Proceedings of the 1992 IEEE
International Conference on Robotics and Automation
Nice, France - May 1992

Hybrid Vision/Position Servo Control of a Robotic Manipulator

Andrés Castaño Seth Hutchinson
The Beckman Institute for Advanced Science and Technology
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Abstract

This paper addresses a number of issues associated with visual servo control of robotic manipulators. First, we derive a set of projection equations that are used in the derivation of the Jacobian matrix for resolved-rate visual servo control. Following this, we present a calibration procedure that determines those parameters that appear in the projection equations.

Given the projection equations and calibration procedure, we next derive a set of equations that can be used to initially position the manipulator at a specified perpendicular distance from the camera such that the tool center of the end effector projects onto a specified pixel on the image plane.

Finally, we derive a Jacobian matrix that is used to effect hybrid control of the manipulator. Specifically, the two degrees of freedom parallel to the image plane of the camera are controlled using visual feedback, and the remaining degree of freedom (perpendicular to the image plane) is controlled using position feedback provided by the robot joint encoders.

1 Introduction

Sensor-based robotic control is essential if robots are to perform adequately in real-world environments. Although a number of advances have been made in the area of force-based control [2, 3, 10], force-based methods have a fundamental limitation: they can only be used to control motion in directions that are tangent to constraint surfaces in the configuration space [11]. One solution to this limitation is to use vision-based techniques to control motion in the remaining directions. Thus, research attention has recently been focused on vision-based control [1, 6, 4, 12, 14].

One approach to vision-based control that has shown great promise is based on concepts from resolved-rate position control [15]. In this approach, a Jacobian matrix that relates differential changes in the robot's motion to differential changes in the camera image (which is either attached to the robot or fixed in the workspace) is derived [4, 14]. A key issue in determining this Jacobian matrix is the derivation of the projection equations that define the imaging geometry of the camera. In [8], we introduced visual constraint surfaces as a mechanism to control two degrees of freedom (parallel to the image plane) with visual feedback using this type of resolved-rate motion control approach. Position control is used to control motion in the direction perpendicular to the image plane.

In Section 2 we derive the projection equations that are used to construct the rows of the Jacobian matrix that correspond to visual control. In Section 3 we derive the corresponding calibration procedure.

In Section 4, we derive a set of equations that can be used to position the manipulator at a specified perpendicular distance from the camera focal center such that the tool center projects onto a specified pixel in the image plane. These equations are used in open-loop control mode to initially position the manipulator on a visual constraint surface (actually, the accuracy of this technique only facilitates placing the manipulator within a small distance of the visual constraint surface).

In Section 5, we derive the Jacobian matrix that is used for the hybrid control of manipulator motion. To do this, we derive the relationship between differential changes in the world coordinates of the manipulator and the differential changes in the image plane coordinates of the projection of the manipulator tool center (vision), and the relationship between differential changes in the world coordinates of the manipulator and the differential changes in the perpendicular distance from the tool center of the manipulator to the focal center of the camera (position).

Finally, in Sections 6 and 7 we describe our progress with an experimental work cell, and summarize the contributions of the work to date.

2 Projection equations

In order to perform visual servo control, the relationships between the robot's workspace and the camera image plane must be known. These relationships are defined in terms of a set of projection equations, which define how points in the workspace project onto the camera image plane via the imaging geometry of the camera. In this section, we derive the projection equations for the robotic system shown in figure 1. They express the relationship between points in the robot's workspace and their projections on the camera image plane. This can be viewed as a transformation between the world coordinate frame and the camera image plane coordinate frame. These two frames are defined as follows:

0-8186-2720-4/92 $3.00 ©1992 IEEE

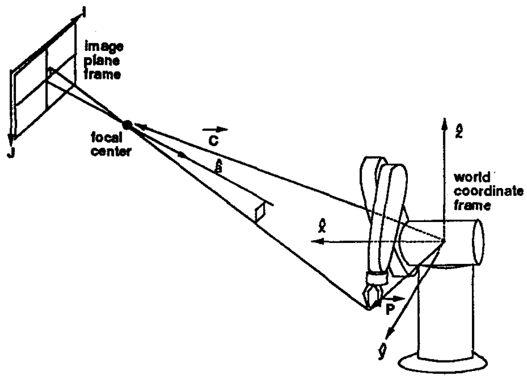

Figure 1: World and image plane coordinate systems.

World Coordinate Frame For the base frame of the robot workspace, we use the standard base frame for the PUMA 560 robot [5].

Image Plane Coordinate Frame The image plane coordinate system is defined by the four vectors $\vec{C}, \hat{h}, \hat{v}, \hat{a}$, where $\vec{C}$ is the position of the focal point of the camera lens (with respect to the world frame), $\hat{a}$ is the unit vector perpendicular to the image plane, and $\hat{h}$ and $\hat{v}$ are the unit vectors parallel to the horizontal and vertical directions in the image plane, respectively.

We now derive the relationships between these two coordinate systems, given that the camera projectivity can be modeled as perspective projection. There has been much work done in the area of camera calibration, and consequently in modeling the projective geometry of cameras ([13] gives an excellent overview of the research in this area). In our work, we have chosen to use the model described in [9]. The reasons for this are two-fold. First, in order to perform visual control in real-time, it is necessary that the projection equations (and related equations) can be solved without any numerical iterative processing. Second, because the projection equations will be used only to derive a Jacobian matrix (which is itself a linear approximation), it is not essential for us to use a method that fully models the non-linearities in the imaging process. The derivation given below closely follows that given in [9].

From figure 2, the following relationships between the world frame coordinates of a point $\vec{P} = [x, y, z]^T$, and the u, v coordinates of its projection hold:

$$\frac{u}{f} = \frac{\vec{D} \cdot \hat{h}}{\vec{D} \cdot \hat{a}}, \quad \frac{v}{f} = \frac{\vec{D} \cdot \hat{v}}{\vec{D} \cdot \hat{a}} \tag{1}$$

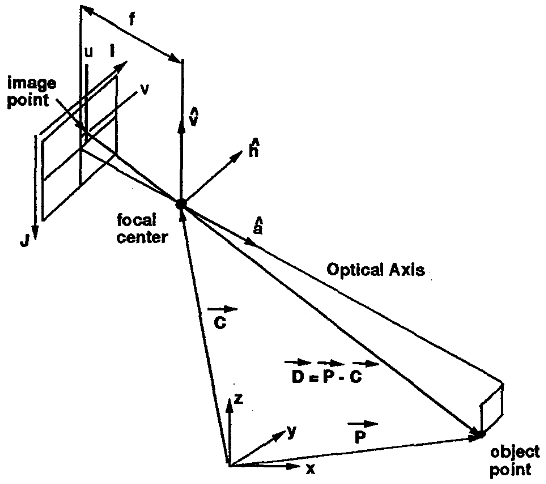

Figure 2: Configuration of coordinate frames in the world.

where f is the focal length of the camera and

$$\vec{D} = \vec{P} - \vec{C} \tag{2}$$

is the vector from the camera focal point to the point $\vec{P}$.

Using standard video hardware (digitizers, frame grabbers), we don't have direct access to the u, v coordinates of an image plane point. Rather, we have access to the discrete coordinates that represent the horizontal and vertical indices in an image array. We use I, J to denote these coordinates. Let Δu and Δv be the horizontal and vertical sampling intervals. Then:

$$u = (I - I_0)\Delta u, \quad v = -(J - J_0)\Delta v \tag{3}$$

where I_0 and J_0 are the coordinates of the center of the image plane frame that correspond to the origin of the image plane. (The direction of the J axis in the figure is opposite the direction of the v axis. This accounts for the difference in the signs in eq. (3). The directions of I, J have been chosen in this way for consistency with our current set of video hardware)

Substituting (3) into (1) we obtain:

$$I = \frac{\vec{D} \cdot \vec{H}}{\vec{D} \cdot \hat{a}}, \quad J = \frac{\vec{D} \cdot \vec{V}}{\vec{D} \cdot \hat{a}} \tag{4}$$

with

$$\vec{H} = \frac{f}{\Delta u}\hat{h} + I_0\hat{a}, \quad \vec{V} = -\frac{f}{\Delta v}\hat{v} - J_0\hat{a}$$

Note that $\vec{H}$ and $\vec{V}$ are *not* unit vectors in the horizontal and vertical directions of the image plane. They are vectors that represent composite information regarding the horizontal and vertical directions of the image plane, sampling intervals of the camera, the focal length of the camera, and the image plane coordinates of the origin of the image plane (i.e. the image plane point defined by the focal point of the camera and the vector $-\hat{a}$).

Using (2), eq. (4) can be rewritten as:

$$I = \frac{\vec{P}\cdot\vec{H} - \vec{C}\cdot\vec{H}}{\vec{P}\cdot\hat{a} - \vec{C}\cdot\hat{a}}, \quad J = \frac{\vec{P}\cdot\vec{V} - \vec{C}\cdot\vec{V}}{\vec{P}\cdot\hat{a} - \vec{C}\cdot\hat{a}}$$

These projection equations are defined in terms of the four camera vectors $\vec{C}$, $\vec{H}$, $\vec{V}$, and $\hat{a}$. We may rewrite these equations as:

$$\begin{aligned} \vec{P}_m\cdot\vec{H} - \vec{C}\cdot\vec{H} - I_m\vec{P}_m\cdot\hat{a} + I_m\vec{C}\cdot\hat{a} &= 0 \quad (5)\\ \vec{P}_m\cdot\vec{V} - \vec{C}\cdot\vec{V} - J_m\vec{P}_m\cdot\hat{a} + J_m\vec{C}\cdot\hat{a} &= 0 \end{aligned}$$

where $\vec{P}_m = [x_m, y_m, z_m]^T$ is a specific point in the workspace and (I_m, J_m) are the coordinates of its projection on the image plane.

Prior to calibration, the unknowns in (5) are $\vec{C}$, $\vec{H}$, $\vec{V}$ and $\hat{a}$, so the equations are nonlinear as functions of the unknowns. We linearize this system by introducing the following variables:

$$\left\{ \begin{aligned} C_a &= \vec{C}\cdot\hat{a} \\ C_H &= \vec{C}\cdot\vec{H} \\ C_V &= \vec{C}\cdot\vec{V} \end{aligned} \right. \qquad (6)$$

Now, rewrite (5) as a linear system of equations:

$$\begin{aligned} \vec{P}_m\cdot\vec{H} - C_H - I_m\vec{P}_m\cdot\hat{a} + I_m C_a &= 0 \quad (7)\\ \vec{P}_m\cdot\vec{V} - C_V - J_m\vec{P}_m\cdot\hat{a} + J_m C_a &= 0 \end{aligned}$$

This is the form of the projection equations used by the calibration procedure, which will produce values for the unknowns C_a, C_V, C_H, $\vec{H}$, $\vec{V}$, and $\hat{a}$. If we desire to obtain $\vec{C}$, we solve the system of equations given by eq. (6).

Finally, we rewrite the projection equations in terms of the parameters that are found directly by the calibration procedure as:

$$I_m = \frac{\vec{P}_m\cdot\vec{H} - C_H}{\vec{P}_m\cdot\hat{a} - C_a}, \quad J_m = \frac{\vec{P}_m\cdot\vec{V} - C_V}{\vec{P}_m\cdot\hat{a} - C_a} \qquad (8)$$

Using these equations, once calibration has been performed we never need to consider a number of parameters, such as the horizontal or vertical sampling intervals of the camera or the focal length of the lens.

3 The calibration procedure

Now that we have the projection equations for the camera, we show a calibration procedure that will produce the desired calibration parameters. To do this, we solve the double system of linear equations (7). To construct the system, we collect a set of world points and the corresponding image plane coordinates (i.e. a set of $\vec{P}_m$ and the corresponding I_m, J_m). Unfortunately, the resulting system of equations is homogeneous, and therefore admits a trivial solution. To obtain a non-trivial solution, we divide each component of (7) by a_x to obtain:

$$\begin{aligned} x_m H'_x + y_m H'_y + z_m H'_z - C'_H & \qquad (9)\\ -I_m y_m a'_y - I_m z_m a'_z + I_m C'_a &= I_m x_m \end{aligned}$$

We show only equations for I. Corresponding equations for J are obtained by replacing I_i with J_i and H_k by V_k for $k = x, y, z$.

Provided that $a_x \neq 0$, this non-homogeneous system leads to a non-trivial solution. Eq. (9) can be expressed in matrix form as:

$$W_I \cdot U'_I = B_I \qquad (10)$$

where

$$W_I = \begin{bmatrix} x_1 & y_1 & z_1 & -I_1 y_1 & -I_1 z_1 & -1 & I_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & z_N & -I_N y_N & -I_N z_N & -1 & I_N \end{bmatrix}$$

and

$$U'_I = \begin{bmatrix} H'_x \\ H'_y \\ H'_z \\ a'_y \\ a'_z \\ C'_H \\ C'_a \end{bmatrix}, \qquad B_I = \begin{bmatrix} I_1 x_1 \\ \vdots \\ I_N x_N \end{bmatrix}$$

Eq. (10) and the corresponding equation for J have seven unknowns each. Therefore seven sample points (x, y, z) with their corresponding (I, J) values would suffice to solve them. However, since noise is introduced into the imaging process (e.g. by sampling the image plane at discrete intervals), we use a least squares solution technique to solve these equations by solving the following system:

$$\begin{aligned} U'_I &= [(W_I)^T(W_I)]^{-1}(W_I)^T B_I \qquad (11)\\ U'_J &= [(W_J)^T(W_J)]^{-1}(W_J)^T B_J \end{aligned}$$

Provided the $\vec{P}_m$ are not coplanar, the accuracy of the solution improves as the number of data points is increased. In practice, we have used 15 to 20 data points with good results.

Once the least squares solutions for U'_I and U'_J have been obtained, they must be scaled to account for the fact that we divided by a_x to obtain a non-homogeneous system as follows:

$$U_I = a_x U'_I, \qquad U_J = a_x U'_J \qquad (12)$$

where $0 < \hat{a}\cdot(\vec{P}_m - \vec{C})$ for all $\vec{P}_m$ holds and

$$a_x = \pm\sqrt{\frac{1}{1 + a'^2_y + a'^2_z}}$$

A number of issues arise when implementing the calibration process described above. We briefly describe three of these issues here.

1. Singular Value Decomposition (SVD) was used to solve eq. (11) due to its ability to handle matrices which are numerically very close to singular. However, even using SVD, double precision arithmetic was required at all intermediate steps of the algorithm to achieve consistent results.

2. Regardless of accuracy, the construction of eq. (9) fails when the value of a_x is 0, or very close to it. These small values of a_x lead invariably to numerical problems. In this case divide (7) by a_y or a_z.

3. Both U_I and U_J lead to values for $\hat{a}$ and C_a. Although the values of $\hat{a}$ and C_a obtained from eqns. (12) tend to be very similar, they are rarely equal, due to loss of information caused by the image plane sampling (one equation uses discretized I values while the other uses J values). In practice, we have found the difference in the final results obtained by using either set to be negligible.

4 Open-loop positioning

By using a hybrid control scheme in which the joint encoders are used for position control in the direction perpendicular to the image plane, we can force the robot manipulator to move along a linear trajectory defined by a projection ray through the focal point of the camera. This results in a type of visually constrained motion that is analogous to compliant motions using force feedback [11]. In this section, we describe how open-loop positioning can be used to initially position the manipulator so that the desired point on the manipulator intersects the desired projection ray.

The goal of the initial positioning of the manipulator is to place some point on the manipulator, say $\vec{P}_m$, so that it projects onto the desired image plane point. Stated another way, given input I_m, J_m, compute the (x_m, y_m, z_m) workspace coordinates for the point $\vec{P}_m$. The immediate problem that we face is that the projection equations given in eq. (8) define a many-to-one mapping from the robot workspace to the image plane. The inverse mapping takes single image plane points and maps them to projection rays. Thus, in order to solve for (x_m, y_m, z_m), we must supply a third parameter, which is used to select a single point on the given projection ray. We will use d_I, the perpendicular distance from the focal point of the lens to the desired workspace point. Thus, solving for (x_m, y_m, z_m) amounts to computing the intersection of a projection ray with a plane parallel to the image plane. This is accomplished by solving a system of three simultaneous equations.

The first two equations that are required are simply the projection equations for the camera, as given in eq. (5). During calibration, the unknowns were $\vec{H}$, $\vec{V}$, $\hat{a}$, C_a, C_V, and C_H. After calibration, these values are known, and (5) can be rewritten to solve for x_m, y_m and z_m given (I_m, J_m):

$$\begin{aligned}(H_x - I_m a_x)x_m + (H_y - I_m a_y)y_m & \\ +(H_z - I_m a_z)z_m &= C_H - I_m C_a \\ (V_x - J_m a_x)x_m + (V_y - J_m a_y)y_m & \\ +(V_z - J_m a_z)z_m &= C_V - J_m C_a\end{aligned} \qquad (13)$$

The third equation needed to solve the system is the equation of the plane parallel to the image plane at a distance d_I from the focal center. The equation for this plane is given by:

$$\hat{a} \cdot \vec{D} = d_I$$

where $\hat{a}$ and $\vec{D}$ are as defined in Section 2 (see figure 2 for a graphical illustration). In order to solve for (x_m, y_m, z_m), we rewrite $\vec{D}$ as $(\vec{P}_m - \vec{C})$ and obtain:

$$\hat{a} \cdot \vec{P}_m = d_I + \hat{a} \cdot \vec{C} \qquad (14)$$

We will define $d = d_I + \hat{a} \cdot \vec{C}$ to simplify notation. This provides us with an equation for the plane expressed with respect to the robot workspace coordinate frame, namely $\hat{a} \cdot \vec{P}_m = d$.

Using (13) and (14), a system that finds the intersection between a projection ray from the camera and the desired plane in the workspace coordinate frame is established. In matrix form:

$$\mathcal{A} \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \begin{bmatrix} C_H - I_m C_a \\ C_V - J_m C_a \\ d \end{bmatrix} \qquad (15)$$

where

$$\mathcal{A} = \begin{bmatrix} H_x - I_m a_x & H_y - I_m a_y & H_z - I_m a_z \\ V_x - J_m a_x & V_y - J_m a_y & V_z - J_m a_z \\ a_x & a_y & a_z \end{bmatrix}$$

Now we can solve for (x_m, y_m, z_m). That $\mathcal{A}^{-1}$ will exist can be seen from the geometric argument that the three equations define a line (the projection ray) and a plane which, by construction, intersects the projection ray at a single point.

5 Hybrid vision/position control

Although open-loop positioning is useful for initially positioning the manipulator near a target projection ray, it rarely succeeds in precisely placing the manipulator at the desired (I, J) coordinates. There are three reasons for this failure: kinematic errors (i.e. uncertainty due to the resolution of the robot joint encoders, or to robot calibration), camera calibration errors (noise in the imaging process) and errors in the camera modeling (we use a simple pin-hole approximation to the camera in the derivation of the projection equations). To achieve the desired accuracy, closed loop control must be used.

When using the visual constraint surface formalism for closed-loop visual servo control [8], we use vision feedback only to control motion in directions that lie

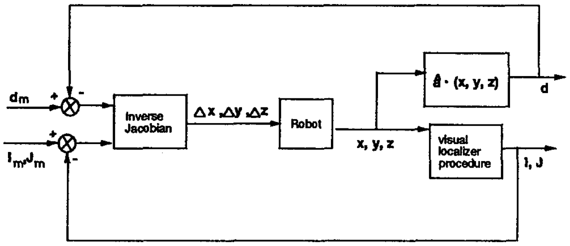

Figure 3: Hybrid-vision/position-control block diagram

in a plane parallel to the camera image plane. To control motion in the direction normal to the image plane, position control is used (where the feedback information is obtained by solving the robot's forward kinematic equations using input from the robot joint encoders). To accomplish this, we use a hybrid control approach. We use a resolved-rate motion control approach [15] in which the first two rows of the Jacobian matrix correspond to vision based control, and the third row corresponds to the position based control. In the remainder of this section, we derive this Jacobian matrix.

We have formulated the control problem as one of controlling the variables I, J, d. Thus, the input to the control system is a vector $[I_m, J_m, d_m]^T$, which would be determined by a trajectory planner (see [4] for a discussion of feature-based trajectory planning). The output of the system is the vector of observed values $[I, J, d]^T$. A block diagram of this system is shown in figure 3.

The Jacobian matrix used in our resolved-rate control scheme, $\mathcal{J}$, relates differential changes in the parameter vector $[I, J, d]^T$ to differential changes in the (x, y, z) coordinates of the manipulator (which are expressed with respect to the world coordinate frame). Note that we use the tool center (the point midway between the manipulator finger tips) to define the position of the end effector. This is expressed as a relationship between the error vector $[\Delta I, \Delta J, \Delta d]^T$, and the corresponding correction vector $[\Delta x_m, \Delta y_m, \Delta z_m]^T$ as follows:

$$\mathcal{J}\begin{bmatrix}\Delta x_m\\ \Delta y_m\\ \Delta z_m\end{bmatrix} = \begin{bmatrix}\Delta I\\ \Delta J\\ \Delta d\end{bmatrix} \qquad (16)$$

We now turn our attention to the first two rows of $\mathcal{J}$. We can solve eq. (7) for I_m to give:

$$I_m = \frac{x_m H_x + y_m H_y + z_m H_z - C_H}{x_m a_x + y_m a_y + z_m a_z - C_a}$$

Now, let u be any of the three coordinates x, y, z. Then for an infinitesimal change ΔI_m there is a corresponding infinitesimal change in Δu_m given by:

$$\frac{\Delta I_m}{\Delta u_m} = \frac{H_u \gamma - a_u(x_m H_x + y_m H_y + z_m H_z - C_H)}{\gamma^2}$$

where

$$\gamma = x_m a_x + y_m a_y + z_m a_z - C_a$$

After a bit of manipulation, this expression can be rewritten as:

$$\frac{\Delta I_m}{\Delta u_m} = \frac{1}{\vec{P}_m \cdot \hat{a} - C_a}\left[H_u - a_u\left(\frac{\vec{P}_m \cdot \vec{H} - C_H}{\vec{P}_m \cdot \hat{a} - C_a}\right)\right]$$

In this form, the quotient term is merely I_m, as given by eq. (8). We can substitute using this equation to obtain:

$$\frac{\Delta I_m}{\Delta u_m} = \frac{1}{\vec{P}_m \cdot \hat{a} - C_a}[H_u - a_u I_m] \qquad (17)$$

The set of equations that are obtained from eq. (17) by replacing u successively with x, y, and z and the corresponding set of equations for J, form the first two rows of the Jacobian $\mathcal{J}$. The third row of the Jacobian is obtained by considering the motion of the manipulator in the direction perpendicular to the image plane. Recall that this distance defines a plane parallel to the image plane, given by:

$$\hat{a} \cdot \vec{P}_m = d$$

By taking partial derivatives of d with respect to x, y, z we obtain:

$$\frac{\Delta d}{\Delta u} = a_u \qquad u = x, y, z$$

We may now write the equation used for resolved-rate control as:

$$\begin{bmatrix}\Delta I_m\\ \Delta J_m\\ \Delta d\end{bmatrix} = \mathcal{J}\begin{bmatrix}\Delta x_m\\ \Delta y_m\\ \Delta z_m\end{bmatrix} \qquad (18)$$

where

$$\mathcal{J} = \begin{bmatrix} \frac{H_x - a_x I_m}{\vec{P}_m \cdot \hat{a} - C_a} & \frac{H_y - a_y I_m}{\vec{P}_m \cdot \hat{a} - C_a} & \frac{H_z - a_z I_m}{\vec{P}_m \cdot \hat{a} - C_a} \\ \frac{V_x - a_x J_m}{\vec{P}_m \cdot \hat{a} - C_a} & \frac{V_y - a_y J_m}{\vec{P}_m \cdot \hat{a} - C_a} & \frac{V_z - a_z J_m}{\vec{P}_m \cdot \hat{a} - C_a} \\ a_x & a_y & a_z \end{bmatrix}$$

This system may be solved using any method to solve simultaneous equations, such as the singular value decomposition technique discussed in Section 3.

6 Experimental results

Our experimental cell consists of a PUMA 560 robot, controlled by a Sun 4/260 using RCCL. RCCL is a package of C library functions that implement real-time robot control [7]. For computer vision hardware we have used both Androx and Datacube environments.

To date, we have implemented the calibration procedure and both the open-loop and hybrid positioning of the manipulator. We have found that the accuracy of the the calibration procedure is sufficient for the open-loop positioning system.

7 Summary and conclusions

We have presented an alternative to force control. Our method relies on a set of virtual constraints that can be enforced by the use of vision sensing. The main advantage of our approach is that motion can be controlled robustly in directions that are not necessarily normal to physical constraint surfaces (unlike force control). Furthermore, the visual control techniques that are presented can be incorporated into hybrid control systems (such as the hybrid position/vision control scheme discussed above).

In the process of deriving the Jacobian matrix for our resolved-rate control system, we have obtained a set of projection equations for our camera system, and a corresponding calibration procedure. Of particular interest is the fact that these projection equations are defined in terms of a minimal set of parameters that are directly obtained from our calibration process. For example, at no point do we explicitly derive the vectors that define the camera coordinate frame, the focal length of the lens, or the vertical and horizontal sampling intervals of the camera.

Acknowledgments

This research was sponsored by the National Science Foundation under contract number IRI-9110270.

References

[1] P. Allen, B. Yoshimi, and A. Timcenko. Real-time visual servoing. In *IEEE International Conference on Robotics and Automation*, pages 851–856, April 1991.

[2] B. R. Donald. Planning multi-step error detection and recovery strategies. *International Journal of Robotics Research*, 9(1):3–60, February 1990.

[3] M. Erdmann. Using backprojections for fine motion planning with uncertainty. *International Journal of Robotics Research*, 5(1):19–45, Spring 1986.

[4] J. T. Feddema and O. R. Mitchell. Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. on Robotics and Automation*, 5(5):691–700, October 1989.

[5] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee. *Robotics: Control, Sensing, Vision, and Intelligence.* McGraw-Hill Book Co., New York, 1987.

[6] K. Furuta and M. Sampei. Path control of a three-dimensional linear motional mechanical system using laser. *IEEE Trans. on Industrial Electronics*, 35(1):52–59, February 1988.

[7] V. Hayward and R. Paul. Robot manipulator control under UNIX: RCCL, a robot control C library. *International Journal of Robotics Research*, 5(2):94–111, Winter 1986.

[8] S. A. Hutchinson. Exploiting visual constraints in robot motion planning. In *IEEE International Conference on Robotics and Automation*, Sacramento, CA, April 1991.

[9] A. C. Kak. Depth perception for robots. In S. Nof, editor, *Handbook of Industrial Robotics.* John-Wiley, NY, 1986.

[10] T. Lozano-Perez, M. T. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research*, 3(1):3–24, Spring 1984.

[11] M. T. Mason. Compliance and force control for computer controlled manipulators. In B. Brady, J. M. Hollerbach, T. L. Johnson, T. Lozano-Perez, and M. T. Mason, editors, *Robot Motion: Planning and Control*, pages 373–404. MIT Press, Cambridge, Mass., 1982.

[12] N. Papanikolopoulos, P. K. Khosla, and T. Kanade. Vision and control techniques for robotic visual tracking. In *IEEE International Conference on Robotics and Automation*, pages 857–864, April 1991.

[13] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, August 1987.

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

[15] D. E. Whitney. The mathematics of coordinated control of prosthetic arms and manipulators. *Journal of Dynamic Systems, Measurement and Control*, 122:303–309, December 1972.