)

where

\[ A_{2i} (q^{-1}) = 1 + a_{11} q^{-1} + a_{21} q^{-2} i = 1, 2 \]

\[ B_1 (q^{-1}) = b_{11} q^{-1} + b_{21} q^{-2} i = 1 \]

\[ C_2 (q^{-1}) = 1 + c_{11} q^{-1} + c_{21} q^{-2} i = 1 \]

The values of the coefficients \( a_{11}, a_{21}, b_{11}, b_{21}, c_{11}, c_{21} \) depend on the values of \( T, b_1, b_2, \) and \( b_3, b_4, \). The noise sequences \( w_i(k) \) are assumed to satisfy the assumptions

\[ E[w_i(k) F_i^{k-1}] = 0 \]

where the symbol \( E \) denotes the expected value of the random variable \( X \) and \( F_i^{k-1} \) denotes data (past control inputs and measurements) up to time \( k-1 \). The index corresponds to the two different SISO ARMAX models per feature point, the scalar input \( u_i(k) \) now represents either \( S_x(k) \) or \( S_y(k) \), and the scalar \( y_i(k) \) corresponds to the measured deviation of the feature point from its desired position in one of the \( X \) or \( Y \) directions. It can be shown that \( b_{10} = b_{11} = b_{21} = b_3 \) and \( b_{12} = b_{22} = b_4 \).

The control and estimation techniques which are used in order to compute an efficient adaptive control law can be found in [7].

4. Issues in Shared and Traded Control

In shared control mode, some degrees of freedom (DOF) are commanded by the human operator and some by the autonomous modules. In traded control mode, there is a transition from the autonomous modules to the human operator and vice versa. There are a lot of different ways of looking at the design of modules that support these modes. In [2], mixing matrices and weights are used to decide when and how the autonomous modules will mix with the human operator. On the other hand, in [17, 3], the human operator and the autonomous modules are not allowed to command the Same degree of freedom (DOF). The reason is to avoid the "fork in the road" problem which may occur when the human operator and the autonomous modules suggest opposite directions of motion along a specific degree of freedom (DOF).

In order to avoid the "fork in the road" problem, we choose to follow the latter approach. As was mentioned in Section 2, the control input signal \( u_i(k) \) is expressed with respect to the camera frame \( (R_i) \). Let us assume that the human operator commands a speed signal \( u_i(k) \) with respect to the joystick frame. The speed signal \( u_i(k) \) can be transformed through the transformation \( T_j \) to the camera frame signal \( u_i(k) \). We define the matrices \( \Psi \) and \( \Omega (\Psi, \Omega \in R^2) \):

\[ \Psi(\alpha, \beta) = \begin{cases} 1 & \text{if } \alpha = \beta \\ 0 & \text{otherwize} \end{cases} \]

(37)

\[ \Omega(\alpha, \beta) = \begin{cases} 1 - \Psi(\alpha, \beta) & \text{if } \alpha = \beta \\ 0 & \text{otherwize} \end{cases} \]

(38)

where \( \Psi(\alpha, \beta) = 1 \) (as \( \{1, \ldots, 6\} \)) implies that the human operator controls the different degrees of freedom (DOF). The new rate reference signal \( u_i(k) \) is expressed in camera frame coordinates as:

\[ u_i(k) = \Psi u_i(k) = c + \Psi u_i(k) \]

(39)

One of the possible problems of shared control mode is the possible coupling between the degrees of freedom (DOF) that the human operator commands and the degrees of freedom commanded by the autonomous modules. For example, in our system, there is a strong coupling between \( T_x(k) \) and \( T_y(k) \), and between \( T_x(k) \) and \( T_z(k) \). Therefore, a modification of one of them makes necessary a modification of the other's value. The implementation of this scheme depends on the knowledge of the human operator about the type of coupling. The selection of the degrees of freedom (DOF) which are manually commanded is not an easy task. The way that the CCD camera is mounted on the end-effector and the 2-D nature of the visual information force the human operator to select \( R_i(k) \), \( T_i \) and \( T_j \), as possible teleoperated degrees of freedom (DOF). This observation is verified by the experimental results which are presented in Section 6.

The experimental results show that the manual control of \( T_x(k) \) and \( T_y(k) \) gives poor tracking performance.

Another potential problem is the delay in the manual rate signal. If \( d_\alpha \) is a delay factor associated with the transmission of the manual rate signal then equation (39) becomes:

\[ u_i(k) = \Omega u_i(k) + c + \Psi u_i(k - d_\alpha) \]

(40)

Large transmission delays which are likely to be present for space teleoperation can significantly influence the tracking performance and the stability of our algorithms. The estimation schemes are influenced too, due to the fact that we apply different inputs to the system than the ones that the estimation schemes consider as current. Therefore, the transmission delays must be taken into consideration during the design phase of the whole system.

The traded control mode is useful during three phases: a) Initialization, b) Emergency events, and c) Final stage. During these phases, there is an exchange of control between the human operator and the autonomous modules. In particular, during Initialization, the
human operator grossly positions the manipulator with respect to the target and selects a number of candidate features for tracking. Then, he/she passes control to the autonomous modules. During emergency events (e.g., loss of one feature due to occlusion by an unknown object, singular configuration of the manipulator), the human operator takes control and tries to solve the accumulated problems. If the problems are solved, then he/she again passes control to the autonomous modules. Finally, during the Final stage, the human operator takes control of the system and places the manipulator in its home position. The next Section describes the architecture of our experimental setup.

5. System Architecture

The vision and joystick modules of the CMU DD Arm II (Direct Drive Arm II) system are parts of a bigger hardware environment which runs under the CHIMERA II [18] real-time operating system. The IDAS/150 image processing system carries out all the computational load of the image processing calculations while the Mercury Floating Point Unit does all the control calculations. An Iyonics board is responsible for the realization of the shared control planner. The IDAS/150 contains a Heurikon 68030 board as the controller of the vision module and two floating point boards, each one with computational power of 20 Mflops. The system can be expanded to contain as many as eight of these boards. The six degrees of freedom joystick is a Dimension 6 Trackball [19, 17]. The applied forces and torques to the trackball are transformed to six reference signals which are computed at once. The reference signals correspond to either velocities or forces. The human Operator can select the type and the reference frame of the reference signals on-line. In our experiments, the user commands motions with respect to the joystick frame, and the motions are transformed into camera frame coordinates. The camera frame is parallel to the end-effector frame.

The software is organized around 5 processes:

- **Vision process.** This process does all the image processing calculations and has a period of 150 ms.
- **Interpolation process.** This program reads the data from the vision system, interpolates the data and sends the reference signals to the robotic cartesian controller. It has a period of 5 ms.
- **Robot controller process.** This process drives the robot and has a period of 3.33 ms. The robot control scheme that is used is a cartesian PD controller with gravity compensation.
- **Joystick process.** This process reads the data from the joystick and does all the necessary data transformations. It has a period of 30 ms.
- **Joystick reference process.** This process interpolates the data from the joystick process in order to guarantee a smooth robot trajectory. Its period is 9 ms.

The next Section describes the experimental results.

6. Experimental Results

A number of experiments were performed on our experimental testbed, the CMU DD Arm II robotic system. A camera, whose parameters are given in Table 1, is mounted on the end-effector. The operator by using the joystick commands motions in the end-effector frame which is parallel to the camera frame. The objects are static or moving (the initial depth of the objects' center of mass with respect to the camera frame Z varies from 500 mm to 1000 mm). The maximum permissible translational velocity of the end-effector is 10 cm/sec and each one of the components (x, z, y) of the end-effector's rotational velocity does not exceed 0.05 rad/sec.

The objective of the first experiment was to move the manipulator so that the image projections of certain chosen features of the object move to some desired image positions. We tried to operate the system first in autonomous mode and then in manual mode. The results are plotted in Figures 3-5. 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 features, based on the confidence measures described in [5]. The same operation can be done automatically by a computer process that runs once and needs between 2 and 3 minutes, depending on the size of the image (or the features which are used). The three best features are selected and used for the robotic visual servoing task. The size of the windows is 10x10. In addition, the user selects the desired positions of these features. The current and the desired positions are continuously highlighted in order to help the human operator accomplish his/her task. The system provides messages about the current servoing errors and the timing of the task. These graphical axes are realized on a graphical overlay of the video display which has a frame rate of 30 frames/sec.

The gain matrices for the autonomous visual control modules are

\[ Q = \text{Id}_d \text{ and } \underline{L} = \text{diag}(0.025, 0.025, 0.2, 2 \times 10^5, 2 \times 10^5, 2 \times 10^5) \].

The computation of the \( H \) matrix is done on a Heurikon 68030 frame. To reduce the computational load of the matrix inversion, we use the techniques described in [9]. The knowledge of the depth \( Z \) is assumed to be inaccurate. As shown in Figures 3-5, the results from the autonomous modules are far better than the results that the human operator achieves. The results of the human operator show large delays and steady-state errors. Especially in Figure 5, one can observe the large steady-state error of the \( x \) component of the tracking error when the human operator is in charge (manual mode). One reason for the bad performance of the human operator is his/her lack of understanding of the depth. The images in the display give a good idea about motions in 2-D but not in 3-D. The large delays are associated with the delays that the human operator introduces in the control loop (delays of reaction). These delays are increased when the human operator is inexperienced which in certain cases results in an unsuccessful completion of the task.

The second experiment involved tracking of a moving (target) 3-D translational motion) under autonomous operation and also mixed operation. In these experiments, we used four features and selected the two best based on the confidence measures described in [5]. The experimental results of the 3-D case are plotted in Figures 6-13. The knowledge of the depth \( Z \) is assumed to be inaccurate. The property of our algorithms to compensate for uncertain depth and camera parameters is extremely valuable due to the fact that in certain robotic applications (space, underwater) we do not have to be able to accurately calibrate the operational space. The \( \text{Me}_P \) vector represents the position of the end-effector with respect to the world frame. In particular, in Figures 6-13 the three dotted trajectories correspond to the trajectories of the center of mass of the object in 3-D. In Figures 7, 9, 11, and 13 the vector \( \{X[0], Y[0], Z[0]\} \) denotes the initial position of the object or the manipulator's end-effector with respect to the world frame, while the vector \( \{X[1], Z[1]\} \) denotes the trajectories in 3-D of the object or the manipulator's end-effector. In Figures 6 and 7, the results from the application of the autonomous adaptive techniques are presented (autonomous mode). Their main characteristic is good tracking performance with small oscillations. The presence of oscillations is due to the fact that the computation of the translational velocity vector \( V \) is sensitive to image noise. Image noise affects the motion of the manipulator in the \( Z \) direction. In addition, a tracking error smaller than 2 mm in the \( Z \) direction cannot be detected as a pixel displacement in the image plane. This is due to the specific camera model and the specific positions of the features' projections on the image plane.

In Figures 8 and 9, we see the results of the combined effort of the human operator with the autonomous tracking modules (shared control). The human operator commands the robot motion across the optical axis of the camera \( \text{Me}_P[2] \) while \( \text{Me}_P[0] \) and \( \text{Me}_P[1] \) are
### 7. Conclusions

This paper has addressed issues related to the integration of a human operator with autonomous robotic visual servoing and tracking modules. We view the telerobotic control as the integration of the human operator with an environment that consists of multiple sensors and robots. Our work provides a framework for such an integration and an experimental setup has been built to test its potential. In particular, we state the problem of telerobotic visual servoing as a problem of combining the abilities and experience of a human operator with the efficiency and speed of simple autonomous visual tracking and servoing modules. We tested these ideas on our experimental test bed, the CMU D-3 Arm II. Experiments in visual servoing and visual tracking were performed while the human operator was included in the control loop through a joystick. The operator received visual information only through a video monitor and was not allowed to see the scene. The autonomous modules were based on the SSD algorithm for detection of motion and on adaptive control techniques for proper estimation of the unknown parameters of the environment (depth camera model, etc.).

The experiments lead us to some interesting observations. The human operator can compensate fast for errors in 2-D and can provide a contingency plan for tracking failures (bad feature selection, sudden target occlusion by another object, etc.). On the other hand, the human operator reacts with delay due to delays associated with the transfer of visual information, and has a poor perception of the depth parameter. In particular, it is extremely difficult for the operator to understand and react to motion parallel to the optical axis of the camera. Autonomous visual servoing modules allow for fast tracking in 3-D, even when the motion is parallel to the optical axis of the camera, adaptability to unknown parameters (unknown shape, inaccurate calibration), fast repositioning of the robot with respect to the target and high accuracy.

The integration of multiple cameras in the system (redundant visual information), the computational improvement of our algorithms, the introduction and use of "snakes" for autonomous contour servoing, the modeling of the delays of the human operator, the use of known CAD models for fast and accurate tracking, and the investigation of stability issues that arise with the presence of the human operator, are issues for future research.

### 8. Acknowledgements

This research was supported by the Defense Advanced 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 to represent the official policies, either expressed or implied, of the funding agencies. We must also thank Innovation Inc. for providing us with the image processing equipment.

### References


<table>
<thead>
<tr>
<th>f</th>
<th>s_x</th>
<th>s_y</th>
<th>c_x</th>
<th>c_y</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.5 mm</td>
<td>12.78 μm/pix</td>
<td>9.86 μm/pix</td>
<td>255 pixels</td>
<td>246 pixels</td>
</tr>
</tbody>
</table>

Table 1: Parameters of the camera model.


Figure 1: Initial image of the target in the servoing example around a static target.

Figure 2: Final image of the target in the servoing example around a static target.

Figure 3: Servoing errors for the first feature (A). The target is static.

Figure 4: Servoing errors for the second feature (B). The target is static.

Figure 5: Servoing errors for the third feature (C). The target is static.

Figure 6: SISO adaptive controllers (3-D case). All the degrees of freedom are controlled by the autonomous visual trackers (autonomous control). This figure shows the trajectories of the moving object and the manipulator's end-effector with respect to the world frame.

Figure 7: SISO adaptive controllers (previous example). All the degrees of freedom are controlled by the autonomous visual trackers (autonomous control). This figure shows the relative trajectories of the object and the manipulator's end-effector in Y-X and Z-X.
Figure 8: SISO adaptive controllers and the human operator (3-D case). \( M_{\text{SP}}[0] \) and \( M_{\text{SP}}[1] \) are commanded by autonomous visual servoing modules while \( M_{\text{SP}}[2] \) is commanded by manual control (shared control). This figure shows the trajectories of the moving object and the manipulator's end-effector with respect to the world frame.

Figure 9: SISO adaptive controllers and the human operator (previous example). \( M_{\text{SP}}[0] \) and \( M_{\text{SP}}[1] \) are commanded by autonomous visual servoing modules while \( M_{\text{SP}}[2] \) is commanded by manual control (shared control). This figure shows the relative trajectories of the object and the manipulator's end-effector in Y-X and Z-X.

Figure 10: SISO adaptive controllers and the human operator (3-D case). \( M_{\text{SP}}[0] \) and \( M_{\text{SP}}[1] \) are commanded by autonomous visual servoing modules (shared control). The human operator has little experience in using the joystick. This figure shows the trajectories of the moving object and the manipulator's end-effector with respect to the world frame.

Figure 11: SISO adaptive controllers and the human operator (previous example). \( M_{\text{SP}}[0] \) and \( M_{\text{SP}}[1] \) are commanded by manual control while \( M_{\text{SP}}[2] \) is commanded by autonomous visual servoing modules (shared control). The human operator has little experience in using the joystick. This figure shows the relative trajectories of the object and the manipulator's end-effector in Y-X and Z-X.

Figure 12: SISO adaptive controllers and the human operator (3-D case). \( M_{\text{SP}}[0] \) and \( M_{\text{SP}}[1] \) are commanded by manual control while \( M_{\text{SP}}[2] \) is commanded by autonomous visual servoing modules (shared control). The human operator has little experience. This figure shows the trajectories of the moving object and the manipulator's end-effector with respect to the world frame.

Figure 13: SISO adaptive controllers and the human operator (previous example). \( M_{\text{SP}}[0] \) and \( M_{\text{SP}}[1] \) are commanded by manual control while \( M_{\text{SP}}[2] \) is commanded by autonomous visual servoing modules (shared control). The human operator has little experience. This figure shows the relative trajectories of the object and the manipulator's end-effector in Y-X and Z-X.