Visual Compliance: Task-Directed Visual Servo Control

Andrés Castaño and Seth Hutchinson, *Member, IEEE*

***Abstract*— This paper introduces visual compliance, a new vision-based control scheme that lends itself to task-level specification of manipulation goals. Visual compliance is effected by a hybrid vision/position control structure. Specifically, the two degrees of freedom parallel to the image plane of a supervisory camera are controlled using visual feedback, and the remaining degree of freedom (perpendicular to the camera image plane) is controlled using position feedback provided by the robot joint encoders. With visual compliance, the motion of the end effector is constrained so that the tool center of the end effector maintains "contact" with a specified projection ray of the imaging system. This type of constrained motion can be exploited for grasping, parts mating, and assembly.**

We begin by deriving the projection equations for the vision system. We then derive equations used to position the manipulator prior to the execution of visual compliant motion. Following this, we derive the hybrid Jacobian matrix that is used to effect visual compliance. Experimental results are given for a number of scenarios, including grasping using visual compliance.

I. Introduction

SENSOR-BASED control is essential if robots are to perform adequately in real-world environments. This has long been recognized by the robotics community, and as a result much research has been done, both in force-based and vision-based control. However, it is not enough to merely develop arbitrary sensor-based control schemes; in order for sensor-based robotic systems to function autonomously, they must also be able to automatically create task plans that fully exploit the available sensor-based control mechanisms. This implies that task-level goals, which are specified by a human or some high level process, must be translated into goals that are specified in terms of controllable parameters.

For the specific case of force-based control, the problem of translating task-level specifications into low-level control goals has been addressed by the literature on fine-motion planning [7], [8], [23], [25]. Equipped with a set of physical laws that govern motion and friction in the configuration space, these fine-motion planners are capable of developing plans that are tolerant of uncertainties in the manipulator's position (represented by an error ball in the configuration space), its trajectory (represented by an error cone), and even in part dimensions (represented by added dimensions in the configuration space [6], [7]). The success of this approach is due in part to a control scheme that exploits physical compliance, which lends itself well to the expression of task level goals [29].

A fundamental limitation of physical compliance-based control schemes is that they can only be used to control motion in directions that are tangent to constraint surfaces in the configuration space [29]. One possible solution to this limitation is to use vision-based techniques to control motion in the remaining directions. Thus, much research attention has recently been focused on vision-based control (see, for example, [2], [3], [9], [10], [15], [16], [22], [27], [30], [31], [32], [33], [34], [35], [37]). Although vision-based control has been used successfully for a number of tasks (for example, in welding applications [1], [5], [21]), none of the systems referenced above lend themselves to task-level specification of goals, and therefore, there are currently no automatic planning systems that can exploit these control systems.

In this paper, we introduce *visual compliance* as a new vision-based control scheme that lends itself to task-level specification of goals. Visual compliance is analogous to physical compliance. With physical compliance, the robot end effector maintains contact with some physical surface during its motion. With visual compliance, the end effector maintains contact with a visual constraint surface [18]. A visual constraint surface is a virtual surface, defined by some object feature in the workspace and that feature's projection onto the image plane of a supervisory camera. Thus, visual compliant motion moves the end effector along a projection ray that passes through the focal center of a supervisory camera. In related work, we have reported a motion planning system that is capable of synthesizing uncertainty-tolerant motion plans that exploit visual compliance [12], [13]. Here we develop the control structure necessary to effect visual compliant motion.

Visual compliance is achieved by a hybrid vision/position control structure. The particular scheme that we use derives from resolved-rate position control [26], [38]. In general, resolved-rate position control is accomplished by using a Jacobian matrix to relate differential changes in the task space to differential changes in the joint space of the robot. For visual compliance, we use a hybrid Jacobian, $\mathcal{J}_{\mathrm{vc}}$. The first two rows of $\mathcal{J}_{\mathrm{vc}}$ relate differential changes in the robot's motion to differential changes in the image that is observed by a camera (as in [10], [37]). The third row of $\mathcal{J}_{\mathrm{vc}}$ relates differential changes in the robot's motion to differential changes in the perpendicular distance between the robot end effector and the camera image plane. Thus, using $\mathcal{J}_{\mathrm{vc}}$, it is possible to achieve

Manuscript received January 8, 1993; revised June 7, 1993. This paper was presented at the International Conference on Robotics and Automation, Nice, France, May 12–14, 1992. This work was supported in part by NSF grants NSF-IRI 9110270 and NSF-IRI 92-16428.

The authors are with The Beckman Institute for Advanced Science and Technology and the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA.

IEEE Log Number 9400179.

1042–296X/94$04.00 © 1994 IEEE

motion that "complies" to a specified projection ray through the camera focal center, moving either toward or away from the camera while keeping the tool center aligned with the projection ray. As described in [11], [12], [13], [18], this type of motion can be exploited for grasping, parts mating, assembly, and other types of robotic manipulation.

The remainder of the paper is organized as follows. In Section II, we derive the projection equations that define the imaging geometry of the camera. In Section III, we derive equations that can be used to position the robot manipulator at a specified perpendicular distance from the camera focal center such that the tool center of the manipulator 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. In Section IV, we derive $\mathcal{J}_{\mathrm{vc}}$, the Jacobian matrix that is used to effect hybrid vision/position control of manipulator motion. In Section V, we present results obtained using an implemented robotic system. Section VI provides a discussion of several related issues, including how our visual servo control system fits into the broader context of autonomous task planning. Finally, in Section VII we summarize the contributions of the work to date.

II. Projection Equations

In order to perform visual servo control, the relationships between the robot's workspace and the camera image plane must be known. In general, these relationships are defined in terms of a set of projection equations that define how points in the workspace project onto the camera image plane via the imaging geometry of the camera (see for example [20], [28], [36]). In this section, we derive the projection equations for the robotic system shown in Fig. 1. Our derivations closely follow those given in [20].

The projection of world points onto the camera image plane can be viewed as a transformation between the world coordinate frame and the camera image plane coordinate frame. For the *world coordinate frame* we use the base frame for the PUMA 560 robot (see for example [14]). The *image plane coordinate frame* is defined by the four vectors $\vec{C}, \hat{h}, \hat{v}$, and $\hat{a}$, 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.

The camera image plane is actually a truncated plane in the robot's workspace, which can be specified by the parametric equation

$$\begin{bmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{bmatrix} = \vec{C} - f\hat{a} + u\hat{h} + v\hat{v}$$

where f is the focal length of the camera. Thus, we may treat (u,v) as coordinates of a point on the image plane expressed in the image plane coordinate frame.

From Fig. 1, we observe the following relationships between the world frame coordinates of a point $\vec{P}$, and the (u,v) coordinates of its projection

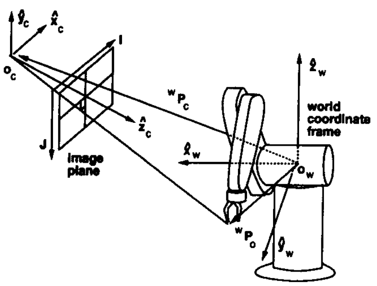


Fig. 1. World and image plane coordinate systems.

$$\frac{u}{f} = \frac{(\vec{P} - \vec{C}) \cdot \hat{h}}{(\vec{P} - \vec{C}) \cdot \hat{a}}, \quad \frac{v}{f} = \frac{(\vec{P} - \vec{C}) \cdot \hat{v}}{(\vec{P} - \vec{C}) \cdot \hat{a}}.$$

Using standard video hardware (e.g., digitizers, frame grabbers), we will not have direct access to the (u,v) coordinates of an image plane point. Rather, we will have access to the discrete coordinates that represent the horizontal and vertical indices into the discrete image array. We use (I,J) to denote these coordinates. Let Δu and Δv be the horizontal and vertical sampling intervals. Then the following relations hold

$$u = (I - I_0)\Delta u, \quad v = -(J - J_0)\Delta v$$

where I_0 and J_0 are the coordinates of the center of the image plane that correspond to the origin of the image plane.

After some manipulations, we obtain the following projection equations

$$I = \frac{\vec{P} \cdot \vec{H} - C_H}{\vec{P} \cdot \hat{a} - C_a}, \quad J = \frac{\vec{P} \cdot \vec{V} - C_V}{\vec{P} \cdot \hat{a} - C_a}, \tag{1}$$

where

$$C_a = \vec{C} \cdot \hat{a}, \quad C_H = \vec{C} \cdot \vec{H}, \quad C_V = \vec{C} \cdot \vec{V}$$

and

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

Note that $\vec{H}$ and $\vec{V}$ are *not* unit vectors in the horizontal and vertical directions of the image plane. Rather, they are vectors that represent composite information regarding the horizontal and vertical directions of the image plane, the horizontal and vertical 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 $-\hat{a}$). A calibration procedure to derive the relevant system parameters is described in [4].

III. Open-Loop Positioning

Before performing visual compliant motion, the end effector of the manipulator must be brought into contact with the specified visual constraint surface. This amounts to positioning the end effector so that the tool center, represented by $\vec{P}$, intersects a specified projection ray. Stated another way, given input (I, J), compute the (x, y, z) workspace coordinates for the point $\vec{P}$. The immediate problem that we face is that the projection equations given in (1) 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, y, z), 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, y, z) 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. Rewriting (1), we obtain

$$(H_x - Ia_x)x + (H_y - Ia_y)y + (H_z - Ia_z)z = C_H - IC_a, \tag{2}$$

$$(V_x - Ja_x)x + (V_y - Ja_y)y + (V_z - Ja_z)z = C_V - JC_a. \tag{3}$$

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

$$\hat{a} \cdot \vec{P} = d, \tag{4}$$

where $d = d_I + \hat{a} \cdot \vec{C}$, and $\hat{a}$, $\vec{P}$, and $\vec{C}$ are as defined in Section II (see Fig. 1 for a graphical illustration).

Using (2), (3) and (4), a system that determines the intersection between a projection ray from the camera and the desired plane in the workspace coordinate frame is established. In matrix form

$$\begin{bmatrix} H_x - Ia_x & H_y - Ia_y & H_z - Ia_z \\ V_x - Ja_x & V_y - Ja_y & V_z - Ja_z \\ a_x & a_y & a_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} C_H - IC_a \\ C_V - JC_a \\ d \end{bmatrix}.$$

Although this type of open-loop positioning is useful for initially positioning the manipulator near a target projection ray, open-loop control rarely succeeds in precisely placing the manipulator to achieve 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 (resulting from noise in the imaging process); and errors in the camera modeling (since we use a simple pin-hole approximation to the camera in the derivation of the projection equations). This does not adversely affect system performance, since closed-loop control (which is described in the next section) is used to effect the visual compliant motion, once the end effector is near the specified projection ray.

IV. Visual Compliance

As described above, when performing visual compliant motion, vision feedback is used only to control motion in directions that lie 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). Therefore, to execute visual compliant motion, we use a hybrid control approach. Specifically, we use a resolved-rate motion control approach [26], [38] 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, $\mathcal{J}_{\text{vc}}$.

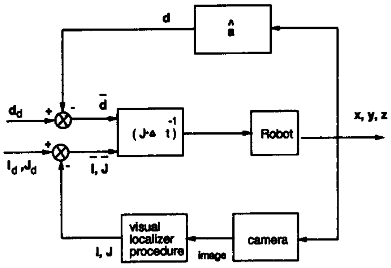


Fig. 2. Hybrid-vision/position-control block diagram.

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 $[I_d, J_d, d_d]^T$, which would be determined by a trajectory planner (see [10] 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 Fig. 2.

The Jacobian matrix used in our resolved-rate control scheme, $\mathcal{J}_{\text{vc}}$, 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 relationship is given by

$$\begin{bmatrix} \dot{I} \\ \dot{J} \\ \dot{d} \end{bmatrix} = \mathcal{J}_{vc} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}.$$

We now turn our attention to the first row of $\mathcal{J}_{\text{vc}}$. We can expand (1) for I to obtain

$$I = \frac{xH_x + yH_y + zH_z - C_H}{xa_x + ya_y + za_z - C_a}.$$

The partial derivative of I with respect to x is given by

$$\frac{\partial I}{\partial x} = \frac{H_x(xa_x + ya_y + za_z - C_a) - a_x(xH_x + yH_y + zH_z - C_H)}{(xa_x + ya_y + za_z - C_a)^2}.$$

This expression can be rewritten as

$$\frac{\partial I}{\partial x} = \frac{1}{\vec{P} \cdot \hat{a} - C_a} \left[H_x - a_x \left(\frac{\vec{P} \cdot \vec{H} - C_H}{\vec{P} \cdot \hat{a} - C_a} \right) \right].$$

In this form, the quotient term at the right is simply the projection equation for I in (1) that was derived in Section II. We can substitute using this equation to obtain

$$\frac{\partial I}{\partial x} = \frac{1}{\vec{P} \cdot \hat{a} - C_a}(H_x - a_x I).$$

Similar manipulations can be performed for y and z, and for the second row of the Jacobian, which corresponds to the J coordinate.

The third row of the Jacobian is obtained by considering the motion of the manipulator in the direction perpendicular to the image plane. By taking partial derivatives of (4) with respect to x, y, z we obtain

$$\frac{\partial d}{\partial x} = a_x, \quad \frac{\partial d}{\partial y} = a_y, \quad \frac{\partial d}{\partial z} = a_z.$$

We may now write the Jacobian as

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

The discrete-time state space formulation of this system is given by

$$\begin{bmatrix} I(k+1) \\ J(k+1) \\ d(k+1) \end{bmatrix} = \begin{bmatrix} I(k) \\ J(k) \\ d(k) \end{bmatrix} + \mathcal{J}_{\mathrm{vc}} \Delta t u(k).$$

Assuming that the sampling time Δt is small, an appropriate discrete-time control law is given by

$$u(k) = (\mathcal{J}_{vc} \Delta t)^{-1} e(k)$$

where the error is defined as

$$e(k) = \begin{bmatrix} I_d(k) \\ J_d(k) \\ d_d(k) \end{bmatrix} - \begin{bmatrix} I(k) \\ J(k) \\ d(k) \end{bmatrix}.$$

This result is similar to that given in [16].

V. Experimental Results

In this section we present several experimental results. Our experimental system consists of a Puma 560 robot, controlled by a Sun 4/260 using RCCL [17], [24]. The vision system consists of Datacube hardware and a Sun 3. The vision system determines the (I, J) image coordinates of the tool center, and sends these coordinates to the Sun 4/260 via an ethernet connection. This introduces a delay between the vision and control system of Δt (approximately 0.6 seconds). In spite of this delay and the relatively slow sampling rate of the vision system, we have been able to achieve good system performance, as can be seen from the results presented in this section. In order to simplify the determination of the tool center image coordinates, a small LED is attached to the robot end effector.

A differential change in the image of the end-effector does not necessarily imply a differential change in the motion of the end-effector. Therefore, it is possible that for small values of $e(k)$, large values for the control, $u(k)$, may result. For this reason, when the control input $u(k)$ is large, we scale its magnitude. This eliminates adverse transient effects associated with large step inputs.

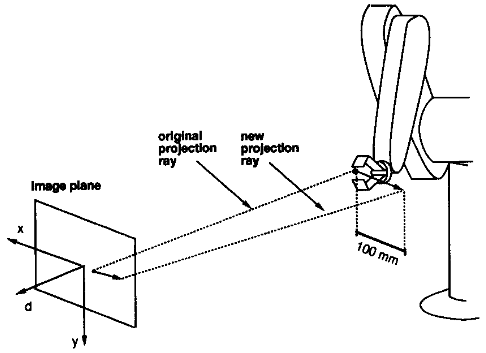


Fig. 3. Motion to a new projection ray.

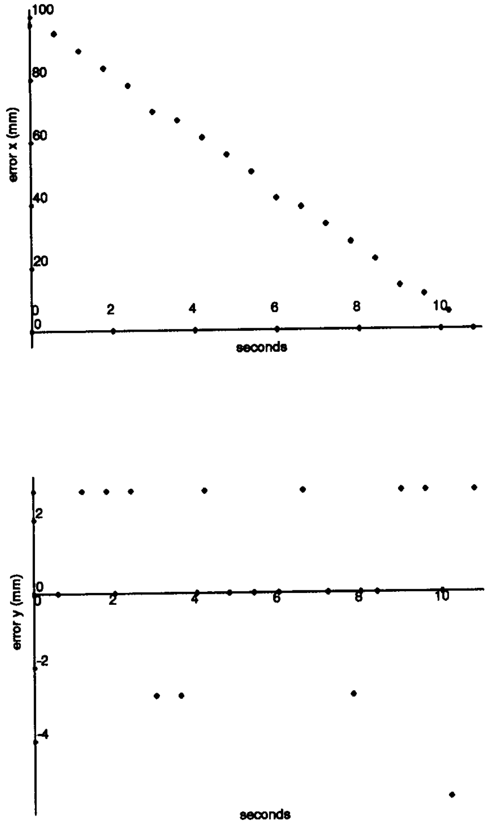


Fig. 4. Error behavior for the motion illustrated in Fig. 3.

A. Positioning the Tool Center on a Specified Projection Ray

Our first set of experiments involves positioning the tool center of the end effector along a specified projection ray, at a specified distance from the camera image plane. Thus, the goal position in 3-space is defined by the intersection of the desired projection ray and a plane parallel to the image plane at the specified distance from the image plane. Visual feedback is used to control the position of the tool in the directions parallel to the image plane and position feedback is used to control the tool in the direction perpendicular to the image plane.

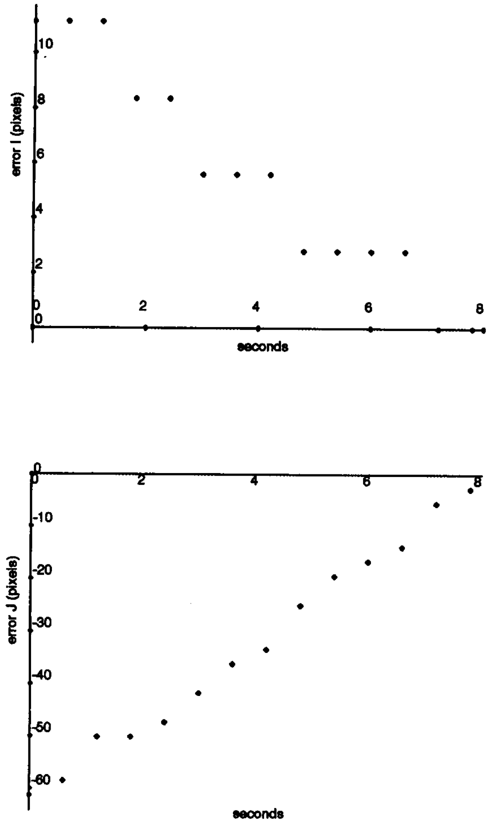


Fig. 5. Error in the I and J direction for initial positioning.

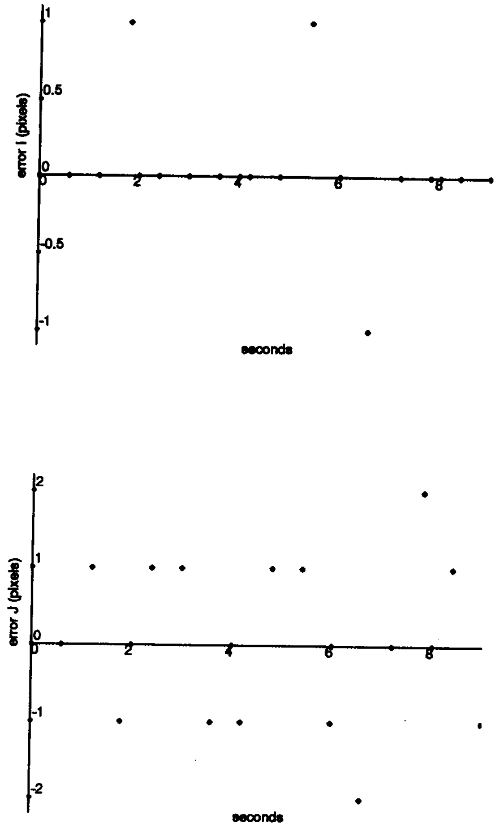


Fig. 7. Error behavior for the motion illustrated in Fig. 6.

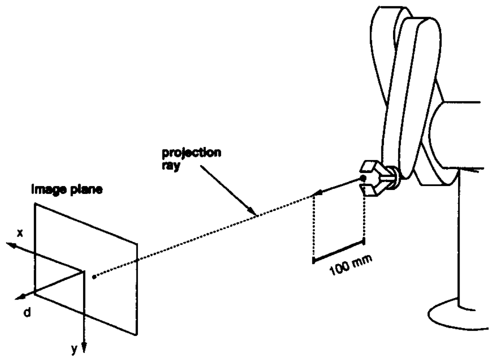


Fig. 6. Visual compliance along a specified projection ray, with a perpendicular distance of 100 mm.

Figs. 3 and 4 illustrate the z, y, d errors with respect to the robot coordinate frame. In this example, the robot is positioned in contact with a certain projection ray, and is commanded to move to a second projection ray in such a way that its end effector will move 100 mm in the x direction of the image plane while maintaining a constant distance to the camera. The error in the x direction is reduced to less that 1 cm in 11 seconds. During this time, the error on the y direction is kept smaller than 1 cm, and the error on the d direction is kept very near zero. The main reason for the disparity in the magnitude of the errors is the difference in time delay for the visual and position loops. An error in the d direction can be corrected by the trajectory planner using the joint encoders (every 0.875 ms), while and error in the x, y directions can only be corrected when a full image is taken and processed (every 0.6 seconds).

Fig. 5 shows the errors in positioning in I and J as a function of time. In this example, the manipulator is moved from a position very near the target projection ray until contact with the projection ray is made. It should be noted that the relationship between errors in pixels (measured in the image plane) and errors in mm (measured in the robot coordinate frame) are related by the projection equations. Of particular importance in this relationship is the distance from the robot to the camera, which was approximately three meters in our experiments.

B. Visual Compliance

Our second set of experiments involves performing visual compliance along a specified projection ray. We show the errors when the robot is moving toward the camera. Fig. 6 illustrates a commanded motion along a projection ray, and Fig. 7 shows the errors in both the normal and tangent

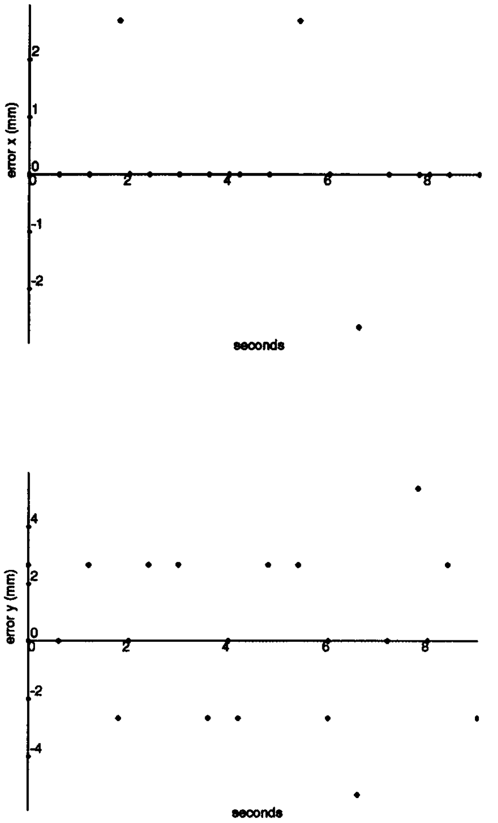


Fig. 8. Error in I, J for visual compliant motion.

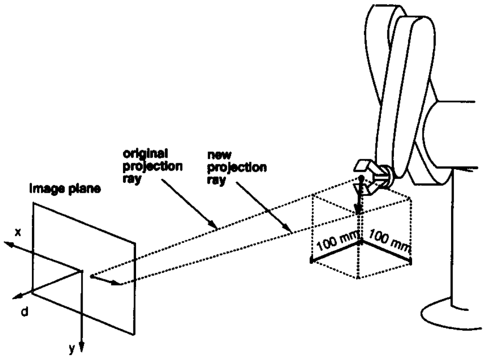


Fig. 9. Motion to a new projection ray, with a perpendicular distance of 100 mm.

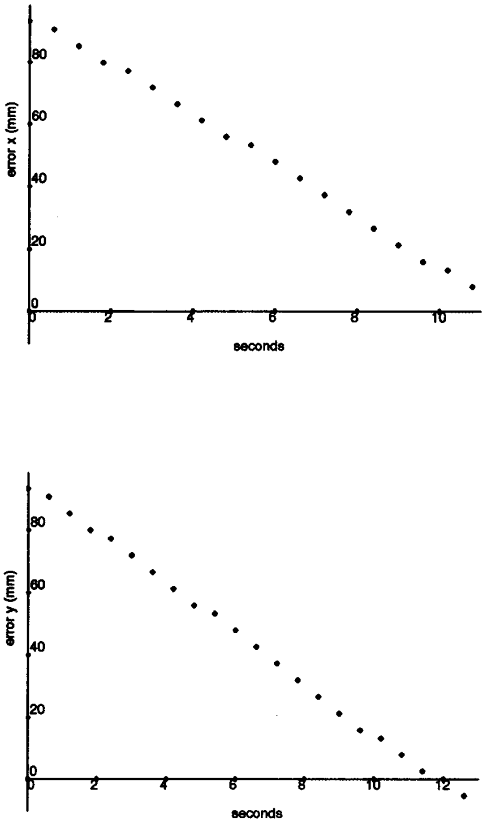


Fig. 10. Error behavior for the motion illustrated in Fig. 9.

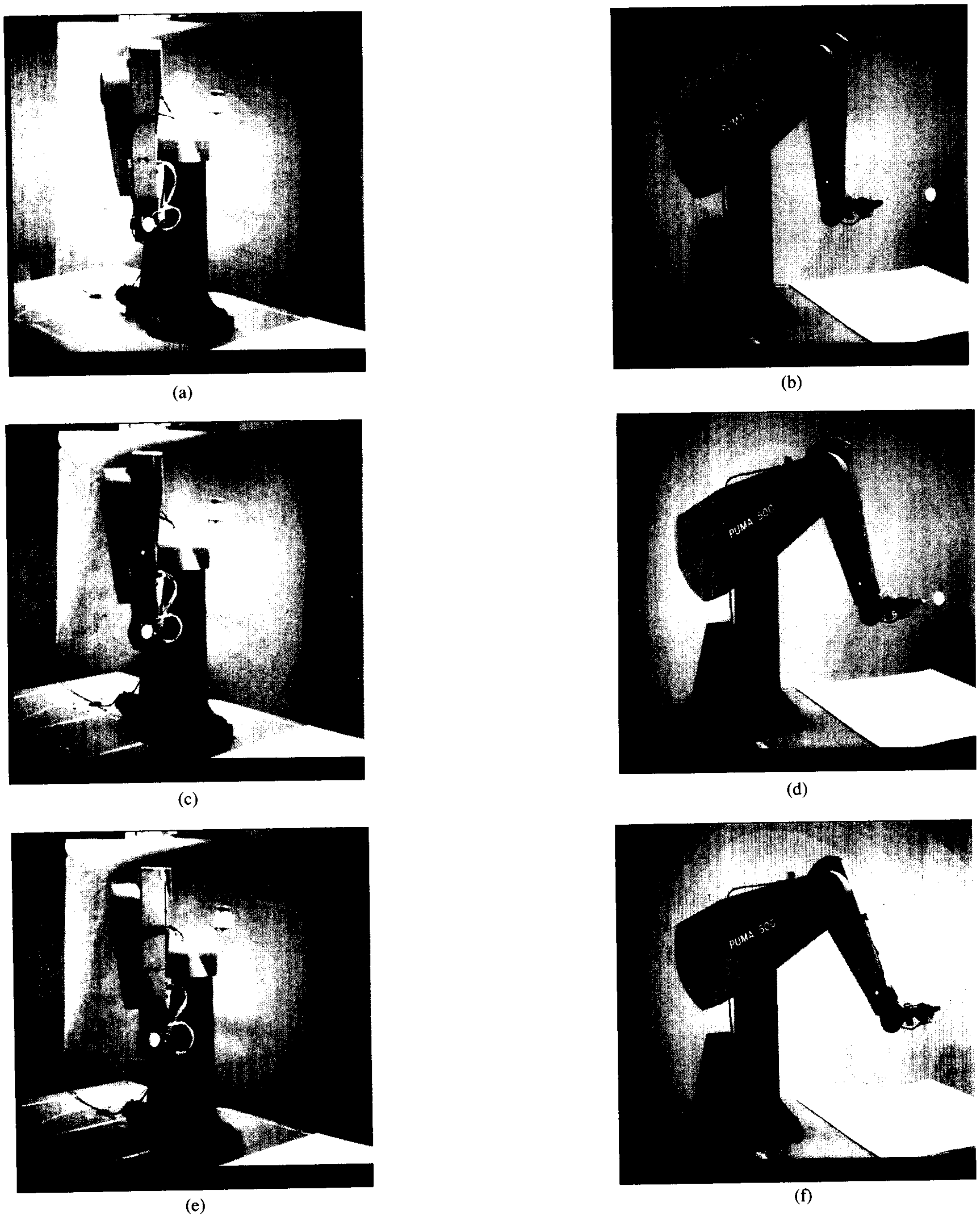
directions with respect to the world coordinate frame. In this example, the manipulator is commanded to move forward 100 mm toward the camera using visual compliance. The final desired position is achieved in nine seconds. The final error in all directions is smaller than one centimeter. Fig. 8 shows the errors in I and J as a function of time, for visual compliance toward the camera.

C. Composite Motion

It is possible to extend the visual compliance paradigm, so that the manipulator moves to a specified projection ray at the same time that it is moving in a direction perpendicular to the camera. This is illustrated in Figs. 9 and 10, where the commanded motion is toward the camera 100mm, and parallel to the image plane in the direction of the camera x axis.

D. A Simple Grasping Task

Fig. 11 shows our final experiment, in which visual compliance was used to grasp a Ping-Pong ball suspended in the robot workspace. In this experiment, the (x, y, z) position of the ball was not calculated. Rather, the task goal was specified in terms of visual compliant motions. Specifically, to execute the grasp, the robot moves its end effector so that one finger of the robot stays in contact with a projection ray that intersects the occluding contour of the ball. The three frames on the left show the robot as it is seen by the supervisory camera. Note that in all three frames the end effector appears to be in the same position. The three frames on the right show the robot from a side view (the three frames on the right were taken to correspond to the three frames on the left). From this figure, it can be seen that visual compliance is effected by regulating the position of the end effector in the image, while simultaneously moving the end effector in a direction toward



(a) (b) (c) (d) (e) (f)

Fig. 11. Puma 560 performing a grasp using visual compliance.

the camera. For these experiments, infrared sensors mounted on the robot fingertips were used to determine when the ball was within the robot's grasp.

VI. Discussion

The system that we have presented facilitates visual compliance, which is analogous to physical compliance effected through the use of force/torque sensing. The analogy can be extended to the planning system. In related work, we have developed a planning system that extends the backprojection planning formalism [7], [8], [23], [25] by allowing visual constraint surfaces to be included in the boundary of the backprojection [12], [13]. Our planning system requires as input a geometric description of the environment and of the task (in terms of a goal region in the robot configuration space). Thus, given a description of the task, our planning system will derive a motion plan that exploits the visual compliance capabilities described in the present paper.

Visual servo control, as described in this paper, requires that a set of features on the manipulator be constantly visible to the supervisory camera. This is a planning issue that we have addressed elsewhere [19]. However, we note that by construction visual constraint surfaces do not intersect obstacles (since in such a case the feature that generates the surface would be occluded from the camera's view). Therefore, occlusion is generally not a problem that affects visual compliance.

Finally, the resolved-rate scheme described in this paper could be improved by incorporating a predictive component in the visual tracking system. This could be done in a fairly straightforward manner, since the robot dynamics and imaging system parameters are fully known. Such improvements to the tracking system are the subject of ongoing research.

VII. Conclusions

We have introduced visual compliance as an alternative to physical compliance using force control. Our method relies on a set of virtual constraints that can be enforced by the use of vision sensing. The main advantages of our approach are that (1) visual compliance lends itself well to task-level specification of manipulation goals, and (2) motion can be controlled in directions that are not necessarily normal to physical constraint surfaces (unlike force control). In related work, we have developed a task planner that directly exploits the existence of this control system for the synthesis of uncertainty-tolerant motion plans [12], [13].

References

[1] J. E. Agapakis, J. M. Katz, J. M. Friedman, and G. N. Epstein, "Vision-aided robotic welding: An approach and a flexible implementation," *Int. J. Robotics Research*, vol. 9, no. 5, pp. 17–33, October 1990.

[2] P. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Trajectory filtering and prediction for automated tracking and grasping of a moving object," in *Proc. IEEE Int'l Conference on Robotics and Automation*, pp. 1850–1856, May 1992.

[3] P. Allen, B. Yoshimi, and A. Timcenko, "Real-time visual servoing," in *Proc. IEEE Int'l Conference on Robotics and Automation*, pp. 851–856, April 1991.

[4] A. Castano and S. A. Hutchinson, "Visual compliance: Task-directed visual servo control," AI Tech. Report UIUC-BI-AI-RCV-93-01 (Robotics/Computer Vision Series), University of Illinois at Urbana-Champaign, 1993.

[5] W. F. Clocksin, J. S. E. Bromley, P. G. Davey, A. R. Vidler, and C. G. Morgan, "An implementation of model-based visual feedback for robot arc welding of thin sheet steel," *Int. J. Robotics Research*, vol. 4, no. 1, pp. 13–26, 1985.

[6] B. R. Donald, "A geometric approach to error detection and recovery for robot motion planning with uncertainty," *Artificial Intelligence*, vol. 37, no. 1–3, pp. 223–271, 1988.

[7] B. R. Donald, "Planning multi-step error detection and recovery strategies," *Int. J. Robotics Research*, vol. 9, no. 1, pp. 3–60, 1990.

[8] M. Erdmann, "Using backprojections for fine motion planning with uncertainty," *Int. J. Robotics Research*, vol. 5, no. 1, pp. 19–45, Spring 1986.

[9] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robotics Automat.*, vol. 8, no. 3, pp. 313–326, 1992.

[10] J. T. Feddema and O. R. Mitchell, "Vision-guided servoing with feature-based trajectory generation," *IEEE Trans. Robotics Automat.*, vol. 5, no. 5, pp. 691–700, October 1989.

[11] A. Fox, A. Castano, and S. A. Hutchinson, "Planning and executing visually constrained robot motions," in *Proceedings SPIE Symposium on Advances in Intelligent Robotic Systems*, Boston MA, 1991.

[12] A. Fox and S. Hutchinson, "Exploiting visual constraints in the synthesis of uncertainty-tolerant motion plans I: The directional backprojection," in *Proc. IEEE Int'l Conference on Robotics and Automation*, pp. 305–310, 1993.

[13] A. Fox and S. Hutchinson, "Exploiting visual constraints in the synthesis of uncertainty-tolerant motion plans II: The nondirectional backprojection," in *Proc. IEEE Int'l Conference on Robotics and Automation*, pp. 311–316, 1993.

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

[15] K. Furuta and M. Sampei, "Path control of a three-dimensional linear motional mechanical system using laser," *IEEE Trans. Ind. Electron.* vol. 35, no. 1, pp. 52–59, February 1988.

[16] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kinura, "Manipulator control with image-based visual servo," in *Proc. IEEE Int'l Conference on Robotics and Automation*, pp. 2267–2272, Sacramento, CA, April 1991.

[17] V. Hayward and R. Paul, "Robot manipulator control under UNIX: RCCL, a robot control C library," *International Journal of Robotics Research*, vol. 5, no. 2, pp. 94–111, Winter 1986.

[18] S. A. Hutchinson, "Exploiting visual constraints in robot motion planning," in *Proc. IEEE Int'l Conference on Robotics and Automation*, Sacramento, CA, April 1991.

[19] S. A. Hutchinson, "Planning visually controlled motions," in *Proceedings of the AAAI Fall Symposium on Sensory Aspects of Robotic Intelligence*, pp. 38–43, 1991.

[20] A. C. Kak, "Depth perception for robots," in *Handbook of Industrial Robotics*, S. Nof, Ed. New York: John-Wiley, 1986.

[21] P. K. Khosla, C. P. Neuman, and F. B. Prinz, "An algorithm for seam tracking applications," *International Journal of Robotics Research*, vol. 4, no. 1, pp. 27–41, 1985.

[22] A. J. Koivo and N. Houshangi, "Real-time vision feedback for servoing robotic manipulator with self-tuning controller," *IEEE Trans. Syst., Man and Cyber.*, vol. 21, no. 1, pp. 134–142, February 1991.

[23] J. C. Latombe, A. Lazanas, and S. Shekhar, "Robot motion planning with uncertainty in control and sensing." *Artificial Intelligence*, vol. 52, pp. 1–47, 1991.

[24] John Lloyd, "Multi-RCCL Reference Manual, Release 4.0," McGill Research Centre for Intelligent Machines, McGill University, Montreal, Quebec, Canada, 1988.

[25] T. Lozano-Perez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *Int. J. Robotics Research*, vol. 3, no. 1, pp. 3–24, Spring 1984.

[26] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "Resolved-acceleration control of mechanical manipulators." *IEEE Trans. Automat. Contr.*, vol. 25, no. 3, pp. 468–474, 1980.

[27] T. Martinez, H. Ritter, and K. Schulten, "Three-dimensional neural-net for learning visuo-motor coordination of a robot arm," *IEEE Trans. on Neural Networks*, vol. 1, pp. 131–136, 1990.

[28] H. A. Martins, J. R. Birk, and R. B. Kelly, "Camera models based on data from two calibration planes," *Computer Vision, Graphics and Image Processing*, vol. 17, pp. 173–180, 1981.

[29] M. T. Mason, "Compliance and force control for computer controlled manipulators," in *Robot Motion: Planning and Control*, B. Brady, J. M. Hollerbach, T. L. Johnson, T. Lozano-Perez, and M. T. Mason, Eds. Cambridge, MA: MIT Press, 1982, pp. 373–404.

[30] B. W. Mel, *Connectionist Robot Motion Planning: A Neurally-Inspired Approach to Visually-Guided Reaching*. Boston, MA: Academic Press, 1990.

[31] W. T. Miller, "Sensor-based control of robotic manipulators using a general learning algorithm," *IEEE Journal of Robotics and Automation*, vol. 3, no. 2, pp. 157–165, 1987.

[32] N. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Vision and control techniques for robotic visual tracking," in *Proc. IEEE Int'l Conference on Robotics and Automation*, vol. pp. 857–864, 1991.

[33] R. Sharma, J.-Y. Herve, and P. Cucka, "Dynamic robot manipulation using visual tracking," in *Proc. IEEE Int'l Conference on Robotics and Automation*, vol. pp. 1844–1849, 1992.

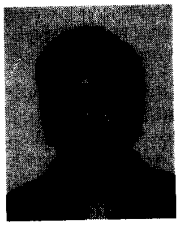
[34] S. B. Skaar, W. H. Brockman, and R. Hanson, "Camera-space manipulation," *International Journal of Robotics Research*, vol. 6, no. 4, pp. 20–32, 1987.

[35] S. B. Skaar, W. H. Brockman, and W. S. Jang, "Three-dimensional camera-space manipulation," *International Journal of Robotics Research*, vol. 9, no. 4, pp. 22–39, 1990.

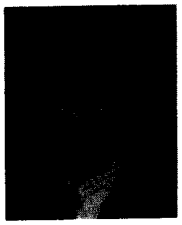
[36] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses." *IEEE Trans. Robotics Automat.*, vol. 3, no. 4, pp. 323–344, 1987.

[37] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE Trans. Robotics and Automat.*, vol. 3, no. 5, pp. 404–417, 1987.

[38] D. E. Whitney, "The mathematics of coordinated control of prosthetic arms and manipulators." *J. Dynamic Systems, Measurement and Control*, vol. 122, pp. 303–309, 1972.



Andrés Castaño received the B.S. and Master degrees from Universidad de los Andes, Colombia, and the M.S. in Electrical Engineering from the University of Illinois at Urbana-Champaign, where he is currently a Ph.D. candidate and a research assistant at the Beckman Institute for Advanced Science and Technology. His current research interests are robotics applied to teleoperation and computer vision.



Seth Hutchinson (S'85–M'88) received the Ph.D. degree from Purdue University in West Lafayette, Indiana in 1988. He spent 1989 as a Visiting Assistant Professor of Electrical Engineering at Purdue University. In 1990 Dr. Hutchinson joined the faculty at the University of Illinois in Urbana-Champaign, where he is currently an Assistant Professor in the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Beckman Institute for Advanced Science and Technology.