IEEE Control Systems Magazine, 15(1), pp. 30-39, 1995

Robot Hand-Eye Coordination Based on Stereo Vision

Gregory D. Hager, Wen-Chung Chang and A. S. Morse

Abstract

This article describes the theory and implementation of a system that positions a robot manipulator using visual information from two cameras. The system simultaneously tracks the robot end-effector and visual features used to define goal positions. An error signal based on the visual distance between the end-effector and the target is defined and a control law that moves the robot to drive this error to zero is derived. The control law has been integrated into a system that performs tracking and stereo control on a single processor with no special purpose hardware at real-time rates. Experiments with the system have shown that the controller is so robust to calibration error that the cameras can be moved several centimeters and rotated several degrees while the system is running with no adverse effects.

Introduction

Over the past several years, the use of sensor feedback in robotic systems has been an active area of research. However, the development of visual feedback mechanisms has continually lagged behind other areas of sensor feedback research. Vision systems are characteristically viewed as computationally expensive, error-prone, and difficult to calibrate. This is largely an artifact of using vision to measure *absolute position* in the robot frame of reference. In order for position estimates to be accurate, the vision system must be extremely well calibrated at all times.

This paper describes an approach to stereo visual servoing where vision is used to *measure error.* Error is defined to be the visual distance between a robot manipulator and a pre-defined position in visual coordinates. Error is computed continuously using *visual tracking.* This approach has a number of interesting features:

- Since there is no longer any explicit computation of absolute position, the system is much less dependent on calibration, and hence is robust to calibration error. This makes the approach practical to use with free-standing cameras as well as with cameras rigidly fixed with respect to the robot. In practice, we have never found it necessary to perform an accurate calibration of our system.

[0]A version of this article was presented at the 1994 IEEE International Conference on Systems, Man and Cybernetics. Greg Hager is with the Department of Computer Science, Yale University, P.O. Box 208285 Yale Station, New Haven, CT 06520-8285. W.C. Change and A.S. Morse are with the Department of Electrical Engineering, Yale University P.O. Box 208267, New Haven, CT 06520-8267. Chang and Morse were supported in part by the NSF under grant ECS-9206021 and by the U.S. AFOSR under grant F49620-92-J-0077. Hager was supported by ARPA grants N00014-91-J-1577 and N00014-93-1-1235, by National Science Foundation grants IRI-9109116 and DDM-9112458, and by funds provided by Yale University.

a b c

Figure 1: On the left, a visual servoing system consisting of two cameras on pan-tilt heads connected via a vision-based controller to a robot arm. The middle and right diagrams illustrate positioning by reducing visual disparity to zero.

- The use of stereo makes it possible to position in three dimensions using minimal information. Furthermore, as long as the system is stable, the final positioning accuracy is only dependent on the *physical* positions of the cameras, irrespective of system calibration errors.
- Target positions for the manipulator are defined using a visual tracking mechanism. Tracking simplifies the vision problem, and makes task specifications immune to changes in position or attitude of the target station.

Furthermore, since visual feedback is implemented in the servo loop, it is possible to combine visually guided motion with position or force feedback. Such feedback modes are extremely useful when performing constrained motion tasks. Finally, it appears that *adaptive* feedback controllers that tune the calibration while the system runs can be developed. Since the system itself is quite robust to calibration errors, adaptive calibration tuning is likely to perform well in real settings.

The remainder of this article is structured as follows. The next section discusses stereo-based positioning in more detail and reviews previous research in visual servoing. In the subsequent section, we formalize the visual control problem, present solutions, and discuss the robustness of the result. We then describe experimental results obtained from a system implementing the visual control algorithms. We close with a discussion of open research problems in this area.

An Overview of the Approach

Figure 1(a) depicts a generic vision-based servoing architecture. The major components are two video cameras on free-standing pan-tilt heads, a robot arm, and computers that perform image-processing, low-level control operations, and other interface-related functions. The "classical" approach to stereo-based positioning assumes the cameras can be calibrated to the robot coordinate system. The positions of observed features are then computed using stereo triangulation, and the robot is servoed to the estimated position. However, in the system shown above, the mapping from robot coordinates to images includes the camera intrinsic parameters (image center, image scale factors, distortion coefficients, and focal length) and extrinsic (positional) parameters for both

cameras. It also includes parameters describing the kinematics of the pan-tilt heads and the kinematics of the robot arm itself. Because of the large number of parameters, modeling errors, and mechanical backlash, even a careful and intensive calibration process will often produce only an imprecise estimate of the hand-eye relationship. In many cases, stereo estimates of point positions will be much less accurate than the positioning capabilities of the robot itself. Systems that use fixed cameras (eliminating pan-tilt inaccuracies), or cameras mounted on the arm itself (eliminating much of the kinematics of the robot arm) can achieve higher accuracy. These configurations have the disadvantage that they limit the system to working configurations where the manipulator is both within the field of view and not obscured from the cameras.

The main point of this paper is that by using visual feedback it is possible to achieve extremely accurate positioning of a servo system *relative to an observed target* despite calibration error. Figures 1(b) and 1(c) illustrate the basic principle for translational positioning. In 1(b), the cameras observe the distances θ_1 and θ_2 between a point on the gripper and the corner of the box. We refer to these angles as the *target-goal disparity,* or disparity for short.[1] The fundamental invariant that we exploit is that *zero disparity between the manipulator and the goal in both images means that they are at the same point in space* as shown in 1(c). This statement is true independent of the locations of the cameras as long as they are not coincident, and the goal and the cameras are not collinear.

Related work

Visual servoing has been an active area of research over the last 30 years with the result that a large variety of experimental systems have been built (see [3] for an extensive review and [7, 13] for a recent collection of articles). In the recent literature, Zheng *et al.* [29] describe a visual servoing system using window-based feature tracking on a monocular image stream. Allen *et al.* [1] describe a stereo motion based system for grasping a toy train using two stationary cameras. Rizzi and Koditschek [24] describe a stereo vision-based juggling system, and Anderson [2] describes a stereo vision-based ping-pong player. None of these systems directly observe the robot end-effector; they only observe and compute a goal position or goal configuration of the system. Moving to the computed position is open-loop with respect to the vision system, and therefore relies on the hand-eye calibration.

Weiss *et al.* [26] proposes a system utilizing visual feedback for controlling a manipulator and simultaneously using adaptive control methods to perform online calibration. More recently, other authors have seized on the idea of visual feedback. Many authors address the problem of servoing a camera to maintain a constant position relative to a known object with a single camera [5, 12, 16, 23, 28, 29]. Wijesoma *et al.* [27] describe a monocular hand-eye system for planar positioning using image feedback. Other authors [21, 22] describe monocular visual servoing systems that use some type of adaptivity to compensate for unknown system parameters or to refine system calibration.

Maru *et al.* [18] describe algorithms for performing stereo vision-based positioning. The principle differences between their approach and the results of this paper are that they use cameras mounted on the robot end-effector, they use extremely simple binary image processing, and their model of the camera system assumes that the cameras are aligned with one another and are perpendicular to the stereo baseline. In addition, the configuration of their system makes it impossible to observe

[1] Note that this differs from the classical definition of disparity as the difference in the location of a single point in two images.

the robot end-effector or objects held within it, so they cannot define visual errors by observing both the robot and a desired goal position. Hence, any application of their system will depend on knowing an exact hand-eye calibration.

A stereo-based visual servoing system using free-standing cameras is described in a recent paper by Hollinghurst and Cipolla [14]. Their approach uses an affine approximation to the inverse perspective transformation to compute the approximate Cartesian positions of both the end-effector and the goal position. They then compute control signals based on the difference between these positions. In this article, error is defined in image coordinates and the camera projective mapping is not approximated.

A Visual Feedback Controller

As noted above, the mapping from robot motions to visual motions includes the robot kinematics, a hand-eye coordinate transformation, and the camera perspective map. We model the robot as a Cartesian positioning device with negligible dynamics, and cast the problem of visual feedback control as one of generating Cartesian velocities from image observations.

In the remainder of this article vectors and vector functions are denoted by lower-case letters, and matrices and matrix functions are denoted by upper-case letters. All vectors are column vectors. We write x' for the transpose of x. The denotation $\vec{x}$ indicates that x is a unit vector. We consider two cameras with indices 1 and 2. To simplify the presentation, both cameras are assumed to have unit focal length. We employ the following nomenclature:

$\mathcal{W}$:	The robot base coordinate system.
c_i:	The position of camera i relative to $\mathcal{W}$.
$\vec{x}_i,\ \vec{y}_i,\ \vec{z}_i$:	The coordinate axes for camera i relative to $\mathcal{W}$.
R_i:	The rotation matrix with rows $\vec{x}_i,\ \vec{y}_i,$ and $\vec{z}_i$.
r:	Cartesian coordinates relative to $\mathcal{W}$ of an observable reference point on a robot.
u:	The robot control input.
θ:	Stereo image coordinates.
g:	The mapping from Cartesian coordinates to stereo image coordinates.

Camera coordinate directions are established as follows: $\vec{x}$ points to the right and $\vec{y}$ points downward in the camera imaging plane, and $\vec{z} = \vec{x} \times \vec{y}$ points outward along the camera optical axis. The camera position is defined by the optical center of the lens system. The term *camera calibration parameters* denotes the collection R_1, R_2, c_1, and c_2. The *baseline* of the camera system is the line segment from c_1 to c_2. The *field of view* of the camera is the open cone defined by the camera center and a rectangle the size of the camera image located one unit along the optical axis. The *workspace* of two cameras is the intersection of their fields of view.

The observation of a point r under stereo perspective projection is given by:

$$\theta = \begin{array}{rcl} \theta_{1,x} & = & \dfrac{(r-c_1)\cdot\vec{x}_1}{(r-c_1)\cdot\vec{z}_1} \\ \theta_{1,y} & = & \dfrac{(r-c_1)\cdot\vec{y}_1}{(r-c_1)\cdot\vec{z}_1} \\ \theta_{2,x} & = & \dfrac{(r-c_2)\cdot\vec{x}_2}{(r-c_2)\cdot\vec{z}_2} \end{array} = g(r). \tag{1}$$

$$\theta_{2,y} = \frac{(r - c_2) \cdot \vec{y}_2}{(r - c_2) \cdot \vec{z}_2}$$

Note that stereo projection is a nonlinear mapping which generates four outputs from three inputs.

We assume that the reference point r and a Cartesian setpoint r^* are observable according to (1). The control problem we consider is to develop a regulator to move r to r^* using stereo camera observations. Let $\theta = g(r)$ as above, and let $\theta^* = g(r^*)$ denote the desired or "setpoint" value of θ. Define the error $e_\theta = \theta - \theta^*$ and the error integrating subsystem $\dot{z}_\theta = e_\theta$. The control synthesis task is to choose u as a function of r, z_θ, e_θ to stabilize the system

$$\dot{z}_\theta = e_\theta \tag{2}$$
$$\dot{e}_\theta = J_g(r)u \tag{3}$$
$$\dot{r} = u, \tag{4}$$

where $J_g(r)$ is the Jacobian of g at r. Defining $D_i = (r - c_i) \cdot \vec{z}_i$, we have

$$J_g(r) = \begin{bmatrix} \frac{\vec{x}_1' D_1 - \vec{z}_1'((r-c_1)\cdot\vec{x}_1)}{D_1^2} \\ \frac{\vec{y}_1' D_1 - \vec{z}_1'((r-c_1)\cdot\vec{y}_1)}{D_1^2} \\ \frac{\vec{x}_2' D_2 - \vec{z}_2'((r-c_2)\cdot\vec{x}_2)}{D_2^2} \\ \frac{\vec{y}_2' D_2 - \vec{z}_2'((r-c_2)\cdot\vec{y}_2)}{D_2^2} \end{bmatrix}. \tag{5}$$

Assuming $J(r)'J(r)$ is nonsingular, we can use a Position-Integral (PI) control law [6] to compute u as

$$u = -(J_g(r)'J_g(r))^{-1}J_g'(r)(k_1 e_\theta + k_2 z_\theta), \tag{6}$$

where k_1 and k_2 are positive gain constants. Empirically, we have found that a controller of this form will provide desired tracking provided the values of k_1 and k_2 are chosen appropriately. However, since J_g is not square, it is difficult to prove asymptotic stability of the system. Furthermore, it is computationally expensive to compute the left inverse.

To simplify the problem, we introduce a projection $\pi : \Re^4 \rightarrow \Re^3$ on system outputs. There are several possible choices for this mapping depending on the geometry of the camera system. Two of these choices are discussed later in this article.

Let $\phi = \pi(g(r))$ and let $\phi^* = \pi(g(r^*))$. As before, define the error $e_\phi = \phi - \phi^*$ as well as the error integrating subsystem $\dot{z}_\phi = e_\phi$. The control synthesis task is to choose u as a function of r, z_ϕ, e_ϕ to stabilize the system

$$\dot{z}_\phi = e_\phi \tag{7}$$
$$\dot{e}_\phi = J_\pi(r)u \tag{8}$$
$$\dot{r} = u, \tag{9}$$

where $J_\pi(r)$ is now the Jacobian of the composition $\pi(g(r))$. Assuming $J_\pi(r)$ is nonsingular, u is computed as

$$u = -J_\pi(r)^{-1}(k_1 e_\phi + k_2 z_\phi). \tag{10}$$

We see by substitution that

$$\dot{e}_\phi = J_\pi(r)u = -J_\pi(r)J_\pi(r)^{-1}(k_1 e_\phi + k_2 z_\phi), \tag{11}$$

which implies that

$$\ddot{e}_\phi = -k_1 \dot{e}_\phi - k_2 e_\phi. \tag{12}$$

This is a second order system, and hence it is asymptotically stable. Furthermore, desired tracking will be achieved even if the camera calibration parameters are approximately correct, provided the approximation errors do not destabilize the system.

We note that it is conceptually straightforward to include the forward kinematics of the robot in the stereo projective map, g. The system state becomes the joint positions of the servo-system, and the control signal is joint velocity. With minor changes, the same controller structure can be used. It is also possible to include rotations of the robot in the formulation of stereo projection. When the point r is not at the axis of rotation, this provides additional degrees of freedom of motion. The interested reader may wish to consult [18] for the formulation of such a system for stereo cameras mounted on a robot arm.

Estimation

Thus far, it has been assumed that the value of r, the robot reference position, is known. This information is not directly accessible and must be estimated. We compute an estimate by first rewriting (1) in the form:

$$A(\theta)r = b(\theta) \tag{13}$$

where

$$A(\theta) = \begin{bmatrix} \vec{z}_1'\theta_{1,x} - \vec{x}_1' \\ \vec{z}_1'\theta_{1,y} - \vec{y}_1' \\ \vec{z}_2'\theta_{2,x} - \vec{x}_2' \\ \vec{z}_2'\theta_{2,y} - \vec{y}_2' \end{bmatrix} \tag{14}$$

and

$$b(\theta) = \begin{bmatrix} (\vec{z}_1'\theta_{1,x} - \vec{x}_1')c_1 \\ (\vec{z}_1'\theta_{1,y} - \vec{y}_1')c_1 \\ (\vec{z}_2'\theta_{2,x} - \vec{x}_2')c_2 \\ (\vec{z}_2'\theta_{2,y} - \vec{y}_2')c_2 \end{bmatrix}. \tag{15}$$

Assuming $A(\theta)$ has rank 3, an initial estimate can be computed by solving for r yielding

$$\hat{r}(0) = (A'(\theta)A(\theta))^{-1}A'(\theta)b(\theta). \tag{16}$$

This estimate can be updated using a combination of prediction based on u and correction based on (13):

$$\dot{\hat{r}} = u - k_3 A'(\theta)(A(\theta)\hat{r} - b(\theta)), \tag{17}$$

where k_3 is a positive gain constant tuned based on the noise characteristics of the system. The resulting estimated values of r are used in (6) or (10) as appropriate. Defining $e_r = \hat{r} - r$, it follows from the expressions above that

$$\dot{e}_r = -k_3 A'(\theta) A(\theta) e_r$$

and so the components of e_r will go to zero exponentially fast if $A(\theta)$ is has rank 3. The rank structure of $A(\theta)$ is discussed below.

Reducing System Dimensionality

The mapping π was introduced to reduce the dimensionality of the system. There are two generally accepted choices for π. The first is to define

$$\pi(\theta) = (A'(\theta)A(\theta))^{-1} A'(\theta) b(\theta), \tag{18}$$

where A and b are given by (14) and (15), respectively. This approach leads to what is commonly referred to as "position-based" control [26]. As the name suggests, the error term is now computed in robot "position" space. The major advantage of this approach is that, if the system calibration is correct, setpoints correspond to positions in the global coordinate system. Furthermore, point-to-point trajectories will be straight lines in Cartesian space. However, the error computation requires two left inverse computations (one for the robot control point and one for the setpoint), and so is relatively expensive to compute. Also, calibration errors distort the geometry of the inverse projection so that in practice the trajectories become curves, and the computed coordinates do not correspond to global coordinates.

A second approach to choosing π is based on an analysis of the geometry of stereo projection. A particularly simple case arises when both cameras have parallel coordinate systems in which the x axis is parallel to the stereo baseline. It is easy to show that $\theta_{1,y}$ and $\theta_{2,y}$ are identical in this configuration, and hence one of these components can be deleted from the projection equations. As a result, the system Jacobian and the matrix A used to estimate r are square and invertible. The same approach can be used even when the cameras deviate from the configuration specified above, provided the deviation is not extremely large. Most stereo camera systems have this geometry.

It is possible to define a function which, given the camera calibration parameters, computes three independent values from a stereo projection without making any assumptions about the geometry of the system [15]. However, the cost of this transformation is comparable to that of computing position-based control.

System Stability

In order for vision-based control to be stable, the Jacobian of the projective map, (5), must be of rank 3. By inspection, we see that J_g looses rank globally if $c_1 = c_2$. This configuration is not physically attainable. J_g also looses rank any time r is collinear with c_1 and c_2. In most applications, this configuration is also not physically attainable because it is out of the field of view of the cameras. It is easy to show that the rows of $J_g(r)$ are identical to the rows of $A(g(r))$ modulo a scale factor. It follows that A looses rank in precisely the same situations as J_g.

When the visual workspace does not include the optical center of either camera, J_g is nonsingular on the entire workspace. A is also nonsingular, so r can be estimated directly using (16). It follows that the control systems discussed above can be made stable for motions that remain in

the workspace. In particular, if only proportional control is used, the trajectory of the system in each image is a straight line from the initial point to the setpoint. Thus, motions that originate in the field of view and move to a point in the field of view remain entirely within the field of view. Hence, for proportional control, the systems described above can be made asymptotically stable on a visual workspace that does not include the center of either camera.

System stability when the workspace includes visual singularities remains an open problem. System stability also depends on the accuracy of system calibration as well as the sampling rate and time-delay present in the implementation. These issues are discussed below.

Calibration Sensitivity

By designing a visual control system as described above, positioning at an observed target can be performed with an accuracy that depends only on the physical configuration of the system and the accuracy to which setpoints can be detected in images. This follows directly from system stability and the presence of the integrator in the controller. In practice, we have found that stability is itself insensitive to errors in camera calibration.

The camera intrinsic parameters, including focal length, image scale, and image center, can all be determined offline to high accuracy and are fixed for the life of the system [25]. Hence, they do not introduce any substantial calibration error into the system. To examine the effect of errors in the estimates of camera position and orientation on stability, consider a system in which the two camera coordinate frames are aligned with the base coordinate system, and the baseline is parallel to the camera x axis. In this configuration, motion in the y direction is independent of motion in the x-z plane. We simplify the problem by discarding the y component and consider a planar version of the control problem. In all that follows, g is the perspective projection function restricted to the $x-z$ plane, and J_g, A, and b are the planar versions of the corresponding quantities defined above. Note that J_g and A become square two by two matrices in this case.

The control problem is independent of the absolute location of the base coordinate system. Hence, under the constraints stated above, the only remaining translation parameter that might affect stability is the length of the camera baseline. We denote this length by l. The only remaining degree of rotational freedom is rotation in the plane. We denote the orientation of camera i by α_i with the convention that $\alpha_i = 0$ indicates that the optical axis is perpendicular to the baseline. We gather these parameters into the vector $q = [l, \alpha_1, \alpha_2]'$.

By substituting the estimated value of r given by (16) into the Jacobian matrix, we obtain

$$u = -J_g(A(\theta, q)^{-1} b(\theta, q), q)^{-1} e_\theta = -U(\theta, q)^{-1} e_\theta, \tag{19}$$

where we have explicitly included the calibration parameters as an argument on all quantities. Let $\hat{q}$ denote estimated calibration parameters used by the controller. Then the closed loop system is given by

$$\dot{e}_\theta = -U(g(r, q), q) U(g(r, q), \hat{q})^{-1} e_\theta = -M(r, q, \hat{q}) e_\theta. \tag{20}$$

Simplification of this expression using a symbolic mathematics package shows that M can be expressed in the form $M = \hat{l} M'$ where M' does not contain $\hat{l}$. Thus, $\hat{l}$, the estimated camera separation, acts as a scaling factor and does not affect the asymptotic stability of the continuous time system.

To examine the effect of rotation errors, we fix $l = \hat{l} = 1$, $\hat{\alpha}_1 = \hat{\alpha}_2 = 0$, and vary the orientation of the physical cameras according to the constraint $-\alpha_1 = \alpha_2 = \beta$. Positive values of β correspond

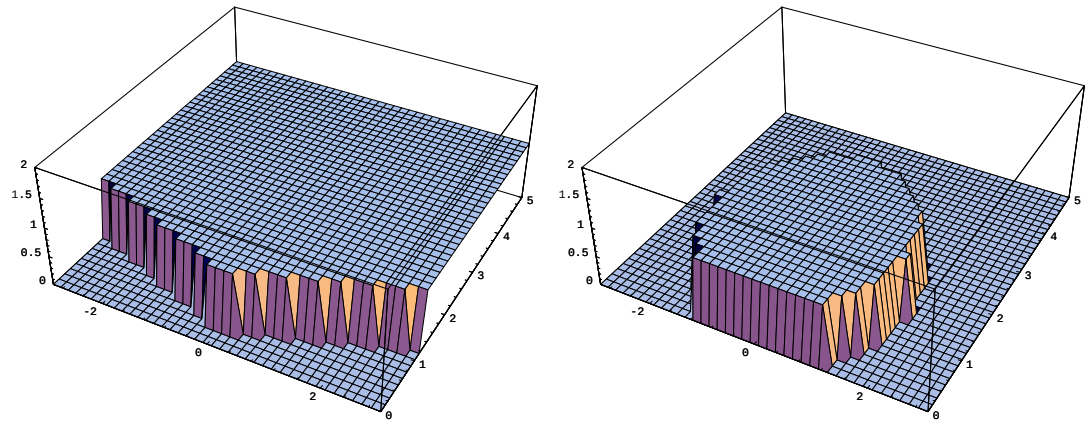

Figure 2: The raised areas indicate the regions of stability when the physical cameras point outward (left) and inward (right) relative to the estimated calibration. The cameras are located along the lower border at $(-1, 0)$ and $(1, 0)$.

to pointing the cameras inward, and negative values correspond to pointing the cameras outward. Figure 2 plots the region of stability of pure proportional control for rotations of $\alpha = -15$ degrees, and $\alpha = 15$ degrees. A rotation outward does not dramatically reduce the region of stability, however a rotation inward quickly introduces unstable points into the camera workspace.

These results can also be verified analytically. Further analysis using a symbolic mathematics package leads to the surprisingly simple result that camera rotations inward ($\beta > 0$) define a region of stability that is a circle with equation

$$(r_z - \cot(2\beta))^2 + r_x^2 = \csc(2\beta)^2. \tag{21}$$

Rotating the cameras outward ($\beta < 0$) defines a region bounded by the lines:

$$\begin{aligned} r_z &= \tan(-\beta)(r_x + 1) \\ r_z &= \tan(\beta)(r_x - 1). \end{aligned}$$

These results suggest that pointing the physical cameras inward relative to their estimated positions will tend to destabilize the system more quickly than pointing them outward. The experimental section contains some empirical observations on this effect.

Choosing Setpoints and Controlling Trajectory

Setpoints for the control algorithm are chosen by tracking visual targets. For example, suppose the goal is to place a manipulator on a fixture located at position p. Locating the fixture in both camera images corresponds to fixing visual coordinates $\theta^* = g(p)$. Servoing the manipulator using one of the control laws described above will move the end-effector so that $r = p$.

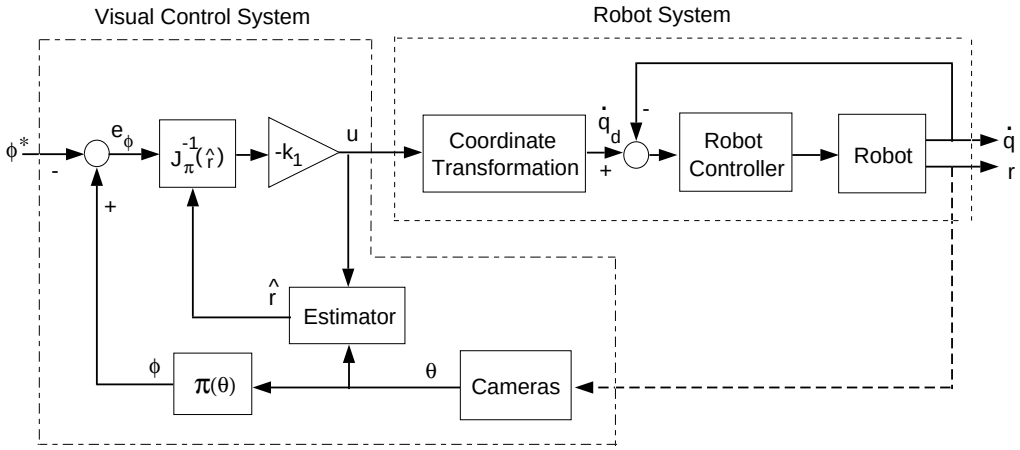

Figure 3: A block diagram of the implemented system indicating the portions which execute on the robot controller and those which execute on the workstation performing tracking and visual control. The mapping π deletes the fourth component of the observation vector. The vector q represents robot joint angles.

Controlled motions from point to point can be defined by using two visual reference points. Let p_1 and p_2 be two observable reference points in the world, and define $\theta_1 = g(p_1)$ and $\theta_2 = g(p_2)$. Define a setpoint that is a linear combination of observed features: $\theta^*(\lambda) = \theta_1 + \lambda(\theta_2 - \theta_1)$. Parameterizing λ according to a desired velocity curve will define a time-varying trajectory for controlled point-to-point motion. With minor modifications, the control algorithms described above can be used to track this trajectory. It is important to note that the trajectory will only be a straight line in *image* space. The Cartesian trajectory of the robot itself will be curved due to the nonlinearity of the perspective mapping.

Performing accurate positioning at a point in space *relative* to observed features is much more difficult than performing positioning *at* observed features. This is because perspective projection does not preserve distances in images, or even simple ratios of distances. However, for small motions, perspective projection can be approximated well by an affine projection [20]. Affine projection preserves collinearity as well as ratios of distances, hence using the notation defined above, controlling the system to the setpoint $\theta^*(\lambda)$ places it at a point approximately $\lambda\|p_1 - p_2\|$ units from p_1 on the line in space containing p_1 and p_2. A companion paper [8] provides more detail on ways for performing relative positioning using perspective images.

Experiments

Our visual servoing system consists of a Zebra Zero robot arm with PC controller, a Zebra pan-tilt head, a Directed Perceptions pan-tilt head, two Sony XC-77 cameras with 12.5mm lenses, and two Imaging Technologies digitizers attached to a Sun Sparc II computer via a Solflower SBus-VME adapter. The workstation and PC are connected by an ethernet link. All image processing and visual control calculations are performed on the Sun workstation. Cartesian velocities are sent to the PC which converts them into coordinated joint motions at 140 hz. The cameras are placed

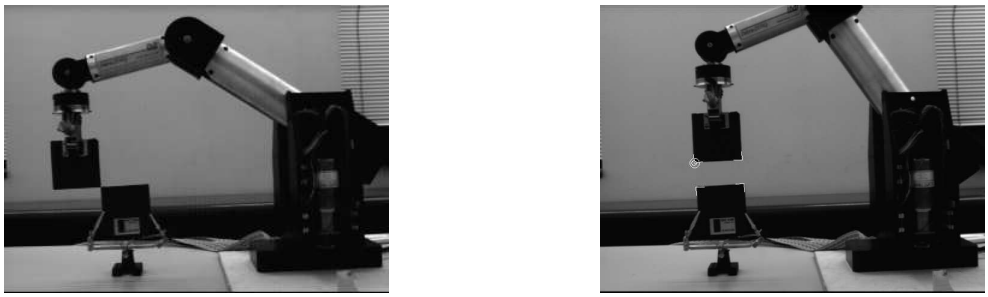

Figure 4: The implemented system during two stages of a visually guided motion trajectory as seen from one of the stereo cameras. Left, the system touching the corners of two floppy disks, and right the system attaining a position placing one disk directly above the other. The right image shows the features that are tracked. The visual setpoints are shown circled.

approximately 90cm from the target along the x axis and 30cm apart along the y axis. They are oriented to point back along the x axis of the robot.

Since we rely on internal robot feedback based on joint encoders to stabilize the system, and the robot controller uses integral control, it is not necessary to use an integrator in the visual controller itself. Thus, all experiments described here are based on pure proportional control. Figure 3 shows a block diagram of the combined system. Unless otherwise noted, the feedback gain was set at 1.0. Because the cameras are nearly parallel, we remove the fourth row of J_g, A, and b to obtain square matrices as discussed previously. The camera workspace is does not contain visual singularities and there is little noise in the system, so robot position is estimated directly using (16). System calibration was performed by tracking the robot through a series of known motions and performing optimization to determine the system calibration parameters relative to robot base coordinates [17].

A custom tracking system written in C++ provides visual input for the controller. The system, more fully described in [11], provides fast edge detection on a memory-mapped framebuffer. In addition, it supports simultaneous tracking of multiple edge segments, and can also enforce constraints among segments. The experiments described here are based on tracking corners formed by the intersection of two line segments. The segment length is 20 pixels, and the search area around a segment is ± 10 pixels. The entire visual control system (including tracking and control signal computation) runs at a rate of approximately 18.5 Hz. At 18.5 Hz, the robot can be tracked at velocities covering up to 185 pixels/sec which, with a 12.5mm lens, converts to maximum velocities of approximately 17cm/sec perpendicular to the camera optical axis at 90 cm.

Images of the experimental setup are shown in Figure 4. The left image shows the system in a goal configuration where it is touching the corners of two 3.5 inch floppy disks. The right image shows the robot reaching a goal position 0.5 disk widths (1.75 inches) above a target disk. The disks are a convenient testing tool since their narrow width (2.5mm) makes them simple to track accurately and the corners provide well-defined setpoints.

The system is tested in two ways. In one set of tests, the robot is started at selected points in the workspace, and servoed to a stationing point. The feature positions and joint positions of the robot are measured for later inspection. In a second set of tests, we control the robot along a square trajectory defined by the sides and top of a target disk. The robot manipulator begins by moving the left corner of its disk to touch the right corner of the target disk. Next, it moves to a position where the left corner of its disk is 1.75 inches above the right corner of the target disk. It

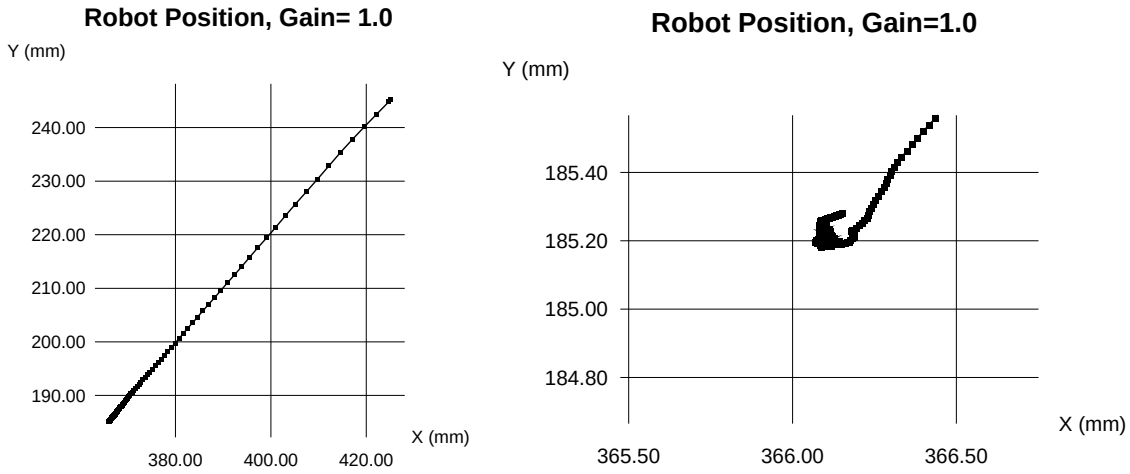

Figure 5: Left, the sequence of positions of the robot as it moves to a setpoint. The time between each dot is 50 milliseconds. Right, an expanded view of the behavior at the setpoint.

then moves the right corner of its disk to a station 1.75 inches above the left corner of the target disk. Finally, it descends to touch the right corner of its disk to the left corner of the target disk. The complete procedure is then reversed, and the entire cycle executed repeatedly. During this experiment, we can alter the position and orientation of the cameras and observe the effects.

Positioning Accuracy

It is possible to predict the ultimate accuracy of positioning based on the error of edge detection [19]. One camera pixel has a width of approximately .013mm. Assuming a maximal error of one pixel and the physical configuration described above, the expected vertical and horizontal positioning accuracy is ± 0.9mm, and the expected accuracy in depth is approximately ± 2.75mm. In practice, the tracking system performs sub-pixel localization, so we expect to achieve better results.

In terms of our physical setup, this means that we expect the manipulator to position the disks so that they overlap. We have found this to always be the case. In hundreds of trails, the observed relative positioning error is typically within a millimeter of depth and a fraction of a millimeter in other directions. This error is measured by positioning the corners of two disks as described above until the robot stabilizes, freezing the robot, and measuring the positioning error using a caliper. We have found the largest source of systematic error to be illumination differences in the two cameras. Large contrast changes cause bright areas to "bleed" into dark areas which biases the edge detector. These problems could be easily corrected by using auto-aperture lenses.

While at a stationing point, the manipulator continues to perform small corrective motions due to noise in the edge tracker. Figure 5 shows a graph of robot positions at a setpoint. The vertical direction corresponds to motion along the robot y axis which is approximately parallel to the camera baseline, and the horizontal direction corresponds to motion along the x axis which is approximately parallel to the camera lines of sight. The dots represent position readings taken each 50 ms. From the graph, we see that the manipulator is nearly motionless at the setpoint.

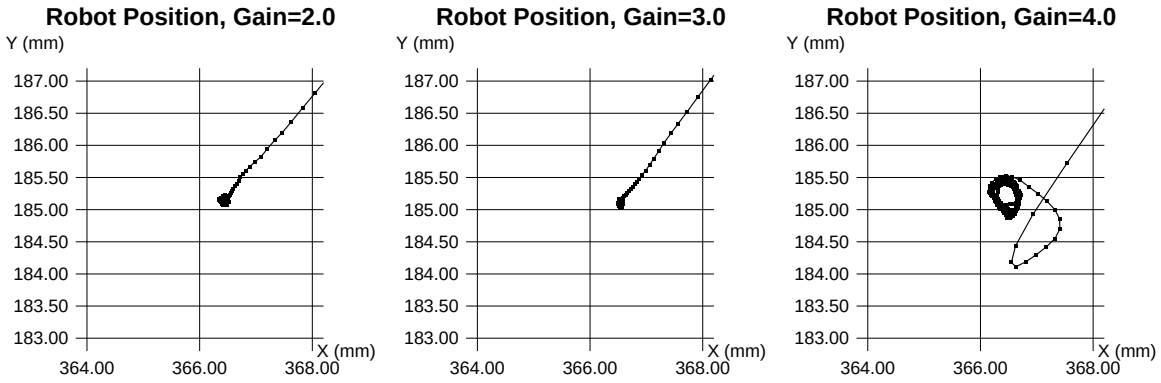

Figure 6: The trajectory of robot while performing point-to-point motion for different feedback gain values.

Stability

The major destabilizing factors in the system are time lag and calibration error. Unmodeled dynamics of the robot do not appear to play a role at speeds of 5cm/sec. To determine the effect of time delay, consider a discrete version of the system with unit time delay and no calibration parameter error:

$$e_{n+1} = e_n + t\ k\ e_{n-1}, \tag{22}$$

where k is a gain coefficient and t is the sampling time (one over the sampling rate). This system has characteristic polynomial $x^2 + x + tk = 0$ [6]. System is stability is guaranteed when $k < 1/t$. The system is overdamped if $tk < 1/4$ and underdamped if $tk > 1/4$. In our case, the delays include 54 milliseconds to calculate a control command, approximately 5 milliseconds to transmit the command to the robot, and up to 4 milliseconds delay until the robot responds. Thus, theory predicts that k must be less than 3.97 to prevent unwanted oscillation at the setpoint.

To verify this analysis, we performed several point-to-point motions at varying gain values. Figure 6 shows the performance of the algorithm for a point-to-point motion for gains of 2.0, 3.0, and 4.0. As expected, at a gain of 4.0 the manipulator overshoots the target, corrects, and then begins to gyrate slightly about the setpoint.

Calibration Sensitivity

Perhaps the most compelling evidence for the stability of the control methods described here is the fact that we have never needed to calibrate the system for normal use. Qualitatively, as long as the cameras are in approximately correct positions, the system performs adequately with unit gain.

As noted in the analysis section above, the system is most sensitive to the relative orientations of the two cameras. In particular, inward rotations of the physical cameras should be most likely to destabilize the system. To test this empirically, we performed a system calibration as described in [17]. We then rotated the cameras approximately 10 degrees inward. The effects are quite marked. As shown in Figure 7, a gain of 1.0 led to large oscillations. However, reducing the gain to 0.5 again stabilized the system. When stable, the positioning precision of the system remained unchanged despite the calibration error.

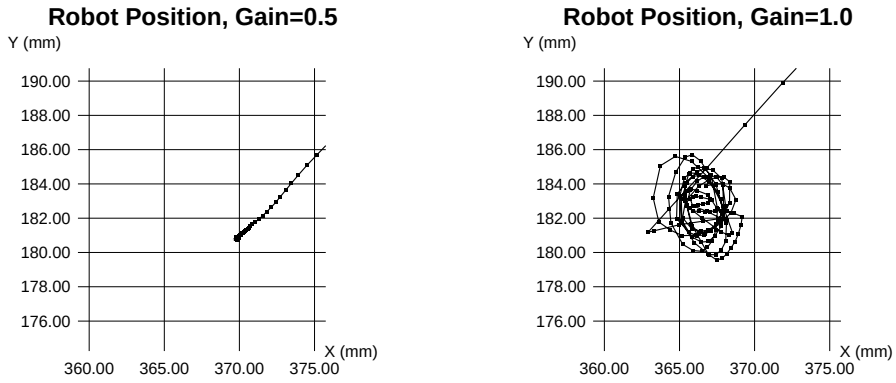

Figure 7: The performance of point-to-point positioning after rotating the cameras inward 10 degrees.

Conclusions

We believe the visual servoing paradigm we have presented has great potential for use in commercial and scientific applications. It is simple, inexpensive, robust, and portable. The current vision processing and control computation system uses no special hardware (other than a standard framegrabber) and can be run on common workstations. At the current rate of progress, frame-rate (60 Hz) servoing will be easily feasible within a year or two. Furthermore, since the entire system, including image processing, runs in software, moving to a newer or more powerful system is largely a matter of recompiling. The robot and camera interfaces are extremely generic, so the system is also easily ported to other robot or imaging hardware. As reported, the current system can easily position the end-effector to within a few millimeters relative to a target. This positioning accuracy could easily be improved by changing the camera configuration to a wider baseline, or by improving the image-processing to be more accurate.

The methods are also extremely robust. A moderately accurate camera calibration can be performed quickly using robot motions to supply input [17]. In most cases, this calibration would be sufficient to provide adequate performance for positioning and short straight-line motions provided the cameras are not moved substantially during operation. We have constructed an adaptive supervisory controller for a planar version of the servoing problem. Initial simulation tests indicate that it is stable, and rapidly calibrates the system as it moves. We are currently moving a full 3D version of the adaptive controller to the real system. If successful, this should make it simple to use mobile or reconfigurable camera systems.

We are continuing to develop other interesting and useful visual control modes and trajectory generation schemes made possible by proper exploitation of image invariants [4] and changing the "focus of attention" of the tracking system. Some early results in this area can be found in [8, 9]. We are also investigating methods for automatically initializing the tracking system using prior knowledge of the scene [10].

References

[1] P. Allen, B. Yoshimi, and A. Timcenko. Hand-eye coordination for robotic tracking and grasping. In K. Hashimoto, editor, *Visual Servoing*, pages 33–70. World Scientific, 1994.

[2] R. L. Anderson. Dynamic sensing in a ping-pong playing robot. *IEEE Transactions on Robotics and Automation*, 5(6):723–739, 1989.

[3] P. I. Corke. Visual control of robot manipulators—a review. In K. Hashimoto, editor, *Visual Servoing*, pages 1–32. World Scientific, 1994.

[4] O. Faugeras. What can be seen in three dimensions with an uncalibration stereo rig? In *Proc. the First European Conference on Computer Vision*, pages 563–578. Springer Verlag, 1993.

[5] J. Feddema, C. Lee, and O. Mitchell. Weighted selection of image features for resolved rate visual feedback control. *IEEE Transactions on Robotics and Automation*, 7(1):31–47, February 1991.

[6] G. Franklin, J. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, 2nd edition, 1991.

[7] G. Hager and S. Hutchinson, editors. *Proc. the Workshop on Visual Servoing*. May 1994.

[8] G. D. Hager. Real-time feature tracking and projective invariance as a basis for hand-eye coordination. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 533–539. IEEE Computer Society Press, 1994.

[9] G. D. Hager. Six DOF visual control of relative position. DCS RR-1038, Yale University, New Haven, CT, June 1994.

[10] G. D. Hager, G. Grunwald, and G. Hirzinger. Feature-based visual servoing and its application to telerobotics. In *Proc. IEEE International Conference on Intelligent Robots and Systems*, pages 164–171. IEEE Computer Society Press, 1994.

[11] G. D. Hager, S. Puri, and K. Toyama. A framework for real-time vision-based tracking using off-the-shelf hardware. DCS RR-988, Yale University, New Haven, CT, September 1993.

[12] K. Hashimoto. LQ optimal and nonlinear approaches to visual servoing. In K. Hashimoto, editor, *Visual Servoing*, pages 165–198. World Scientific, 1994.

[13] K. Hashimoto, editor. *Visual Servoing*. World Scientific, 1994.

[14] N. Hollinghurst and R. Cipolla. Uncalibrated stereo hand eye coordination. Technical Report TR-126, Cambridge University, Dept. of Engineering, September 1993.

[15] B. K. Horn. *Robot Vision*. MIT Press, 1986.

[16] S. Hutchinson. Exploiting visual constraints in robot motion planning. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1722–1727. IEEE Computer Society Press, 1991.

[17] C. Lu, E. J. Mjolsness, and G. D. Hager. Online computation of exterior orientation with application to hand-eye calibration. DCS RR-1046, Yale University, New Haven, CT, August 1994.

[18] N. Maru, H. Kase, A. Nishikawa, and F. Miyazaki. Manipulator control by visual servoing with the stereo vision. In *Proc. IEEE International Conference on Intelligent Robots and Systems*, pages 1866–1870. IEEE Computer Society Press, 1993.

[19] L. Matthies and S. Shafer. Error modeling in stereo navigation. *IEEE Transactions on Robotics and Automation*, RA-3(3):239–248, June 1987.

[20] J. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, 1992.

[21] B. Nelson and P. K. Khosla. Increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance. In *Proc. IEEE International Conference on Robotics and Automation*, pages 418–423. IEEE Computer Society Press, 1993.

[22] N. Papanikolopoulos, P. Khosla, and T. Kanade. Vision and control techniques for robotic visual tracking. In *Proc. IEEE International Conference on Robotics and Automation*, pages 857–864. IEEE Computer Society Press, 1991.

[23] P. Rives, F. Chaumette, and B. Espiau. Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In *Proc. IEEE International Conference on Robotics and Automation*, pages 2248–2254. IEEE Computer Society Press, 1991.

[24] A. Rizzi and D. E. Koditschek. Further progress in robot juggling: The spatial two-juggle. In *Proc. IEEE International Conference on Robotics and Automation*, pages 919–924. IEEE Computer Society Press, 1993.

[25] R. Tsai. A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Transactions on Robotics and Automation*, RA-3:323–344, Aug. 1987.

[26] L. Weiss, A. Sanderson, and C. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Transactions on Robotics and Automation*, RA-3(5):404–417, Oct. 1987.

[27] S. Wijesoma, D. Wolfe, and R. Richards. Eye-to-hand coordination for vision-guided robot control applications. *International Journal of Robot Research*, 12(1):65–78, 1993.

[28] W. Wilson. Visual servo control of robots using Kalman filter estimates of robot pose relative to work-pieces. In K. Hashimoto, editor, *Visual Servoing*, pages 71–104. World Scientific, 1994.

[29] J. Y. Zheng, Q. Chen, and S. Tsuji. Active camera guided manipulation. In *Proc. the IEEE International Conference on Robotics and Automation*, pages 632–638. IEEE Computer Society Press, 1991.