A Modular System for Robust Positioning Using Feedback from Stereo Vision

Gregory D. Hager

Research Report YALEU/DCS/RR-1074
May 1995

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

A Modular System for Robust Positioning Using Feedback from Stereo Vision

Gregory D. Hager
Department of Computer Science
Yale University
New Haven, CT 06520-8285

Phone: 203 432-6432
Email: hager@cs.yale.edu

May 4, 1995

Abstract

This article introduces a modular framework for robot motion control using stereo vision. The approach is based on a small number of generic motion control operations referred to as *primitive skills.* Each primitive skill uses visual feedback to enforce a specific task-space kinematic constraint between a robot end-effector and a set of target features. By observing both the end-effector and target features, primitive skills are able to position with an accuracy that is independent of errors in hand-eye calibration. Furthermore, primitive skills are easily combined to form more complex kinematic constraints as required by different applications.

These control laws have been integrated into a system that performs tracking and control on a single processor at real-time rates. Experiments with this system have shown that it is extremely accurate, and that it is insensitive to camera calibration error. The system has been applied to a number of example problems, showing that modular, high precision, vision-based motion control is easily achieved with off-the-shelf hardware.

Submitted to the IEEE Transactions on Robotics and Automation.

1 Introduction

Over the last several years, a great deal of research has been devoted to using vision to guide robotic systems. Despite these efforts, vision-based robotic systems are still the exception rather than the rule. One major roadblock to progress has been the processing of visual data. In order to provide feedback at servo rates, most systems rely on task-specific image processing algorithms, often combined with specialized hardware. This is costly in terms of both time and money as it forces the designer to "reinvent" the vision component for each application. Another difficulty arises from the fact that the accuracy of many visual servoing systems is extremely sensitive to camera or robot miscalibration. Although calibration is a well-understood problem, computing a highly accurate calibration can be extremely laborious. There are also finite limits to how well any mechanical system can be calibrated. Finally, little work has been done to make vision-based motion control modular, reconfigurable, and usable by "non-experts."

We are developing an approach to visual servoing that emphasizes modularity, simplicity, and accuracy. In this approach, vision-based control systems are constructed by combining a set of motion control and visual tracking operations subsequently referred to as *hand-eye skills*. The hand-eye skills for performing a specific task are developed out of a smaller set of building blocks referred to as *primitive skills*. The goal of the skill-based paradigm is to demonstrate that by developing a small repertoire of modular primitive skills and a reasonable set of "composition" operations, a large variety of tasks can be solved in an intuitive, modular and robust fashion.

This article describes the basis for a set of primitive skills which are designed to enforce kinematic constraints using generic visual inputs. This inputs include the projections of edges, vertices or lines in an image. The algorithms use error metrics defined by observing *both* the position of the manipulator and its stationing point. These errors are computed simultaneously in two camera images, making it possible to perform full three-dimensional positioning operations. In addition, these primitive skills are constructed so that positioning accuracy is in fact *independent* of errors in the hand-eye calibration of the system.

One of the cited disadvantages of many previous approaches to vision-based motion control is the fact that motions are defined implicitly in a non-intuitive projective space. In

contrast, each primitive skill described here maintains a kinematic constraint or performs a motion that can be explicitly defined *in the robot task space.* In this way, it is possible to program or plan an operation in terms of the geometry of the robot task space, and to automatically translate the plan into a robust vision-based motion control operation.

Skill-based systems are inexpensive and simple to implement. The visual information needed to instantiate hand-eye skills is extremely local: the location of features such as corners and edges in one or more images. The image processing needed to extract these features is straightforward, and can easily be performed on standard workstations or PC's [13]. The interface to the robot hardware is also extremely simple: a one-way stream of velocity or position commands expressed in robot base coordinates. This enhances portability and modularity, making it simple to retro-fit an existing system with visual control capabilities. It also makes it simple to superimpose task-space motion or force control operations to produce hybrid control systems.

The remainder of this paper discusses these points in more detail and presents experimental results from an implemented system. The next section discusses some of the relevant visual servoing literature. Section 3 defines the vision-based positioning framework that forms the basis of the skill-based approach. Section 4 describes three primitive skills and illustrates their application to three example problems. In addition, the sensitivity of these algorithms to calibration error is examined. Section 5 describes an implemented system and presents several experiments devoted to determine the accuracy and stability of the servoing algorithms. The final section describes work currently in progress.

2 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 [6] for an extensive review and [16, 19] for a recent collection of articles). Systems can be categorized according to several properties as discussed below.

The first criteria is whether visual feedback is directly converted to joint torques (referred to as direct visual servo), or whether internal encoder feedback is used to implement a

velocity servo in a hierarchical control arrangement (referred to as look-and-move) [33]. One advantage of employing a look-and-move arrangement is that the robot appears as a perfect integrator of joint or Cartesian velocity inputs to the visual control system. In practice, practically all implemented systems are of the look-and-move variety as are all systems described in this article.

The second criteria is the number of cameras and their kinematic relationship to the end-effector. A majority of the recently constructed visual servoing systems employ a single camera, typically mounted on the arm itself *e.g.* [3, 7, 9, 18, 28, 29, 35]. A single camera minimizes the visual processing needed to perform visual servoing, however the loss of depth information complicates the control design as well as limiting the types of positioning operations than can be implemented. Two cameras in a stereo arrangement can be used to provide complete three-dimensional information about the environment [1, 2, 5, 20, 23, 21, 25, 30]. Stereo-based motion control systems have been implemented using both free-standing and arm-mounted cameras, although the former arrangement is more common. This article discusses a free-standing stereo camera arrangement, although with minor modifications the same formulation could be used for an end-effector mounted stereo camera system.

A third major distinction is between systems that are *position-based* versus those that are *image-based.* The former define servoing error in a Cartesian reference frame (using vision-based pose estimation) while the latter compute feedback directly from errors measured in the camera image. Most stereo systems are position-based, while monocular systems tend to be image-based (for an exception, see [35]). Arguments have been made for both types of systems. In particular, position-based systems are often characterized as more "natural" to program since they inherently operate in the robot task space, whereas image-based systems operate in a less intuitive projection of the task space. Image-based systems are typically less sensitive to errors in the camera calibration than position-based systems, but they introduce nonlinearities into the control problem and hence have proven problematic to analyze theoretically. This article employs image-based methods to develop primitive positioning skills. However, these skills are chosen so that they are directly related to task-space kinematic constraints, thereby combining the positive attributes of both image-based and position-based methods.

Most visual control systems only observe the features used to define the stationing point or trajectory for the manipulator. In this article, these systems will be referred to as "endpoint-open-loop" (EOL) systems since the control error does not involve actual observation of the robot end-effector. In particular, for position-based systems such as the stereo systems mentioned above, the fact that they are EOL means that the positioning accuracy of the system is limited by the accuracy of stereo reconstruction and the accuracy of the hand-eye calibration.

A system that observes *both* the manipulator and the target will be referred to as an "endpoint-closed-loop" (ECL) system. Few ECL systems have been reported in the literature. Wijesoma *et.al.* [34] describe an ECL monocular hand-eye system for planar positioning using image feedback. An ECL solution to the problem of three DOF positioning using stereo is described in [14]. A six DOF ECL servoing system employing stereo vision is described [21]. The latter employs an affine approximation to the perspective transformation to reconstruct the position and orientation of planes on an object and on a robot manipulator. Reconstructed pose forms the basis of a position-based servo algorithm for aligning and positioning the gripper relative to the object. The affine approximation leads to a linear estimation and control problem, however it also means that the system calibration is only locally valid. A similar image-based system appears in [5, 31] with the difference that an attempt is made to modify the approximate linear model online. This article describes an image-based ECL system that uses a globally valid perspective model.

Most visual servoing research has concentrated on developing solutions to isolated problems. The problem of developing a general framework for synthesizing task-specific visual servoing algorithms has not been been widely discussed. An exception is found in [4] where it is noted that it would be possible to compile a "library" of canonical visual tasks. However, the types of positioning operations that are possible are limited by the use of a single end-effector mounted camera. In many cases, the proposed operations require metric information on the structure of the observed feature or object. This restricts their use to structured environments as well as making their accuracy sensitive to calibration errors. In contrast, the use of stereo vision in this article makes it possible to develop positioning primitives which do not require metric information about observed features, and which are insensitive

to calibration errors. Since they do not employ metric information, the primitives are easily applied in unstructured as well as structured environments.

3 Background and Problem Setting

This section establishes notational conventions and provides general background for the remainder of the article. The first part describes a general framework for vision-based control of position and points out several important properties of the approach. The second part reviews results related to the projection and reconstruction of points and lines from stereo images.

3.1 A Framework for Vision-Based Control of Position

Unless otherwise noted, all positions, orientations and feature coordinates are expressed relative to the robot base coordinate system denoted by $\mathcal{W}$. The pose of an object in this coordinate system is represented by a pair $\mathbf{x} = (\mathbf{t}, \mathbf{R})$, $\mathbf{t} \in \Re(3)$, $\mathbf{R} \in \mathbf{SO}(3)$. The space of all poses is the special Euclidean Group $\mathbf{SE}(3) = \Re(3) \times \mathbf{SO}(3)$. Define $\mathcal{T}_e \subseteq \mathbf{SE}(3)$ to represent the space of end-effector configurations and $\mathcal{T}_t \subseteq \mathbf{SE}(3)$ to represent the space of target configurations. $\mathbf{x}_e \in \mathcal{T}_e$ and $\mathbf{x}_t \in \mathcal{T}_t$ denote the pose of the end-effector and of the target in world coordinates, respectively.

The goal in any visual servoing problem is to control the pose of an end-effector relative to a target object or target features. In this article, relative positioning is defined in terms of observable features rigidly attached to the end-effector and to the target object. Let $\mathcal{C}_e$ and $\mathcal{C}_t$ be the joint configuration space of the features rigidly attached to the end-effector and to the target, respectively, and define $\mathbf{C}_i : \mathcal{T}_i \rightarrow \mathcal{C}_i$, $i = e, t$ to be the corresponding mappings relating pose to feature configuration. With these definitions, feature-based relative positioning in the robot configuration space can be specified in terms of a constraint on the relationship between observable features.

Definition 3.1 A feature-based relative positioning task is described by a function $\mathbf{E} : \mathcal{C}_e \times \mathcal{C}_t \rightarrow \Re(n)$. This is subsequently referred to as the *task-space error function.* The end-effector is in the desired configuration if task-space error is zero.

It follows that the task-space error function implicitly defines a kinematic constraint, $\mathbf{E}'$, between target pose and end-effector pose as $\mathbf{E}'(\mathbf{x}_e, \mathbf{x}_t) = \mathbf{E}(\mathbf{C}_e(\mathbf{x}_e), \mathbf{C}_t(\mathbf{x}_t))$. Suppose that $\mathbf{E}'(\mathbf{x}_e, \mathbf{x}_t) = \mathbf{0}$, $\mathbf{x}_t$ is held fixed, and $\mathbf{E}'$ considered as a function of $\mathbf{x}_e$ satisfies the conditions of the implicit function theorem [27]. Then in the neighborhood of $\mathbf{x}_e$, the task error function defines a manifold of dimension n. This manifold represents the directions in which the manipulator can move while maintaining the desired kinematic relationship with the target. Equivalently, the task error constrains $d = 6 - n$ degrees of freedom of the manipulator. The value of d is subsequently referred to as the *degree* of the task-space error function.[1]

As a concrete illustration, suppose a point on the end-effector with coordinates $\mathbf{P}$ is to be positioned at a target point with coordinates $\mathbf{S}$. Then $\mathcal{C}_e = \mathcal{C}_t = \Re(3)$, and the task error function is simply $\mathbf{E}(\mathbf{P}, \mathbf{S}) = \mathbf{P} - \mathbf{S}$, $\mathbf{P} \in \mathcal{C}_e, \mathbf{S} \in \mathcal{C}_t$. In order to determine the constraint on the manipulator, let ${}^e\mathbf{P}$ denote the coordinates of $\mathbf{P}$ in the end-effector frame and let ${}^t\mathbf{S}$ denote the coordinates of $\mathbf{S}$ in the target frame. Define the change of coordinates operator $\circ$ as

$$\mathbf{x}_a \circ {}^a\mathbf{P} = (\mathbf{R}, \mathbf{t}) \circ {}^a\mathbf{P} = \mathbf{R}\ {}^a\mathbf{P} + \mathbf{t} = \mathbf{P}.$$

Then the feature mapping functions for this problem are:

$$\mathbf{C}_e(\mathbf{x}_e) = \mathbf{x}_e \circ {}^e\mathbf{P}, \tag{1}$$

$$\mathbf{C}_t(\mathbf{x}_t) = \mathbf{x}_t \circ {}^t\mathbf{S}. \tag{2}$$

and the constraint on end-effector pose is then

$$\mathbf{E}'(\mathbf{x}_e, \mathbf{x}_t) = \mathbf{x}_e \circ {}^e\mathbf{P} - \mathbf{x}_t \circ {}^t\mathbf{S}. \tag{3}$$

This is a constraint of degree 3 which is kinematically equivalent to a spherical joint [4].

The visual servoing problem is to define a control system that moves the end-effector into a configuration in which the task-space error is zero. The end-effector is modeled as a Cartesian positioning device with negligible dynamics. As noted above, this is a reasonable model for a look-and-move style system in which the robot is stabilized by internal encoder feedback. The target pose is assumed to be stationary. The instantaneous motion of the robot consists of a translational velocity $\mathbf{v}$ and a rotational velocity ω combined into a single

[1] This is closely related to the notion of "class" defined in [7].

velocity screw $\mathbf{r} = (\mathbf{v}; \omega)$ about a specified point $\mathbf{O}$. In most cases, $\mathbf{O} = \mathbf{0}$, the origin of the base coordinate system. However, as will be discussed subsequently, it is sometimes advantageous to choose $\mathbf{O}$ so as to yield a particular type of motion or to reduce the effects of miscalibration.

In this article, all features are observed by a stereo (two) camera system and a set of measurements called a *feature vector* are computed. Let $\mathcal{F}_e$ and $\mathcal{F}_t$ be the set of all feature vector values for the end-effector and the target object and define the mappings $\mathbf{F}_i : \mathcal{C}_i \rightarrow \mathcal{F}_i,\ i = e, t$ to relate end-effector and target pose to their respective feature vectors. It is assumed that each mapping is invertible except for a small set of configurations referred to as the *singular set* for that feature mapping.

Feedback control is based on defining an error function, $\mathbf{e}$, on feature configuration such that $\mathbf{e}$ describes the same kinematic configuration as a task-space error $\mathbf{E}$.

Definition 3.2 A image error function $\mathbf{e} : \mathcal{F}_e \times \mathcal{F}_t \rightarrow \Re(n)$ is *equivalent* to a task-space error function $\mathbf{E}$ of degree $d \leq n$ if and only if $\mathbf{E}(\mathbf{c}_e, \mathbf{c}_t) = \mathbf{0} \Leftrightarrow \mathbf{e}(\mathbf{F}_e(\mathbf{c}_e), \mathbf{F}_t(\mathbf{c}_t)) = \mathbf{0}$ for all $\mathbf{c}_e \in \mathcal{F}_e$ and $\mathbf{c}_t \in \mathcal{F}_t$ and except for possibly a small set of singular configurations.

Singular configurations are end-effector poses where the constraint on position defined by the image error function is locally of lower degree than that of the equivalent kinematic error function. For example, any configuration at which feature projection is singular is also a singular configuration for the error function.

All feedback algorithms in this article employ image errors in proportional control arrangements [11] as follows. Define

$$\mathbf{J}_e(\mathbf{x}_e) = \left.\frac{\partial \mathbf{e}}{\partial \mathbf{x}_e}\right|_{\mathbf{x}_e}$$

Recall $\mathbf{r}$ denotes the end-effector velocity screw. Considering now all quantities as functions of time, it follows that

$$\dot{\mathbf{e}} = \mathbf{J}_e(\mathbf{x}_e)\mathbf{r} \tag{4}$$

describes the relationship between change in end-effector pose and change in the image error. Presuming for the moment that $\mathbf{J}(t) = \mathbf{J}_e(\mathbf{x}_e(t))$ is square and full-rank on $\mathcal{T}_e$, hence invertible, it is well known that the proportional control law

$$\mathbf{u} = -k\,\mathbf{J}^{-1}\mathbf{e} \tag{5}$$

will drive the observed error to zero in the absence of noise or other disturbances. In particular, since the robot control system acts as a perfect integrator $\mathbf{r} = \mathbf{u}$. Combining (4) and (5) we have

$$\dot{\mathbf{e}} = \mathbf{J}\mathbf{u} = -k\,\mathbf{J}\mathbf{J}^{-1}\mathbf{e} = -k\,\mathbf{e}. \tag{6}$$

Thus, image error is an exponentially decreasing function of time and the system is asymptotically stable. However, asymptotic stability implies that $\lim_{t\to\infty} \mathbf{e} \to 0$. If $\mathbf{e}$ is equivalent to a task error $\mathbf{E}$, it follows that $\lim_{t\to\infty} \mathbf{E} \to 0$. Hence, by design the control system is guaranteed to achieve the equivalent task-space kinematic constraint.

As indicated in this development, the mapping from pose to image feature measurements is nonlinear, hence the Jacobian is, in general, a function of end-effector pose. As is shown in Section 4, the Jacobian matrix for the applications described in this paper can always be parameterized in terms of image measurements—end-effector pose estimation is not required.

In practice the system Jacobian is computed from feature measurements and estimates of the camera location and internal imaging parameters. Let $\hat{\mathbf{J}}$ denote the estimated Jacobian matrix. Then (6) becomes

$$\dot{\mathbf{e}} = \mathbf{J}\mathbf{u} = -k\,\mathbf{J}\hat{\mathbf{J}}^{-1}\mathbf{e} = -k\,\mathbf{M}\mathbf{e} \tag{7}$$

where $\mathbf{M} = \mathbf{J}\hat{\mathbf{J}}^{-1}$. A differential equation of the form $\dot{\mathbf{e}} = -k\,\mathbf{M}\mathbf{e}$ is asymptotically stable if the eigenvalues of $\mathbf{M}$ have strictly positive real parts [11]. In general, the entries of the Jacobian matrix are continuous functions of the system calibration parameters, and therefore the eigenvalues of $\mathbf{M}$ vary continuously with the calibration parameters. Hence, if the idealized closed loop system is asymptotically stable, a "slightly miscalibrated" system will also be asymptotically stable. Thus, image-based control systems with the structure outlined above have the following robustness property:

> **Calibration Insensitivity:** The accuracy with which a stable image-based control system maintains the equivalent kinematic constraint is independent of errors in the calibration of the system. This includes the extrinsic (positional) and intrinsic (imaging) parameters of both cameras as well as the manipulator kinematics.

This is one of the principle advantages of image-based control over functionally equivalent

position-based systems. A problem where position-based control is not able to perform with calibration-insensitive accuracy is given later in this article.

The task-space error need not constrain all degrees of freedom of the manipulator. More often than not, the degree of the task-space error function is smaller than the dimension of the task space. In this case, the Jacobian of the image error function (which must have rank equal to the degree of the task-space error function) is not square, and the matrix right inverse (or pseudo-inverse) must be used to compute the feedback signal. The matrix right inverse is defined as

$$\mathbf{J}^{+} = \mathbf{J}^{T}(\mathbf{J}\mathbf{J}^{T})^{-1}. \tag{8}$$

Substituting $\mathbf{J}^{+}$ for $\mathbf{J}^{-1}$ in (5) produces the velocity screw $\mathbf{u}$ which has minimum norm over all vectors which solve the original system of equations. It is occasionally the case that $\mathbf{J}$ has more rows than columns. In this case, the pseudo-inverse is defined as

$$\mathbf{J}^{+} = (\mathbf{J}^{T}\mathbf{J})^{-1}\mathbf{J}^{T}. \tag{9}$$

The appropriate interpretation of $\mathbf{J}^{+}$ is always clear from the dimensions of the matrix. Note that when $\mathbf{J}$ is square, $\mathbf{J}^{+} = \mathbf{J}^{-1}$.

Another important property of image-based systems of the form described above is that they are explicitly connected to the geometry of the robot task space. Thus, it is possible to synthesize control algorithms by "composing" the desired kinematic constraint using task-space error functions, and to automatically derive the equivalent image error function. To illustrate, suppose that $\mathbf{E}_1$ and $\mathbf{E}_2$ are two kinematic error functions and that there is at least one end-effector pose which simultaneously satisfies both. Let $\mathbf{e}_1$ and $\mathbf{e}_2$ represent the equivalent image error functions, and $\mathbf{J}_1$ and $\mathbf{J}_2$ represent the corresponding Jacobian functions. The combined task-space constraint is represented by "stacking" the system:

$$\mathbf{E}((\mathbf{c}_{1,1};\mathbf{c}_{1,2}),(\mathbf{c}_{2,1},\mathbf{c}_{2,2})) = \begin{bmatrix} \mathbf{E}_1(\mathbf{c}_{1,1},\mathbf{c}_{2,1}) \\ \mathbf{E}_2(\mathbf{c}_{1,2},\mathbf{c}_{2,2}) \end{bmatrix}. \tag{10}$$

Considered as a function of end-effector pose, this function is zero only if the component

functions are both simultaneously zero. It follows directly that the equivalent image error is

$$\mathbf{e}((\mathbf{f}_{1,1};\mathbf{f}_{1,2}),(\mathbf{f}_{2,1};\mathbf{f}_{2,2})) = \begin{bmatrix} \mathbf{e}_1(\mathbf{f}_{1,1},\mathbf{f}_{2,1}) \\ \mathbf{e}_2(\mathbf{f}_{1,2},\mathbf{f}_{2,2}) \end{bmatrix}. \tag{11}$$

and the corresponding Jacobian is

$$\mathbf{J}(\mathbf{x}_e) = \begin{bmatrix} \mathbf{J}_1(\mathbf{x}_e) \\ \mathbf{J}_2(\mathbf{x}_e) \end{bmatrix}. \tag{12}$$

The resulting image-based control system will be calibration insensitive. In short, calibration insensitivity is preserved under combination of kinematic constraints in the robot task space.

Finally, it is possible to superimpose task-space motions onto visually defined kinematic constraints, provided that the motions do not "conflict" with the constraint. Suppose that $\mathbf{u}_m$ is a motion that is to be superimposed on the end-effector while maintaining a kinematic constraint with image error $\mathbf{e}$. The motion can be projected "onto" the direction in which the robot is locally free to move and combined with feedback to preserve the kinematic constraint as follows:

$$\mathbf{u} = -k\,\mathbf{J}^{+}\mathbf{e} + (\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\mathbf{u}_m \tag{13}$$

3.2 Projection and Estimation of Point and Line Features

Camera positions in are represented by the poses $\mathbf{x}_{c1} = (\mathbf{R}_1, \mathbf{c}_1)$ and $\mathbf{x}_{c2} = (\mathbf{R}_2, \mathbf{c}_2)$. It is assumed that $\mathbf{c}_1 \neq \mathbf{c}_2$. A camera rotation matrix, $\mathbf{R}_i$, may be decomposed into three rows represented by the unit vectors $\vec{\mathbf{x}}_i$, $\vec{\mathbf{y}}_i$, and $\vec{\mathbf{z}}_i$. The infinite line containing both camera positions is referred to as the *baseline* of the system[2]. A plane containing the baseline is referred to as an *epipolar plane,* and the intersection of an epipolar plane with the camera imaging plane is referred to as an *epipolar line.* It is assumed that estimates of camera intrinsic parameters (parameters describing the mapping from camera pixel coordinates to metric units) and camera extrinsic parameters (the spatial position of the cameras relative to the manipulator coordinate system) are available. To simplify the exposition, all observed values are expressed in *normalized coordinates* in which values are scaled to metric units for a

[2]This differs somewhat from use in *e.g.* [22] where the baseline is the line *segment* defined by the two center points.

camera fitted with a unit focal length lens [8]. The units for linear and angular quantities are millimeters and degrees, respectively, unless otherwise specified. Finally, when dealing with vector or matrix quantities, the notation $(\mathbf{a};\mathbf{b})$ is shorthand for the column concatenation (stacking) of the vectors $\mathbf{a}$ and $\mathbf{b}$ and $(\mathbf{a} \mid \mathbf{b})$ is shorthand for the row concatenation of $\mathbf{a}$ and $\mathbf{b}$.

Points A point in three-dimensional space is written in uppercase boldface Roman letters, *e.g* $\mathbf{P}$ or $\mathbf{S}$. A subscripted lowercase boldface letter, *e.g.* $\mathbf{p}_i$ denotes the projection of $\mathbf{P}$ in camera i. The stereo projection of $\mathbf{P}$ is written in lowercase boldface without the subscript, *e.g.* $\mathbf{p} = (\mathbf{p}_1;\mathbf{p}_2)$. The projection of a point $\mathbf{P} = (x, y, z)^T$ expressed in world coordinates to a normalized homogeneous vector $\mathbf{p}_i = (u_i, v_i, 1)^T$ is given by

$$\begin{aligned} \mathbf{P}' &= \mathbf{R}_i(\mathbf{P} - \mathbf{c}_i) \\ \mathbf{p}_i &= (u_i, v_i, 1)^T = \frac{\mathbf{P}'}{\mathbf{P}'_z}. \end{aligned} \tag{14}$$

In vector form, this is written $\mathbf{p}_i = \mathbf{g}_i(\mathbf{P})$.

To estimate point location from a stereo observation, (14) is rewritten in the form:

$$\mathbf{A}_i(\mathbf{p}_i)\mathbf{P} = \mathbf{b}_i(\mathbf{p}_i) \tag{15}$$

where

$$\mathbf{A}_i = \begin{bmatrix} \vec{\mathbf{z}}_i u_i - \vec{\mathbf{x}}_i \\ \vec{\mathbf{z}}_i v_i - \vec{\mathbf{y}}_i \end{bmatrix} \tag{16}$$

and $\mathbf{b}_i$ is

$$\mathbf{b}_i = \mathbf{A}_i\mathbf{c}_i. \tag{17}$$

For two cameras, by defining $\mathbf{p} = (\mathbf{p}_1;\mathbf{p}_2)$, $\mathbf{A}(\mathbf{p}) = (\mathbf{A}_1(\mathbf{p}_1);\mathbf{A}_2(\mathbf{p}_2))$ and $\mathbf{b}(\mathbf{p}) = (\mathbf{b}_1(\mathbf{p}_1);\mathbf{b}_2(\mathbf{p}_2))$ the joint system can be written:

$$\mathbf{A}(\mathbf{p})\mathbf{P} = \mathbf{b}(\mathbf{p})$$

and it is possible to estimate $\mathbf{P}$ as:

$$\hat{\mathbf{P}} = \mathbf{A}^{+}(\mathbf{p})\mathbf{b}(\mathbf{p}). \tag{18}$$

Subsequently, the estimate of $\mathbf{P}$ from $\mathbf{p}$ is written $\hat{\mathbf{P}}(\mathbf{p})$.

The camera baseline is the singular region for stereo point projection.

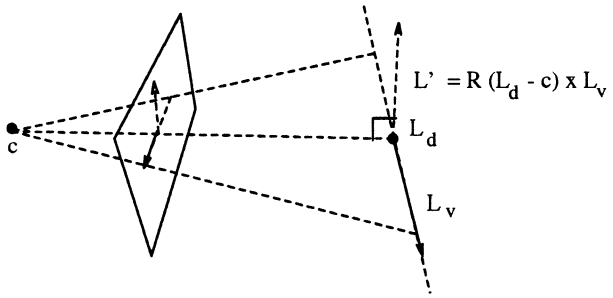

Figure 1. The geometry of line projection.

Lines An arbitrary line is parameterized by a six-tuple $\mathbf{L} = (\mathbf{L}_d; \mathbf{L}_v)$ where $\mathbf{L}_d \in \Re(3)$ is fixed point on the line and $\mathbf{L}_v \in \Re(3)$ is a unit vector representing the direction of the line. The vector $\mathbf{l}_i$ parameterizing the projection of $\mathbf{L}$ in normalized coordinates in camera i is given by

$$\begin{aligned} \mathbf{L}' &= \mathbf{R}_i (\mathbf{L}_v \times (\mathbf{L}_d - \mathbf{c}_i)) \\ \mathbf{l}_i &= \frac{\mathbf{L}'}{\sqrt{\mathbf{L}'^2_x + \mathbf{L}'^2_y}}. \end{aligned} \tag{19}$$

In vector form, this is written $\mathbf{l}_i = \mathbf{h}_i(\mathbf{L})$. As with points, the stereo projection of $\mathbf{L}$ is written $\mathbf{l} = (\mathbf{l}_1; \mathbf{l}_2)$.

Geometrically, $\mathbf{L}'$ is normal to the plane passing through the center of projection of the camera containing the point $\mathbf{L}_d$ and the vector $\mathbf{L}_v$. The projection of $\mathbf{L}$ in a camera image is the intersection of this plane with the imaging plane. By normalizing the projection as shown, the first two components of $\mathbf{l}_i$ encode the normal to the projection of the line, and the final component is the distance from the line to the image origin (Figure 1). Note that line projection is not defined when $\mathbf{L}_v$ is parallel to the viewing direction $(\mathbf{L}_d - \mathbf{c}_i)$.

The projection of a line in an image can be easily related to the projection of points on the line. Let $\mathbf{L}$ be a line containing $\mathbf{P}$ and $\mathbf{S}$ with $\mathbf{L}_v$ directed from $\mathbf{P}$ to $\mathbf{S}$. Then it is easy to show that $\mathbf{l}_i$ is given by

$$\mathbf{L}' = \mathbf{s}_i \times \mathbf{p}_i \tag{20}$$

$$\mathbf{l}_i = \frac{\mathbf{L}'}{\sqrt{\mathbf{L}'^2_x + \mathbf{L}'^2_y}}. \tag{21}$$

Furthermore, for any homogeneous vector $\mathbf{p}_i$ in the image, it can be shown that $\mathbf{p}_i \cdot \mathbf{l}_i$ is

the distance between the point and the projection of the line in the image plane. It follows that a homogeneous vector $\mathbf{p}_i$ in camera image i lies on the line projection $\mathbf{l}_i$ if and only if $\mathbf{p}_i \cdot \mathbf{l}_i = 0$.

In order to estimate the parameters of a line from a stereo observation, observe that the direction of the line can be computed by

$$\hat{\mathbf{L}}_v = \frac{\mathbf{R}_1^T \mathbf{l}_1 \times \mathbf{R}_2^T \mathbf{l}_2}{\|\mathbf{R}_1^T \mathbf{l}_1 \times \mathbf{R}_2^T \mathbf{l}_2\|}. \tag{22}$$

It is assumed that $\mathbf{L}_d$ is some fixed observable point on the line, so its value can be computed from its stereo projection as described above. Note that (22) is not defined when $\mathbf{l}_1$ and $\mathbf{l}_2$ are parallel. This occurs only when $\mathbf{L}$ lies in an epipolar plane. Hence any epipolar plane is a singular region for stereo line projection.

There is an ambiguity in the sign of $\hat{\mathbf{L}}_v$ in this construction. It can be resolved by computing the values $s_i = \text{signum}(\mathbf{h}_i(\hat{\mathbf{L}}))$, $i = 1, 2$. If $s_1 * s_2 < 0$, the assignment of line normals to inputs is *inconsistent* (there is no line such that the direction of the input agrees with the specified direction of the line in both images). Otherwise, the correct estimate of line direction is $\text{signum}(s_1)\widehat{\mathbf{L}_v}$.

Subsequently, the estimate of $\mathbf{L}$ from $\mathbf{l}$ is written $\hat{\mathbf{L}}(\mathbf{l})$.

4 Positioning Skills Based on Points and Lines

As discussed in the introduction, our paradigm for developing robust control algorithms for hand-eye coordination proceeds as follows:

1. Develop a set of simple, generic task-space kinematic constraints and their equivalent image error functions, referred to here as *primitive skills*. As discussed in Section 3.1, by construction these skills will define image-based positioning operations that are calibration insensitive.

2. For a particular application produce, via composition of primitive skills, the desired task-space kinematic constraints. Use the equivalent error functions and equivalent Jacobians to produce a vision-based regulator for achieving these kinematic constraints.

3. Superimpose any desired motions onto the kinematic constraints using (13).

In this section, three primitive positioning operations utilizing point and line features are defined. These positioning operations are illustrated in three example applications. Finally, the sensitivity of stability of the primitives to camera calibration error is examined, and the effect of the choice of center of rotation is discussed.

4.1 Three Positioning Primitives

4.1.1 Point-to-Point Positioning

Recall the example of Section 3. Formally, the problem is

> Given a reference point $\mathbf{P}$ fixed with respect to a target object and a point $\mathbf{S}$ rigidly attached to the end-effector, develop a regulator that positions the end-effector so that $\mathbf{P} = \mathbf{S}$.

The corresponding feature-based task error function is

$$\mathbf{E}(\mathbf{S}, \mathbf{P}) = \mathbf{S} - \mathbf{P}. \tag{23}$$

The solution to this problem is based the observation that two points not on the camera baseline are coincident in space if and only if their stereo projections are coincident. This motivates the error function

$$\mathbf{e}_{pp}(\mathbf{s}, \mathbf{p}) = \mathbf{s} - \mathbf{p}. \tag{24}$$

Since the error function is a linear function of stereo point projection the singular set of stationing configurations is exactly the singular set of the point projection function. Thus, the system cannot execute a positioning operation that requires stationing at any point along the camera baseline.

To solve this problem, first consider computing only pure translations, $\mathbf{v}$. The solution to this problem was presented in [14]. Defining $D_i = (\mathbf{S} - \mathbf{c}_i) \cdot \vec{\mathbf{z}}_i$, the Jacobian of point projection is obtained by differentiating (14):

$$\mathbf{J}_{g_i}(\mathbf{S}) = \frac{1}{D_i^2} \begin{bmatrix} \vec{\mathbf{x}}_i^T D_i - \vec{\mathbf{z}}_i^T((\mathbf{S} - \mathbf{c}_i) \cdot \vec{\mathbf{x}}_1) \\ \vec{\mathbf{y}}_i^T D_i - \vec{\mathbf{z}}_i^T((\mathbf{S} - \mathbf{c}_i) \cdot \vec{\mathbf{y}}_1) \\ 0 \end{bmatrix} \tag{25}$$

Combining (25) with point estimation of $\mathbf{S}$ from its stereo projection, the Jacobian for the error term $\mathbf{e}_{pp}$ is

$$\mathbf{J}_{pp}(\mathbf{s}) = \begin{bmatrix} \mathbf{J}_{g_1}(\hat{\mathbf{S}}(\mathbf{s})) \\ \mathbf{J}_{g_2}(\hat{\mathbf{S}}(\mathbf{s})) \end{bmatrix}. \tag{26}$$

Note that $\mathbf{J}_{pp}$ is not square. This is because $\mathbf{g}$ maps three values—the Cartesian position of a point—into six values—the homogeneous camera image locations of the projections of the point. Thus, the desired robot translation is computed by

$$\mathbf{v} = -k\, \mathbf{J}_{pp}^{+}(\mathbf{s})\mathbf{e}_{pp}(\mathbf{s}, \mathbf{p}), \quad k > 0. \tag{27}$$

In order to compute the six degree of freedom velocity screw, observe that the motion of $\mathbf{S}$ for a given velocity screw $\mathbf{r} = (\mathbf{v}, \omega)$ and center of rotation $\mathbf{O}$ is

$$\dot{\mathbf{S}} \;=\; \omega \times (\mathbf{S} - \mathbf{O}) + \mathbf{v} \tag{28}$$

Recall that the expression $\mathbf{a} \times \mathbf{b}$ can be written as $sk(\mathbf{a})\mathbf{b} = sk(-\mathbf{b})\mathbf{a}$ where the skew symmetric matrix is:

$$sk((x, y, z)^T) = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}.$$

Defining $\mathbf{D}(\mathbf{a}) = (\mathbf{I} \mid sk(-\mathbf{a}))$ where $\mathbf{I}$ is the 3 by 3 identity matrix, (28) can be rewritten

$$\dot{\mathbf{S}} = \mathbf{D}(\mathbf{S} - \mathbf{O})\mathbf{r}. \tag{29}$$

Thus, a value for the end-effector screw can be defined by utilizing (28) and the matrix right inverse to compute a velocity screw that performs the minimum norm six degree-of-freedom motion equivalent to the pure translation needed to solve the problem:

$$\mathbf{u} = \mathbf{D}^{+}(\hat{\mathbf{S}}(\mathbf{s}) - \mathbf{O})\mathbf{v} = -k\, \mathbf{D}^{+}(\hat{\mathbf{S}}(\mathbf{s}) - \mathbf{O})\mathbf{J}_{pp}^{+}(\mathbf{s})\mathbf{e}_{pp}(\mathbf{s}, \mathbf{p}), \quad k > 0. \tag{30}$$

This expression can be simplified if the dimensionality of the image error can be made to match that of the kinematic constraint. For example, if the cameras are arranged as a stereo pair so that the y axes are parallel, then one of the y components of the camera observations can be discarded. Let $\mathbf{J}'_{pp}$ be $\mathbf{J}_{pp}$ with the two zero rows and the row corresponding to the

redundant error term removed. Let $\mathbf{e}_{pp6}$ be the error term with the corresponding elements deleted. Then the Jacobian relating the end-effector screw to stereo point motion is

$$\mathbf{J}_{pp6}(\mathbf{O}, \mathbf{s}) = \mathbf{J}'_{pp}(\mathbf{s})\mathbf{D}(\hat{\mathbf{S}}(\mathbf{s}) - \mathbf{O}) \tag{31}$$

and the end-effector screw is given by

$$\mathbf{u} = \mathbf{J}^{+}_{pp6}(\mathbf{O}, \mathbf{s})\mathbf{e}_{pp6}(\mathbf{s}, \mathbf{p}). \tag{32}$$

4.1.2 Point-to-Line Positioning

> Given two a reference points **P** and **Q** fixed with respect to a target object and a reference point **S** rigidly attached to the end-effector, develop a regulator that positions the end-effector so that **P**, **Q**, and **S** are collinear.

The corresponding task error function is given by

$$\mathbf{E}_{pl}(\mathbf{P}, \mathbf{Q}, \mathbf{S}) = (\mathbf{S} - \mathbf{P}) \times (\mathbf{Q} - \mathbf{P}) \tag{33}$$

Although $\mathbf{E}_{pl}$ is a mapping into $\Re(3)$, placing a point onto a line is a constraint of degree 2. It is interesting to note that this is a positioning operation which cannot be performed in a calibration insensitive fashion using position-based control [12].

The points **P** and **Q** define a line in space. Let **L** parameterize this line. Then a functionally equivalent task specification is:

> Given a reference line **L** rigidly attached to a target object and a reference point **S** rigidly attached to the end-effector, develop a regulator that positions the end-effector so that $\mathbf{S} \in \mathbf{L}$.

The corresponding task error function is given by

$$\mathbf{E}_{pl}(\mathbf{S}, \mathbf{L}) = (\mathbf{S} - \mathbf{L}_d) \times \mathbf{L}_v \tag{34}$$

The latter is more compact and will be used subsequently with the understanding that any two points are equivalent to a line.

The equivalent error term for $\mathbf{E}_{pl}$ is based on the observation that for an arbitrary line **L** that does not lie in an epipolar plane and a point **P** not on the baseline, $\mathbf{l}_1 \cdot \mathbf{p}_1 = \mathbf{l}_2 \cdot \mathbf{p}_2 = 0$ if

and only if $\mathbf{P} \in \mathbf{L}$. This fact can be verified by recalling that the projection of $\mathbf{L}$ in a camera image defines a plane containing $\mathbf{L}$. If the projection of $\mathbf{P}$ is on this line, the $\mathbf{P}$ must be in this plane. Applying the same reasoning to a second camera, it follows that $\mathbf{P}$ must lie at the intersection of the planes defined by the two cameras. But, this is exactly the line $\mathbf{L}$. Thus, define a positioning error $\mathbf{e}_{pl}$ as:

$$\mathbf{e}_{pl}(\mathbf{s}, \mathbf{l}) = \begin{bmatrix} \mathbf{s}_1 \cdot \mathbf{l}_1 \\ \mathbf{s}_2 \cdot \mathbf{l}_2 \end{bmatrix} \tag{35}$$

The Jacobian is

$$\mathbf{J}_{pl}(\mathbf{O}, \mathbf{s}, \mathbf{l}) = \begin{bmatrix} \mathbf{l}_1^T \\ \mathbf{l}_2^T \end{bmatrix} \mathbf{J}_{pp6}(\mathbf{O}, \mathbf{s}) \tag{36}$$

where $\mathbf{J}_{pp6}$ is as defined in (26). The error function is a linear function of line projection, hence the set of singular stationing configurations are those which require placing a point on a line lying in an epipolar plane.

4.1.3 Line-to-Point Positioning

Consider now the following modification of the previous problem

> Given a reference line $\mathbf{L}$ rigidly attached to the end-effector and a reference point $\mathbf{S}$ rigidly attached to a target object, develop a regulator that positions the end-effector so that $\mathbf{S} \in \mathbf{L}$.

This problem has the same task error function and image-space error function as the previous problem, but now $\mathbf{e}_{pl}$ depends on the time derivative of $\mathbf{l}$. By the chain rule, this derivative is composed of two terms: the Jacobian of the normalization operation, and the Jacobian of the unnormalized projection. The Jacobian of the normalization operation is:

$$\mathbf{N}((a,b,c)^T) = \frac{1}{(a^2+b^2)^{3/2}} \begin{bmatrix} a^2 & -ab & 0 \\ -ab & b^2 & 0 \\ -ca & -cb & (a^2+b^2) \end{bmatrix}. \tag{37}$$

The Jacobian of the expression $\mathbf{L}_1' = \mathbf{R}_1(\mathbf{L}_v \times (\mathbf{L}_d - \mathbf{c}_1))$ is

$$\mathbf{J}_1'(\mathbf{L}) = \mathbf{R}_1 \left(sk(\mathbf{L}_v) \mid (sk(\mathbf{L}_v) sk(\mathbf{O} - \mathbf{L}_d) + sk(\mathbf{L}_d - \mathbf{c}_1) sk(\mathbf{L}_v)) \right) \tag{38}$$

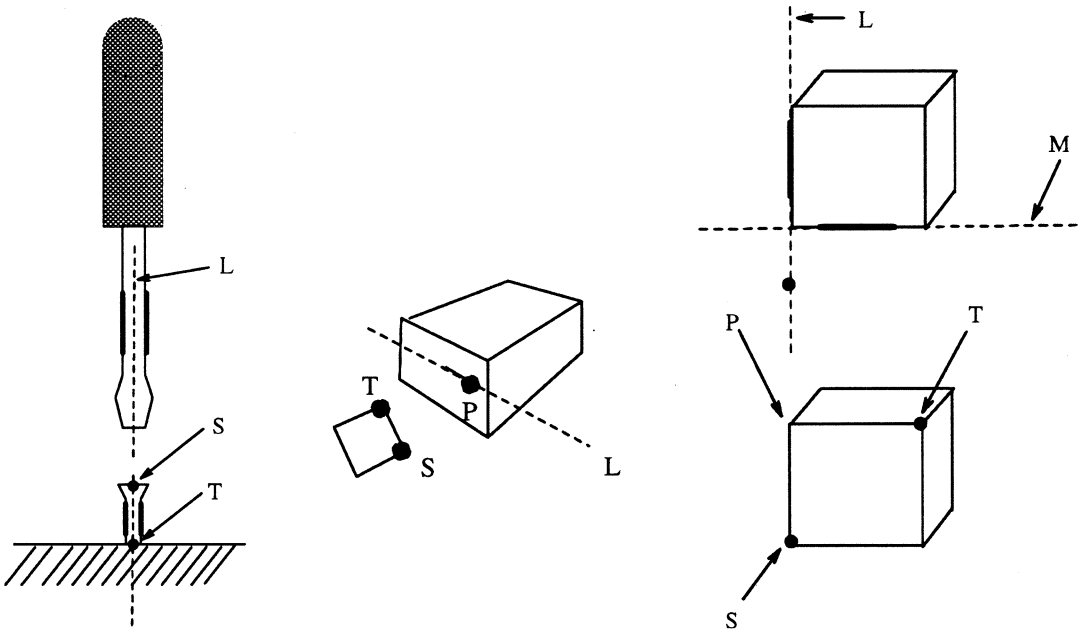

Figure 2. Examples of tasks using visual positioning. The thick lines and points indicated tracked features. Left: aligning a screwdriver with a screw. Middle: positioning a floppy disk at a disk drive opening. Right: stacking blocks.

Note that if $\mathbf{L}_d$ is chosen as the point of rotation of the system, $\mathbf{O} = \mathbf{L}_d$ and (38) simplifies to

$$\mathbf{J}_1'(\mathbf{L}) = \mathbf{R}_1 \left(sk(\mathbf{L}_v) \mid sk(\mathbf{L}_d - \mathbf{c}_1) sk(\mathbf{L}_v) \right) \tag{39}$$

Combining this with the estimation of $\mathbf{L}$ from its stereo projection, the Jacobian is

$$\mathbf{J}_{lp}(\mathbf{O}, \mathbf{l}, \mathbf{s}) = \begin{bmatrix} \mathbf{s}_1^T \mathbf{N}(\mathbf{l}_1) \mathbf{J}_1'(\hat{\mathbf{L}}(\mathbf{l})) \\ \mathbf{s}_2^T \mathbf{N}(\mathbf{l}_2) \mathbf{J}_2'(\hat{\mathbf{L}}(\mathbf{l})) \end{bmatrix} \tag{40}$$

The singular stationing points are points along the camera baseline.

4.2 Example Compositions

The error measures for the point-to-point and point-to-line operations can be used to define a number of higher degree kinematic constraints.

4.2.1 Alignment

Consider Figure 2 (left) in which a visual positioning operation is to be used to place a screwdriver onto a screw. The desired task-space kinematic constraint is to align the axis of the screwdriver with the axis of the screw. Because the central axes of the screwdriver and the screw are not directly observable, other image information must be used to compute their locations. The occluding contours of the screwdriver shaft and the screw provide enough information to determine the "virtual" projection of the central axis [12]. The intersection of the axis with tip of the screwdriver and the top of the screw respectively form fixed observable points on each as required for line parameterization.

One possibility for solving this problem is to extend the set of primitive skills to include a "line-to-line" positioning primitive. A second possibility using only the tools described above can be developed by noting that the intersection of the screw with the surface defines a second fixed point on the screw. This motivates the following positioning problem:

> Given a reference line $\mathbf{L}$ rigidly attached to the end-effector and two points $\mathbf{S}$ and $\mathbf{T}$ rigidly attached to a target object, develop a regulator that positions the end-effector so that $\mathbf{S} \in \mathbf{L}$ and $\mathbf{T} \in \mathbf{L}$.

This task can now be solved using two line-to-point operations. Define

$$\mathbf{E}_{al}(\mathbf{S},\mathbf{T},\mathbf{L}) = \begin{bmatrix} \mathbf{E}_{pl}(\mathbf{S},\mathbf{L}) \\ \mathbf{E}_{pl}(\mathbf{T},\mathbf{L}) \end{bmatrix} \tag{41}$$

Then, the equivalent image-based error is

$$\mathbf{e}_{al}(\mathbf{s},\mathbf{t},\mathbf{l}) = \begin{bmatrix} \mathbf{e}_{pl}(\mathbf{s},\mathbf{l}) \\ \mathbf{e}_{pl}(\mathbf{t},\mathbf{l}) \end{bmatrix} \tag{42}$$

The line feature is associated with the moving frame, so the Jacobian is

$$\mathbf{J}_{al}(\mathbf{O},\mathbf{s},\mathbf{t},\mathbf{l}) = \begin{bmatrix} \mathbf{J}_{lp}(\mathbf{O},\mathbf{l},\mathbf{s}) \\ \mathbf{J}_{lp}(\mathbf{O},\mathbf{l},\mathbf{t}) \end{bmatrix} \tag{43}$$

This defines a collinearity constraint that aligns two points to an axis, but leaves rotation about the axis and translation along the axis free. Once the alignment is accomplished, a

motion along the alignment axis can be superimposed (using (13)) to place the screwdriver onto the screw, and finally a rotation about the alignment axis can be superimposed to turn the screw. Note that the screw cannot be parallel to the camera baseline as this is a singular configuration for the component positioning operations.

4.2.2 Positioned Alignment

Consider inserting a floppy disk into a disk drive as shown Figure 2 (middle). The desired task-space kinematic constraint can be stated as placing one corner of the disk at the edge of the drive slot, and simultaneously aligning the front of the disk with the slot. This motivates the following positioning problem:

> Given a reference point $\mathbf{P}$ on a line $\mathbf{L}$ rigidly attached to a target object and two reference points $\mathbf{S}$ and $\mathbf{T}$ rigidly attached to the end-effector, develop a regulator that positions the end-effector so that $\mathbf{P} = \mathbf{S}$ and $\mathbf{T} \in \mathbf{L}$ using their stereo projections.

The task error is

$$\mathbf{E}_{pal}(\mathbf{P}, \mathbf{L}, \mathbf{S}, \mathbf{T}) = \begin{bmatrix} \mathbf{E}_{pp}(\mathbf{S}, \mathbf{P}) \\ \mathbf{E}_{pl}(\mathbf{T}, \mathbf{L}) \end{bmatrix} \tag{44}$$

The corresponding image error and Jacobian result by stacking the corresponding image error terms and Jacobians for the primitive operations as above. The singular set is the union of the singular sets of the primitives.

4.2.3 Six Degree-of-Freedom Positioning

Suppose an application requires a stacking operation as illustrated in Figure 2 (right). The desired task-space kinematic constraint is to align one side the bottom of the upper block with the corresponding side and top of the lower block, respectively. This constraint forms a rigid link between the two blocks. Consider the following definition of a rigid link between end-effector and target frames:

> Given three non-collinear, reference points $\mathbf{P}$, $\mathbf{S}$ and $\mathbf{T}$ rigidly attached to a target object, and two non-parallel reference lines $\mathbf{L}$ and $\mathbf{M}$ rigidly attached to the end-effector, develop a regulator that positions the end-effector so that $\mathbf{P} \in \mathbf{L}$, $\mathbf{S} \in \mathbf{L}$, and $\mathbf{T} \in \mathbf{M}$.

To see that these constraints fully define the position of the end-effector relative to the target, note that positioning the points $\mathbf{P}$ and $\mathbf{S}$ on $\mathbf{L}$ is the four degree-of-freedom alignment operation described above. When $\mathbf{P} \in \mathbf{L}$ and $\mathbf{S} \in \mathbf{L}$, satisfying $\mathbf{T} \in \mathbf{M}$ can be accomplished by first rotating about the line $\mathbf{L}$ until $((\mathbf{T} - \mathbf{M}_d) \times \mathbf{M}_v) \cdot \mathbf{L}_v = 0$. $\mathbf{L}_v$ is now parallel to the plane defined by $\mathbf{M}$ and $\mathbf{T}$, so it is possible to translate along $\mathbf{L}$ until $\mathbf{T} \in \mathbf{M}$.

The task error is

$$\mathbf{E}_{rp}(\mathbf{L}, \mathbf{M}, \mathbf{P}, \mathbf{S}, \mathbf{T}) = \begin{bmatrix} \mathbf{E}_{pl}(\mathbf{P}, \mathbf{L}) \\ \mathbf{E}_{pl}(\mathbf{S}, \mathbf{L}) \\ \mathbf{E}_{pl}(\mathbf{T}, \mathbf{M}) \end{bmatrix}. \tag{45}$$

The corresponding image error and Jacobian result by stacking the corresponding image error terms and Jacobians for the primitive operations. The singular set is for this operation is any setpoint which forces $\mathbf{L}$ or $\mathbf{M}$ to lie in an epipolar plane.

4.3 Choosing a Center of Rotation

In the discussion thus far, the choice of origin for rotations has been left free. The usual choice for $\mathbf{O}$ is $\mathbf{O} = \mathbf{0}$, placing the center of rotation at the origin of the robot coordinate system. However, by strategically placing $\mathbf{O}$, rotations and translation can be decoupled at a specific point leading to more "intuitive" motions. For example, choosing $\mathbf{O}$ to be the tip of the screwdriver causes the tip to undergo pure translation, with all rotations for alignment leaving the tip position fixed.

In this example $\mathbf{O}$ can be calculated directly using the point estimation techniques described above. This makes it possible to completely parameterize Jacobian matrices in terms of observable quantities, and has the additional advantage of reducing the effect of calibration error. For example, the Jacobian relating the end-effector screw to the motion of a point $\mathbf{P}$ depends on the expression $\mathbf{P} - \mathbf{O}$. Estimating both points and computing the difference cancels any constant reconstructive error, *e.g.* an error in the position of the robot relative to the cameras. Furthermore, errors in point reconstruction due to miscalibration typically increase with distance from the camera. If $\mathbf{P}$ is close to $\mathbf{O}$, the effect of nonlinear reconstructive errors will be kept relatively small.

In a hierarchical control scheme, the desired center of rotation in end-effector coordinates is needed in order to parameterize the resolved rate control. Thus, in order to choose an arbitrary center of rotation, its location relative to the physical center of the wrist must be known. As above, in order to minimize the effect of calibration and reconstruction errors the desired wrist center should be set by computing a difference between estimates of the physical wrist center and the desired origin. If the physical wrist center is not directly observable it is well-known that any three non-collinear points with known end-effector coordinates can be used to reconstruct its location [17, Chapter 14].

4.4 Sensitivity to Calibration Errors

In the absence of noise and calibration error, the systems defined above are guaranteed to be asymptotically stable at points where the Jacobian matrix is nonsingular. Implementations of these algorithms have shown them to be remain stable even when exposed to radical errors in system calibration.

In this section, the sensitivity of stability to certain types of calibration error is briefly examined. In particular, it is well known that the accuracy of stereo reconstruction is most sensitive to the length of the camera baseline and the relative camera orientation. Consider a two-dimensional coordinate system in which the camera baseline forms the x axis. The distance between two cameras is parameterized by the length of the baseline, l. The direction of gaze is parameterized by a vergence angle θ where $\theta = 0$ means the cameras point along the y axis. Positive values of θ denote symmetric vergence inward, and negative values denote symmetric vergence outward. Let $\hat{\theta}$ and $\hat{l}$ denote the estimated values of θ and l, respectively.

Consider the point-to-point control algorithm resulting from 2-D versions of expressions (14), (16), (17) and (26). As noted in Section 3, the closed-loop behavior of the resulting control system at a setpoint $\mathbf{P} = (x, y)$ is described by a linear differential equation of the form

$$\dot{\mathbf{e}} = -\mathbf{J}_{pp}(\mathbf{P})\hat{\mathbf{J}}_{pp}^{-1}(\mathbf{P}) = -\mathbf{M}(\mathbf{P})\mathbf{e} \tag{46}$$

where $\mathbf{J}_{pp}(\mathbf{P})$ is the true system Jacobian and $\hat{\mathbf{J}}_{pp}(\mathbf{P})$ represents the Jacobian matrix computed using $\hat{l}$ and $\hat{\theta}$ as well as observed values of $\mathbf{P}$. A noted earlier, such a system is

asymptotically stable at $\mathbf{P}$ if and only if the eigenvalues of $\mathbf{M}(\mathbf{P})$ have strictly positive real parts [11].

First fix $l = \hat{l} = 1$ and consider the effect of errors in the estimate of camera vergence. Expression (46) for this case was constructed using a symbolic mathematics package, and the equations describing the points at which the system is asymptotically state were computed. Particularly simple yet representative solutions result when the physical cameras point straight ahead ($\theta = 0$) and $\hat{\theta}$ is allowed to range freely. When $\hat{\theta} > 0$, the region of stable points is bounded by two lines:

$$y = -\tan(\hat{\theta})(x-1) \tag{47}$$

$$y = \tan(\hat{\theta})(x+1) \tag{48}$$

These two equations define an open-ended cone that is bounded by lines forming angles $\hat{\theta}$ with the baseline. When $\hat{\theta} < 0$, the set of stable points is a bounded circular region,

$$(y + \cot(\hat{\theta}))^2 + x^2 = \csc(\hat{\theta})^2. \tag{49}$$

Note that the size of the circle shrinks quickly as the magnitude of $\hat{\theta}$ increases. This accords with intuition. For a fixed value of θ and a point $\mathbf{P}$, as $\hat{\theta}$ is decreased the effect is to "push" the estimated location of the observed point away from the cameras. Eventually, the calibration describes a camera system which could not physically produce the actual observations of $\mathbf{P}$ and $\mathbf{P}$ then becomes an unstable equilibrium. Numerical computation of the set of stable points for problems of higher degree has shown qualitatively similar effects.

Suppose now that $\theta = \hat{\theta}$ and consider errors in the estimated baseline $\hat{l}$. In the closed loop equation, the ratio $\hat{l}/l$ appears as a gain term, and therefore does not affect stability of the continuous time system. Gain terms do however affect the stability of discrete time systems. The discrete time model for a perfectly calibrated system with unit time delay is of the form

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

where k is a gain coefficient and t is the sampling time (one over the sampling rate). This system has characteristic polynomial $x^2 + x + tk = 0$ [11]. System is stability is guaranteed when $k < 1/t$. The system is overdamped if $tk < 1/4$ and underdamped if $tk > 1/4$. Thus,

for example, overestimating the baseline distance by 10% has the effect of introducing a fixed gain factor of 1.1 into the closed-loop system and is therefore a destabilizing factor. Errors in any other coefficients that enter the equations as a scale factor, including camera focal length and scaling from pixel to metric coordinates, exhibit similar effects. These parameters can typically be estimated quite precisely (easily to within 1%) so their effects are minute compared to the impact of errors in the relative position, particularly orientation, of the cameras.

5 Experiments

All of the primitive and composed skills described above have been implemented and tested on an experimental visual servoing system. The system consists of a Zebra Zero robot arm with PC controller, two Sony XC-77 cameras with 12.5 mm 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 continually sent to the PC which converts them into coordinated joint motions using a resolved-rate controller operating at 140 Hz. The Sun-PC connection is implemented using an optimized ethernet package which yields transmit delays below a millisecond on an isolated circuit. As the system runs, it logs 5 minutes of joint motion information at 20 Hz which can be used to examine the dynamic behavior of the system. All test cases were designed not to pass near singularities.

A custom tracking system written in C++ provides visual input for the controller. The system, more fully described in [15, 13], provides extremely 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 occluding contours with edge trackers arranged as corners or parallel strips. In all experiments, the occluding contours were of high contrast so that other background distractions were easily ignored by the tracker. Specifics of the tracking setup for each application are described below.

The hand-eye system was calibrated by tracking a point on the manipulator as it moved to a series of positions, and applying a least-squares minimization to generate the calibration parameters [24].

5.1 Accuracy

Several experiments were performed to determine the positioning accuracy and stability of the control methods. Stereo images from the experimental setup are shown in Figure 4. The top set of images shows the system in a goal configuration where it is attempting to touch the corners of two 3.5 inch floppy disks. The disks are a convenient testing tool since their narrow width (approximately 2.5 mm) makes them easy to track and at the same time makes it simple to measure the accuracy of positioning and orientation. Motions are defined by tracking one, two, or three corners of the disks. The length of the tracked segments was 20 pixels, and the search area around a segment was ± 10 pixels. The cameras were placed 80 centimeters from the robot along the x axis, 30 centimeters apart along the y axis, and were oriented to point back roughly along the x axis of the robot with a vergence of approximately 10 degrees.

Positioning To test the accuracy and repeatability of point-to-point positioning, the robot was guided along a square trajectory defined by the sides and top of a target disk. At each endpoint, it descended to touch opposing corners of the disks. It was allowed to settle for a few seconds at each trajectory endpoint and the accuracy of the placement was observed. The expected positioning accuracy at the setpoint depends on the error in edge localization. One camera pixel has a width of approximately 0.01mm. At 80 cm with 12.5 mm focal length lenses on both cameras, the expected vertical and horizontal positioning accuracy is ± 0.32 mm, and the expected accuracy in depth is ± 1.75 mm. Consequently, the system should be able to reliably position the corners of the disks so that they nearly touch one another.

The system has performed several hundred cycles of point-to-point motion under varying conditions over a period of several months. In nearly all cases, the system was able to position the disks so that the corners touched. In fact, typical accuracy was well below that predicted—usually less than a millimeter of relative positioning error. This is an order of

magnitude better than the absolute positioning precision of the robot itself. As expected, this error is independent of the fidelity of the system calibration. Occasionally the system failed due to systematic detection bias in the edge tracking system. These biasing problems are due to "blooming" effects in the CCD cameras. These only appear when the contrast across an edge becomes excessive.

The entire visual control system (including tracking and control signal computation) runs at a rate of 27 Hz. For these trials robot velocities were limited to a maximum of 8 cm/sec. The total time lag in the system (from images to robot motion) is estimated as follows: the maximum frame lag (1/30 sec.) plus processing time (1/27 sec.) plus transmission delay from Sun to robot (measured at less than 1/1000 sec.) plus delay in the resolved rate control (1/140 sec.) yielding a worst case delay of 0.079 sec. Using the discrete-time model given in Section 4.4, this suggests that the system should first begin to exhibit underdamped behavior at a proportional gain of 3.18.

Several trials were performed to test this prediction. Each trial consisted of having the system move from a fixed starting position to a setpoint. The proportional gain values were varied for each trial. Figure 3 shows the recorded motions. As expected, the system is well-behaved, exhibiting generally small corrections of ± 0.6mm about the setpoint for gains of up to 3.0. At a gain of 4.0 slightly underdamped behavior can be observed and at a level of 5.0 the system is clearly underdamped.

Position and Orientation The point-to-point skill was combined with a point-to-line skill to examine the effectiveness of orientation control. Input was provided by tracking an additional corner on both disks. The point of rotation was chosen to be the corner of the disk used to define the point-to-point motion in order to decouple translation to the setpoint from rotation to produce alignment. Experimentally, the positioning accuracy of the system was observed to be unchanged. The accuracy of the alignment of the sides of the two disks was observed to be within ± 1.0 degrees. With the increased tracking load and numerical calculations, the cycle time dropped to 9.5 Hz. At this rate, the system is expected to be overdamped until a gain of 1.7. Figure 5 shows the system response to a step input for varying gain values. The values shown are the angles between the tool z axis and the world

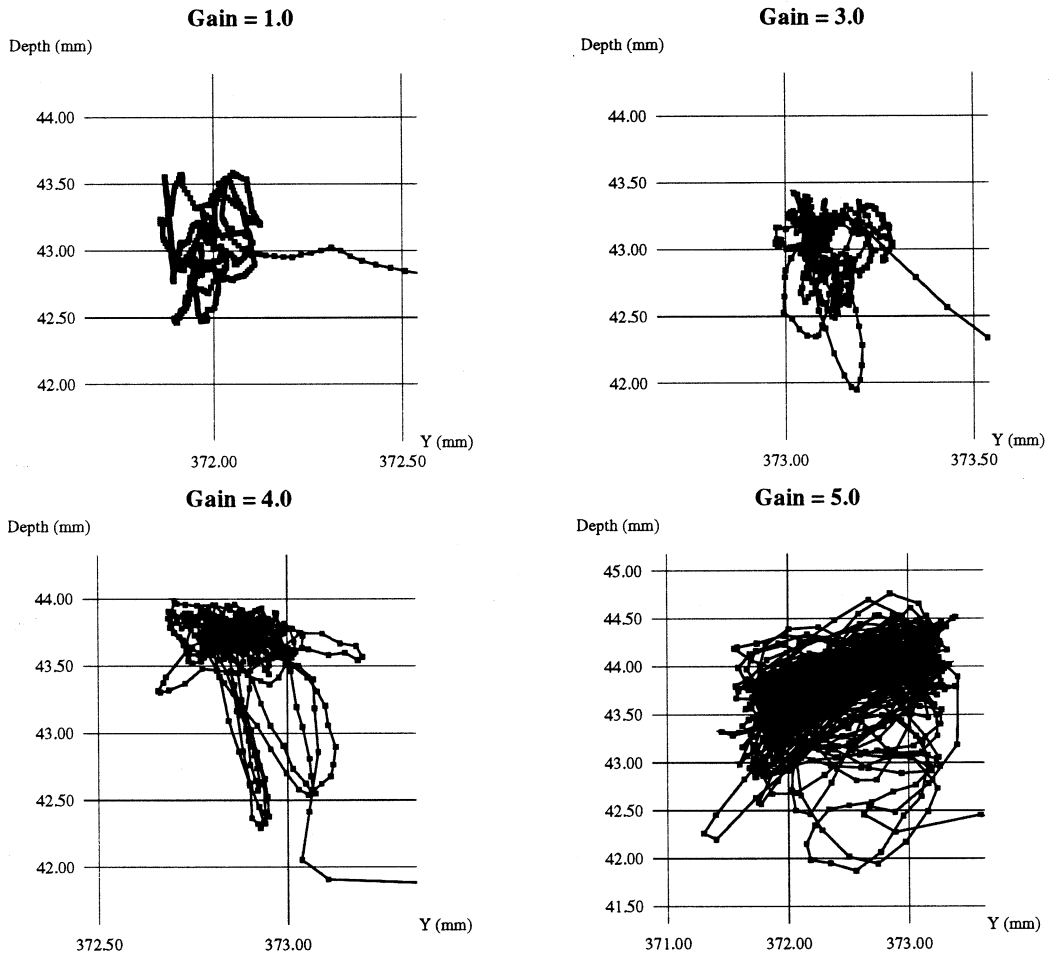

Figure 3. The position of the robot end-effector during execution of the same point-to-point motion with various proportional gains. The dots are separated by 1/20 second.

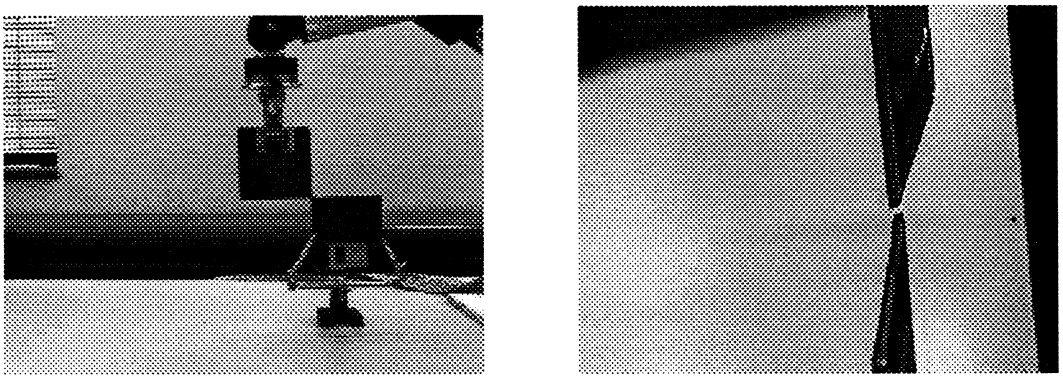

Figure 4. Left, the left camera eye view of the system touching the corners of two floppy disks. Right, the accuracy in depth with which the positioning occurs.

$x - z$ and $y - z$ planes. As seen from the upper left graph, the system tends to exhibit a constant cyclic motion about the setpoint on the order of ± 0.6 degrees relative to the $y - z$ plane and ± 0.05 degrees relative to the $x - z$ plane. The former is the direction along the optical axes (the depth direction) which explains the lower accuracy. At higher gain levels the data also shows some aliasing (due to the discrete nature of images) as well as backlash and friction in the robot drive system.[3] As expected, with the exception of these effects the system behaves consistently to a gain level of 2.0 and becomes rapidly underdamped thereafter.

5.2 Calibration Insensitivity

Experiments were performed to test the calibration sensitivity of the system. The point-to-point positioning controller was used. The proportional gain was set to 2.0 and the system was allowed to settle at the setpoint. Then, the physical cameras were perturbed from their nominal position while the system was running until clearly underdamped behavior resulted in response to a small step input produced by jostling the target disk.

First, the left camera was rotated inward. As noted in Section 4.4, this is the type of miscalibration to which the system is expected to be most sensitive. The system became

[3]The shaft encoders are mounted before the gear train driving the joints, hence the inverse kinematics exhibit hysteresis.

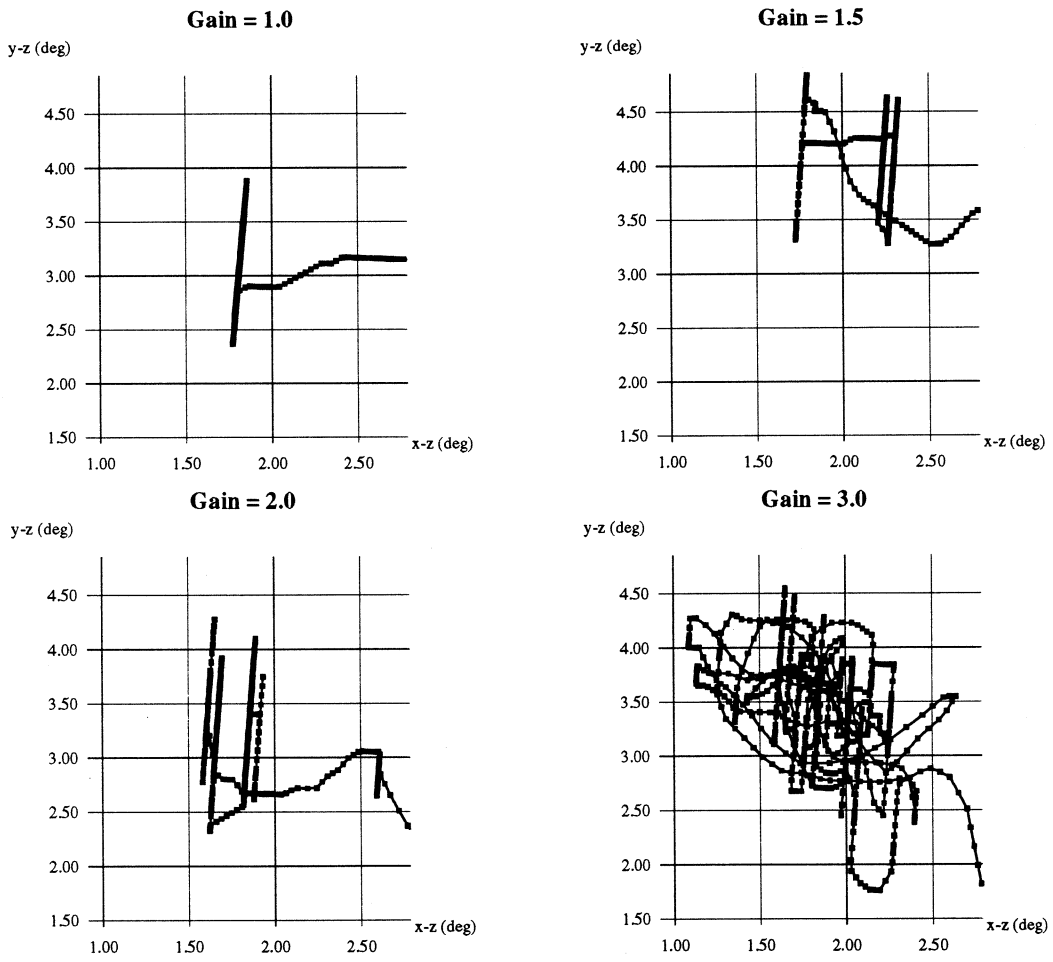

Figure 5. The orientation of the robot end-effector while performing positioning and alignment for varying gain levels. The values shown are degrees of angle with the $X - Z$ and $Y - Z$ planes. The dots are separated by 1/20 second.

observably underdamped after a rotation of 7.1 degrees. Both cameras were then rotated outward. In this case the left and right cameras were rotated 12.0 and 14.5 degrees, respectively, with no observable instability. It was not possible to rotate the cameras further and maintain the target within the field of view.

Next, the right camera was moved toward the left camera to decrease the baseline distance. The initial baseline was 30cm. According to the predictions in Section 4.4, the system should begin to exhibit signs of underdamped behavior with a baseline distance of $l = 30(2.0/3.24) = 18$ cm. Experimentally, at distance of 16 cm the system was observed to become underdamped. The cameras were then moved outward to a baseline of 60 cm with no apparent effect on system behavior.

Perhaps the strongest testament to the calibration insensitivity of the system is the fact that it has been demonstrated dozens of times after placing the cameras by hand and operating the system without first updating the calibration. One reason calibration error does not become a problem is that the camera field of view is a strong constraint on camera position and orientation. Placing the cameras with the robot workspace approximately centered in the image and with a baseline of about 30cm typically orients them within a few degrees of their nominal positions. This level of calibration error is tolerable for most normal operations.

5.3 Three Example Applications

The three applications described in Section 4.2 were implemented and tested to demonstrate the use of skills in realistic situations.

Screwdriver Placement Section 4.2 described the use of an alignment constraint to place a screwdriver onto a screw. A system was constructed to determine the feasibility of this operation. The objects were an unmodified wood screw with a head diameter of 8mm, and a typical screwdriver with its shaft darkened to enhance its trackability. Both the screw and the screwdriver were tracked as parallel edge segments as illustrated in Figure 6 (left). Because of the small size of the objects, the cameras were placed about 50 cm from the objects in question. The baseline was 20 cm. Despite the change in camera configuration,

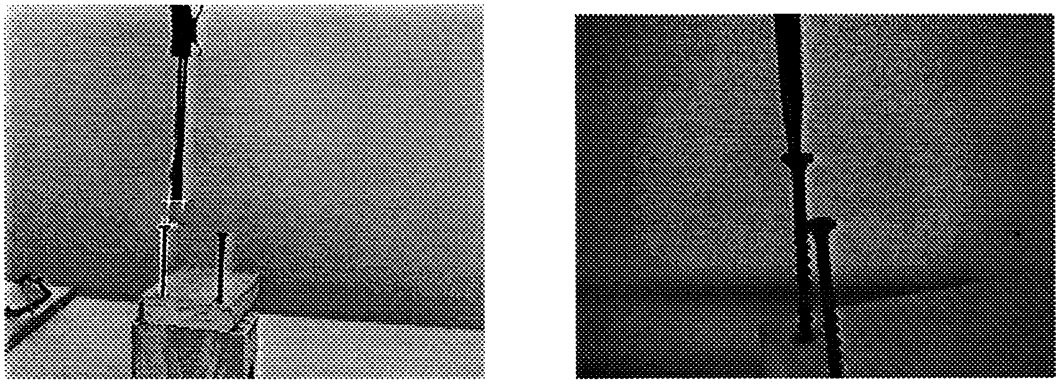

Figure 6. Left, a view of the tracking used to place a screwdriver onto a screw. Right, a close up of the accuracy achieved. The screw in this picture is 8mm in diameter.

the same system calibration was employed. The tracking system ran at 20 Hz without control calculations, and 12 Hz with control calculations.

The screwdriver was placed in an arbitrary position in the robot gripper. Visual servoing was used to first perform an alignment of the screwdriver with the screw. Once aligned, a motion along the calculated alignment axis was superimposed using (13) while maintaining the alignment constraint. The screwdriver was successfully placed near the center of the screwhead in all but a few trials. There was no discernible error in the alignment of the screwdriver with the screw. Figure 6(right) shows the final configuration of one of the experimental runs.

In those cases where the system failed, the failure occurred because the robot executed a corrective rotation just before touching the screw. Due to kinematic errors in the robot, this caused the tip to move slightly just before touching down, and to miss the designated location. These failures could be alleviated by monitoring the alignment error and only moving toward the screw when alignment is sufficiently accurate.

Floppy Disk Insertion The disk tracker and the tracker for parallel lines were combined to perform the insertion of a floppy disk into a disk drive as described in Section 4.2. The experimental configuration and the tracking used to define the setpoint are shown in Figure 7. The cameras were again moved and rotated to provide a better view of the drive slot, but the system calibration was not recomputed. The floppy disk is 2.5mm wide, and the disk

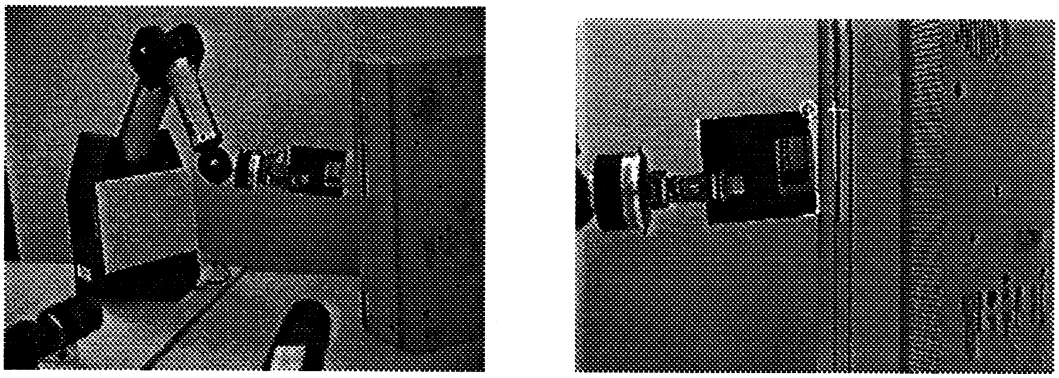

Figure 7. The robot inserting a disk into a disk drive. The slot is about 4mm wide and the disk is 2.5mm wide.

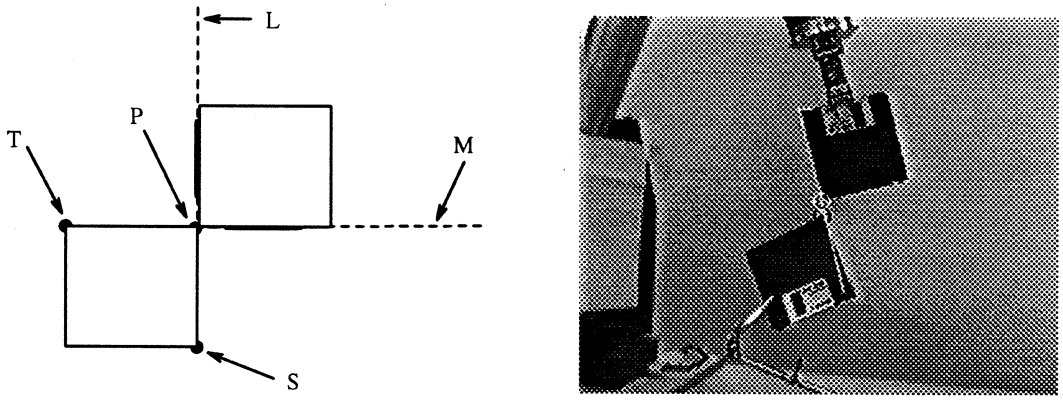

Figure 8. Left, the geometry used to align two floppy disks. Right, a live view from the right camera with the tracking overlaid.

slot is 4mm wide. Over several trials, the system missed the slot only once due to feature mistracking.

Six Degree of Freedom Relative Positioning Three point-to-line regulators were combined to perform full six degree of freedom relative positioning of two floppy disks. The final configuration was defined using three corners of each disk to achieve the configuration pictured in Figure 8. When correctly positioned, the disks should be coplanar, corresponding sides should be parallel, and the disks should touch at the corner. Because the epipolar plane is a singular configuration, the disks were rotated 30 degrees from horizontal. The complete closed-loop system including tracking and control operated at 7Hz.

Experimentally, the accuracy of the placement was found to be somewhat lower than that reported for the previous problems. Typically, orientation was within ± 2 degrees of rotation and positioning was within a few millimeters of the correct value. Most of the lower accuracy can be attributed to the fact that third point used for positioning ($\mathbf{T}$ in Figure 8) was located far from the corners used to define the the second line ($\mathbf{M}$ in Section 8). Thus, small errors in tracking the corners used to define $\mathbf{M}$ and $\mathbf{T}$ were magnified by the problem geometry.

6 Discussion

This article has presented a framework for visual control that is simple, robust, modular, and portable. A particular advantage to the approach is that kinematic constraints and motions can be chosen in the robot task space, yet implemented using image-based feedback methods that are insensitive to system calibration.

The system is extremely accurate. 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, improving the image-processing to be more accurate, or increasing the focal length of the cameras. The current vision processing and control computation system uses no special hardware (other than a standard framegrabber) and could be run on off-the-shelf PC's. 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. At the current rate of progress, frame-rate (60 Hz) servoing will be easily feasible in a year or two.

Clearly, a wider variety of positioning skills must be developed, as well as a richer notion of skill composition. For example, all of the skills described here have focussed on moving points and lines into "contact" with one another. Another natural type of motion is to move "between" two visual obstacles, avoiding contact with either. Similarly, while performing a task, there is often a natural "precedence" between skills. For example, as noted experimentally, the motion to place a screwdriver onto a screw should only take place when the tip of the screwdriver lies along the axis of the screw.

An important open problem at the control level is the development of globally asymptotically stable control methods for visual control. The methods described in this article are locally stable, and perform well away from singularities. However, trajectories that move through a singularity cannot be performed. Early experiments indicate that a switching controller based on a combination of transpose-based and inverse-based controllers works well for motions that cross visual singularities.

The robustness of visual tracking continues to be a major problem. In the experiments described above, the features used were relatively easy to distinguish and were never occluded. These limitations must be overcome before visual servoing is truly practical. Work is proceeding on occlusion detection and compensation. In particular, the design of motion strategies that plan an occlusion-free path offline or online are of interest. Offline vision planning using visibility models and a prior world model information has already been investigated [32, 10]. Online motion compensation based on occlusion detection does not appear to have been considered to date.

Work is also proceeding on extending the framework to more complex task representations. In recent work [13], it was noted that projective invariants [26] provide a basis for specifying robot positions and motion independent of geometric reconstructions, and consequently independent of camera calibration. Development of these concepts is currently underway, including both the visual tracking methods needed to compute projective invariants, and the design and implementation of vision-based motion strategies that employ invariants.

Acknowledgements This research was supported by ARPA grant N00014-93-1-1235, Army DURIP grant DAAH04-95-1-0058, by National Science Foundation grant IRI-9420982, and by funds provided by Yale University. The author would like to thank the anonymous reviewers for many useful comments on an earlier version of this article.

References

[1] P.K. Allen, B. Yoshimi, and A. Timcenko. Hand-eye coordination for robotics 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 Transaction on Robotics and Automation*, 5(6):723–739, 1989.

[3] A. Castano and S. A. Hutchinson. Visual compliance: Task-directed visual servo control. *IEEE Transactions on Robotics and Automation*, 10(3):334–342, June 1994.

[4] F. Chaumette, P. Rives, and B. Espiau. Classification and realization of the different vision-based tasks. In K. Hashimoto, editor, *Visual Servoing*, pages 199–228. World Scientific, 1994.

[5] W.Z. Chen, U.A. Korde, and S.B. Skaar. Position control experiments using vision. *Int. J. of Robot Res.*, 13(3):199–208, June 1994.

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

[7] B. Espiau, F. Chaumette, and P. Rives. A New Approach to Visual Servoing in Robotics. *IEEE Transactions on Robotics and Automation*, 8:313–326, 1992.

[8] O.D. Faugeras. *Three-Dimensional Computer Vision.* MIT Press, Cambridge, MA, 1993.

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

[10] A. Fox and S. Hutchinson. Exploiting visual constraints in the synthesis of uncertainty-tolerant motion plans. *IEEE Transactions on Robotics and Automation*, 11:56–71, 1995.

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

[12] G. D. Hager. Calibration-free visual control using projective invariance. DCS RR-1046, Yale University, New Haven, CT, December 1994. To appear Proc. ICCV '95.

[13] G. D. Hager. Real-time feature tracking and projective invariance as a basis for hand-eye coordination. In *Proc. IEEE Conf. Comp. Vision and Patt. Recog.*, pages 533–539. IEEE Computer Society Press, 1994.

[14] G. D. Hager, W-C. Chang, and A. S. Morse. Robot hand-eye coordination based on stereo vision. *IEEE Control Systems Magazine*, 15(1):30–39, February 1995.

[15] 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.

[16] G.D. Hager and S. Hutchinson, editors. *Proceedings of the Workshop on Visual Servoing.* IEEE, May 1994.

[17] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision: Volume II.* Addison Wesley, 1993.

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

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