15

Visual Servoing: Theory and Applications

Farrokh Janabi-Sharifi

15.1 Introduction

Conventional robotic systems are limited to operating in highly structured environments, and considerable efforts must be expended to compensate for their limited accuracy. This is because conventional robotic manipulators operate on an open kinematic chain for placing and operating a tool (or end-effector) with respect to a workpiece. This is done by joint-to-joint kinematic transformations in order to define the *pose* (position and orientation) of the endpoint with respect to a fixed-world coordinate frame. Similarly, the workpiece must be placed accurately with respect to the same coordinate frame. Any uncertainty or error about the pose of the endpoint or workpiece would lead to task failure. Potential sources (such as gear backlashes, bending of the links, joints slippage, poor fixturing) would contribute to errors in the endpoint or workpiece poses. Therefore, considerable effort and cost are expended to overcome the above issues, for example, to design and manufacture special-purpose end-effectors, jigs, and fixtures. Consequently, due to the need for an accurate world model, the cost of changing the robot task would be quite high.

Alternatively, visual servoing provides direct measurements and control of the robot endpoint with respect to the workpiece and hence does not rely on open-loop kinematic calculations. **Visual servoing** or vision-guided servoing is the use of vision in the feedback loop of the lowest level of a (usually robotic) system control with fast image processing to provide reactive behavior. The task of visual servoing for robotic manipulators (or robotic visual servoing, RVS) is to control the pose of the robot's end-effector relative to either a world coordinate frame or an object being manipulated, using real-time visual features extracted from the image. A camera can be fixed or mounted at the endpoint (eye-in-hand configuration). The advantages of robotic visual servoing can be summarized as follows:

1. It will relax the requirement for the *exact* specification of the workpiece pose. Therefore, it will reduce the costs associated with robot teaching and special-purpose fixtures. For example, it will allow operations on moving or randomly placed workpieces.
2. The requirement for *exact* positioning of the endpoint will be relaxed. Therefore, the robot operation will not highly depend on stiffness and mechanical accuracy of the robot structure so

0-8493-1162-4/03/$0.00+$1.50

the robot mechanisms could be built lighter. This will lead to reduced cost of robot manufacture and operation, and decreased robot cycle time.

In summary, task specifications in a visual servoing framework would support robotic cells that are relatively robust to many disturbing effects in unstructured environments and can adapt readily to minor changes in the task or workpiece without needing to be reprogrammed.

One must distinguish visual servoing from traditional vision integration with robotic systems. Traditional vision-based control systems [Shirai and Inoue, 1973] are typically *look-and-move* structures, where visual sensing and manipulation control are combined in an open-loop fashion. Therefore, image processing and robot control are independent with sequential operations. With look-and-move systems, image-processing times are long (in the order of 0.1 to 1 seconds), and the accuracy depends on the accuracy of the vision system and the robot manipulator. In real visual servoing systems, the control feedback loop is closed around real-time image processing and measurements. The visual feedback loop operates at a high sampling rate in the range of 100 Hz for direct control of the robot endpoint with respect to an object and allows the operation of robot inner joint-servo loops at high sampling rates. Therefore, visual servoing systems provide improved accuracy and robustness to disturbing elements of unstructured environments such as kinematic modeling errors and randomly positioned parts. Visual servoing is a truly mechatronic stream combining results from real-time image processing, kinematics, dynamics, control theory, real-time computation, and information technology.

The fundamentals of system modeling and image projection will be summarized in Section 15.2, the basic classes of visual servoing systems will be presented in Section 15.3, simulation and experimental examples will be provided in Section 15.4, and the applications will be given in Section 15.5. Finally, the chapter will be summarized in Section 15.6.

15.2 Background

As shown in Figure 15.1, the coordinate frames that might be used for visual servoing include coordinate frames attached to the base (world frame), endpoint of the robot, camera, and object. For example, the location of an object viewed by the camera would be calculated with respect to the camera frame, or the location of the object might be determined with respect to the base frame. The homogeneous transforms between these frames are $\overline{T}_B^E, \overline{T}_E^C, \overline{T}_C^O$, and $\overline{T}_B^O$. Here $\overline{T}_j^i$ is the homogenous transform from frame i to frame j, specified by a rotation matrix $\boldsymbol{R}_j^i$ and translation vector $\boldsymbol{T}_j^i$. The camera is usually fixed with respect to the endpoint or the world coordinate frame. Therefore, $\overline{T}_E^C$ or $\overline{T}_B^C$ could be obtained from kinematic calibration tests. A common configuration is eye-in-hand configuration, where the camera is mounted at the endpoint, hence $\overline{T}_E^C = I$, i.e., coordinate frames E and C coincide. This configuration provides better viewing possibilities with less likelihood of viewing obstruction from the moving arm and target.

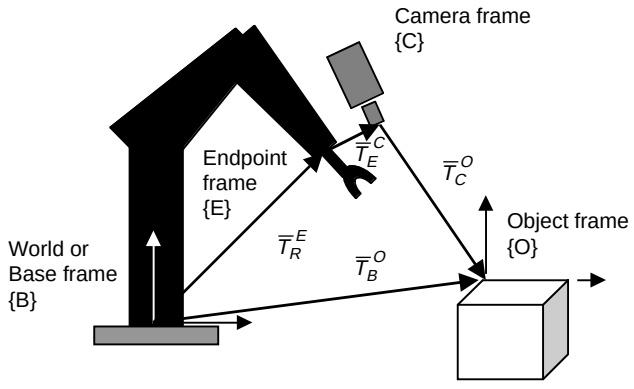

FIGURE 15.1 Relations between different coordinate frames: world, endpoint, camera, and object.

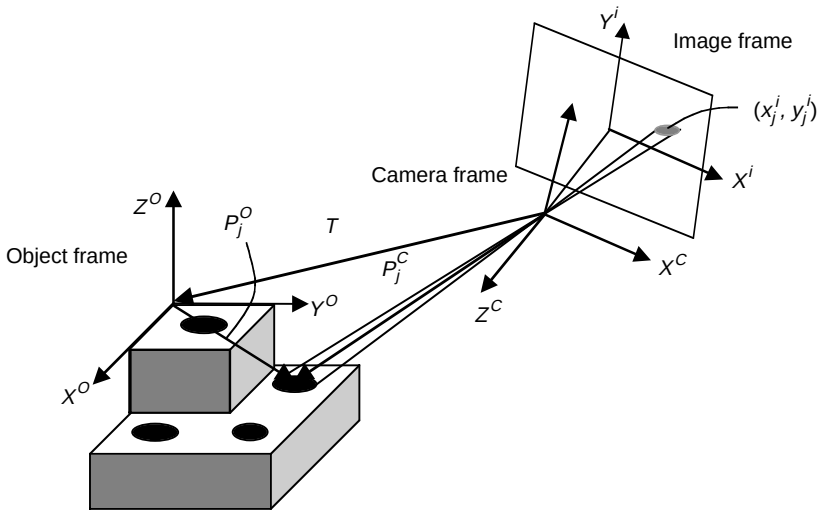

FIGURE 15.2 Projection of an object feature onto the image plane.

For simplicity, in the remainder of this chapter, we will use frames *E* and *C* interchangeably, unless otherwise specified.

Let $\boldsymbol{T} = (X, Y, Z)^T$ denote the relative position vector of the object frame with respect to the camera frame (Figure 16.2). We will also denote $= (\ ,\ ,\ )^T$ as the relative orientation (or viewing direction) vector with roll, pitch, and yaw parameters, respectively. The pose $\boldsymbol{W}$ (position and orientation vector) of the object relative to the robot endpoint (or camera) will be then

$$W = (T,\ )^T = (X, Y, Z,\ ,\ ,\ )^T. \tag{15.1}$$

We can define the relative task space $= \mathrm{SE}^3 = \Re^3$ SO^3 as the set of all possible positions and orientations that could be attained by the end-effector. In general, however, we prefer to represent the relative pose by a 6D vector $\boldsymbol{W}$ $\Re^6$ of rather than by $\boldsymbol{W}$.

The camera-projection model is shown in Figure 15.2. Each camera contains the projection of a three-dimensional scene in its two-dimensional (2D) image plane. Because depth information is lost, additional information will be required to estimate the three-dimensional coordinates of a point $\boldsymbol{P}$. This information could be obtained by multiple views of the object or by knowledge of the geometric relationship between a set of **feature** points $_i$ on the object. This information is obtained from **image feature parameters**. Although different image feature parameters are used in vision, in this chapter we will use the coordinates of image feature points as image feature parameters. Good visual features depend on many parameters such as the feature's radiometric properties, its visibility from many view points, and its ease of extraction without any ambiguity. Feature selection and planning for visual servoing is an important component of any visual servoing system [Janabi-Sharifi and Wilson, 1997] and received a detailed discussion in another chapter (see Chapter 14). In practice, hole and corner features are readily available in many objects and have proven to serve well for many visual servoing tasks [Feddema et al., 1991; Wilson et al., 2000]. Therefore, the rest of this chapter will focus on the use of hole and corner features.

A set of image feature parameters could be chosen to provide information about 6D relative pose vectors at any instance of servoing. Therefore, image feature vector will be defined as $\boldsymbol{s} = [s_1, s_2, \ldots, s_k]^T$ where each s_i $\Re$ is a bounded image feature parameter. Therefore, image feature parameter space (or shortly image space) $\boldsymbol{S}$ is defined as $\boldsymbol{s}$ $\boldsymbol{S}$ $\Re^k$. The projection will be a mapping denoted by

$$G: \quad \boldsymbol{S}. \tag{15.2}$$

For instance, each pair of s_i could be thought of as the coordinates $[x^i, y^i]^T$ of the projection of a feature point $\boldsymbol{P}$ onto the image plane, and then $\boldsymbol{S}$ $\Re^2$. The number of feature points used depends on the servoing strategy, but for 6D relative pose estimation, the image feature coordinates of at least three features will be necessary [Yuan, 1989]. Three projection models used in visual servoing are perspective projection, scaled orthographic projection, and affine projection [Hutchinson et al., 1996]. However, the scaled orthographic projection and affine projection models are approximations of the perspective projection model; therefore, we will adopt the commonly used perspective projection. Here $P_j^c = (X_j^c, Y_j^c, Z_j^c)^T$ and $P_j^o = (X_j^o, Y_j^o, Z_j^o)^T$ are the coordinate vectors of the jth object feature center in the camera and object frames, respectively (Figure 15.2). The feature point can be described in the camera frame using the following transformation:

$$P_j^c = T + R(\ ,\ ,\ )P_j^o \tag{15.3}$$

where the rotation matrix is

$$R(\ ,\ ,\ ) = \begin{matrix} \cos\ \cos & \cos\ \sin\ \sin & \sin\ \cos & \cos\ \sin\ \cos\ + \sin\ \sin \\ \sin\ \cos & \sin\ \sin\ \sin\ + \cos\ \cos & \sin\ \sin\ \cos & \cos\ c\sin \\ \sin & \cos\ \sin & \cos\ \cos & \end{matrix} . \tag{15.4}$$

The position vector of the feature in the object frame, P_j^o, is usually known from the CAD model of the object. The task of visual servoing is to control the pose $\boldsymbol{W}$ using the visual features extracted from the object image. The coordinates of the projection of this feature center on the image plane will be x_j^i, y_j^i, given by Figure 15.2:

$$\begin{matrix} x_j^i \\ y_j^i \end{matrix} = \frac{F}{Z_j^c} \begin{matrix} \frac{X_j^c}{P_X} \\ \frac{Y_j^c}{P_Y} \end{matrix} \tag{15.5}$$

where P_X and P_Y are interpixel spacing in X^i and Y^i axes of the image plane, respectively, and F is the focal length. This model assumes that the origin of the image coordinates is located at the principal point and $|Z_j^c| \gg F$. The perspective projection model requires both **intrinsic** and **extrinsic camera parameters**. The intrinsic camera parameters (P_X, P_Y, F) and coordinates of optical axis on the image plane (principal point) O^i are determined from camera calibration tests. Additionally, camera calibration tests will provide radial distortion parameters (r, K_1, K_2), tangential distortion parameters (P_1, P_2) and aspect ratio [Ficocelli, 1999]. The extrinsic camera parameters include the pose of the camera with respect to the end-effector or the robot base frame. The extrinsic camera parameters are calculated by inspection of the camera housing and kinematic calibration [Corke, 1996]. Excellent solutions to the camera calibration problem exist in the literature [Tsai and Lenz, 1989] (see also Further Information).

15.3 Servoing Structures

Three major classifications of visual servoing structures are *position-based* visual servoing (PBVS), *image-based* visual servoing (IBVS), and *hybrid* visual servoing (HVS). The basic structures of these systems are shown in Figures 15.3 to 15.5. The first two classifications were initially introduced by Sanderson and Weiss [1980]. In all of the structures the image features are extracted and image feature parameters are measured (i.e., mapping G). Processing of the entire image would be time consuming for real-time visual servoing, so windowing techniques are used to process a number of selected features. Windowing methods not only provide computational speed but also reduce the requirement for special-purpose hardware.

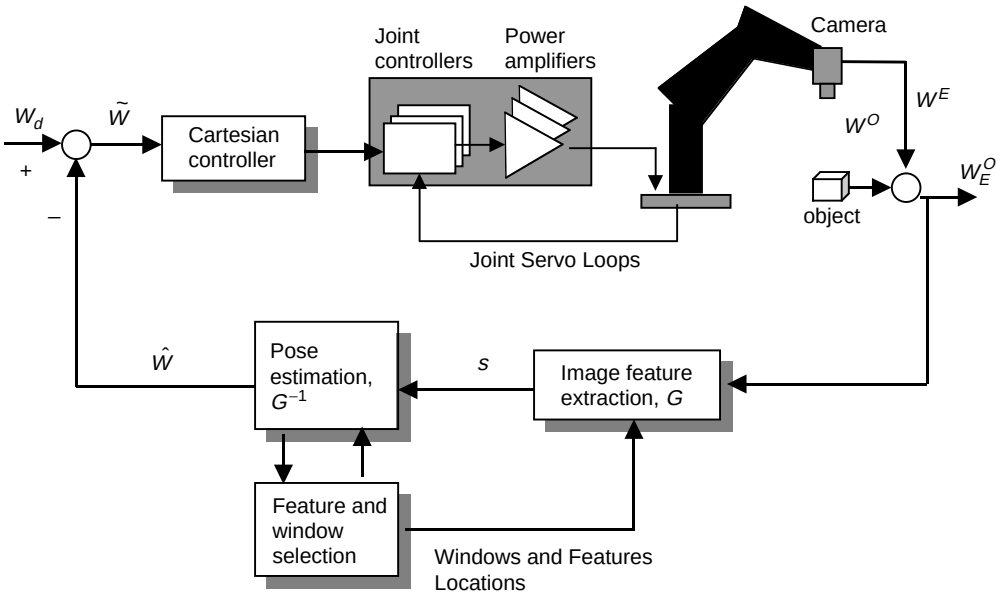

FIGURE 15.3 Structure of position-based visual servoing (PBVS).

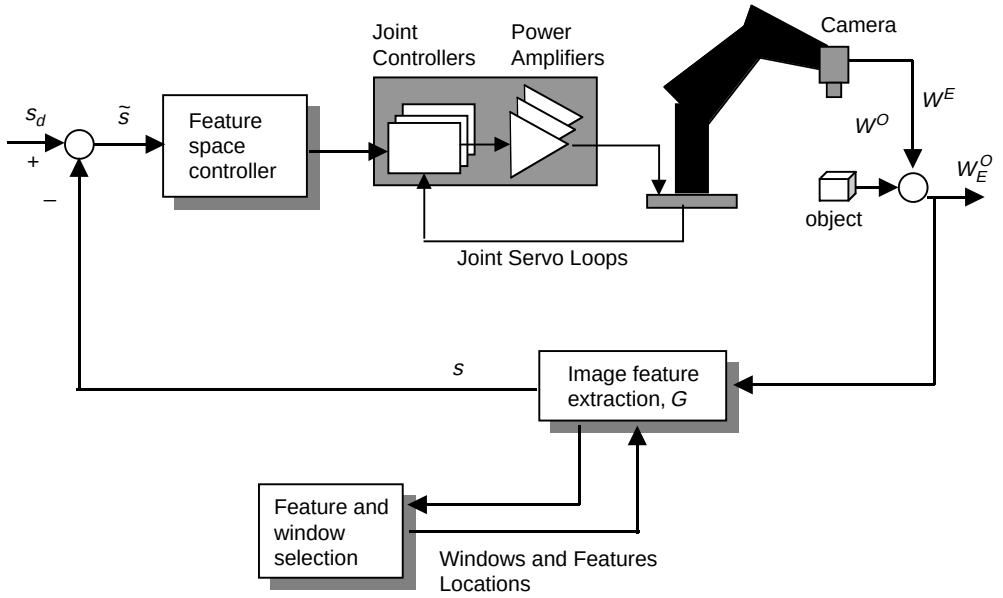

FIGURE 15.4 Structure of image-based visual servoing (IBVS).

Real-time feature extraction and robust image processing are crucial for successful visual servoing and will be discussed in detail in another chapter (see Chapter 10). The feature- and window-selection block in all of the shown structures uses current information about the status of the camera with respect to the object and the models of the camera and environment to prescribe the next time-step features and the locations of the windows associated with those selected features. Feature-selection and planning issues are discussed in another chapter as well (see Chapter 14). In all of the structures, the visual servo controllers

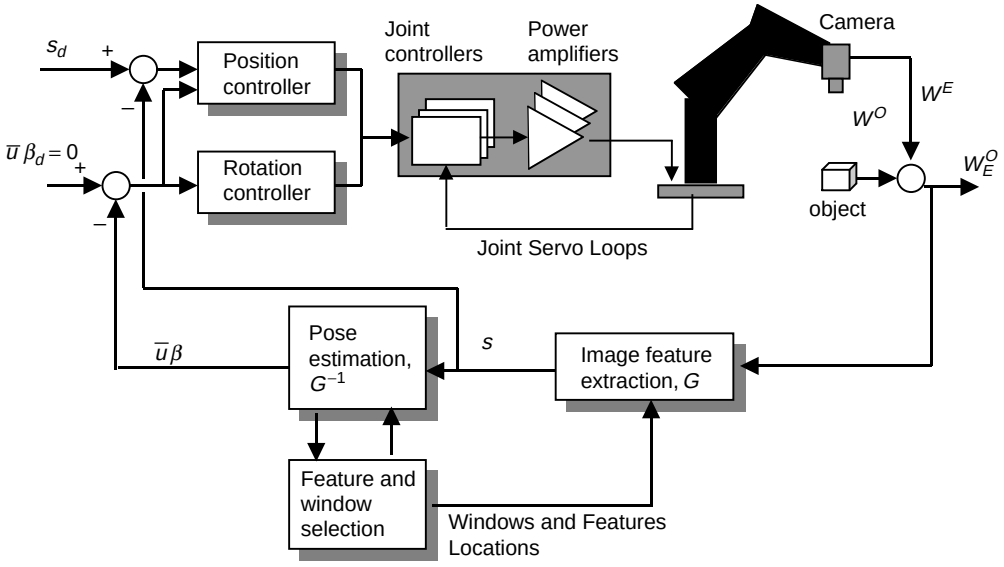

FIGURE 15.5 Hybrid 2-1/2D visual servoing (HVS).

determine set points for the robot joint-servo loops. Because almost any industrial robot has a joint–servo interface, this simplifies visual servo-control integration and portability. Therefore, the internal joint-level feedback loops are inherent to the robot controller, and visual servo-control systems do not need to deal with the complex dynamics and control of the robot joints.

In position-based control (Figure 15.3) the parameters extracted ($\boldsymbol{s}$) are used with the models of camera and object geometry to estimate the relative pose vector ($\hat{\boldsymbol{W}}$) of the object with respect to the end-effector. The estimated pose is compared with the desired relative pose ($\boldsymbol{W}_d$) to calculate the relative pose error ($\tilde{\boldsymbol{W}}$). A Cartesian control law reduces the relative pose error, and the Cartesian control command is transformed to the joint-level commands for the joint-servo loops by appropriate kinematic transformations.

In image-based control (Figure 15.4), the control of the robot is performed directly in the image parameters space. The feature parameters vector extracted ($\boldsymbol{s}$) is compared with the desired feature parameter vector ($\boldsymbol{s}_d$) to determine the feature-space error vector ($\tilde{\boldsymbol{s}}$). This error vector is used by a feature-space control law to generate a Cartesian or joint-level control command.

In hybrid control (Figure 15.5) such as 2-1/2D visual servoing, the pose estimation is partial and determines rotation parameters only. The control input is expressed partially in three-dimensional Cartesian space and in part in two-dimensional image space. An image-based control is used to control the camera translations, while the orientation vector $\overline{u}$ is extracted and used to control the camera rotational degrees of freedom.

Each of the above strategies has its advantages and limitations. Several articles have reported the comparison of the above strategies (see Further Information). In the next sections these methods will be discussed, and simulation results will be provided to show their performance.

15.3.1 Position-Based Visual Servoing (PBVS)

The general structure of a PBVS is shown in Figure 15.3. A PBVS system operates in Cartesian space and allows the direct and natural specification of the desired relative trajectories in the Cartesian space, often used for robotic task specification. Also, by separating the pose-estimation problem from the control-design

problem, the control designer can take advantage of well-established robot Cartesian control algorithms. As will be shown in the Examples section, PBVS provides better response to large translational and rotational camera motions than its counterpart IBVS. PBVS is free of the image singularities, local minima, and camera-retreat problems specific to IBVS. Under certain assumptions, the closed-loop stability of PBVS is robust with respect to bounded errors of the intrinsic camera-calibration and object model. However, PBVS is more sensitive to camera and object model errors than IBVS. PBVS provides no mechanism for regulating the features in the image space. A feature-selection and switching mechanism would be necessary [Janabi-Sharifi and Wilson, 1997]. Because the relative pose must be estimated online, feedback and estimation are more time consuming than IBVS, with accuracy depending on the system-calibration parameters.

Pose estimation is a key issue in PBVS. Close-range photogrammetric techniques have been applied to resolve pose estimation in real-time. The disadvantages of these techniques are their complexity and their dependency on the camera and object models. The task is to find (1) the relative pose of the object relative to the endpoint ($\boldsymbol{W}$) using two-dimensional image coordinates of feature points ($\boldsymbol{s}$) and (2) knowledge about the camera intrinsic parameters and the relationship between the observed feature points (usually from the CAD model of the object). It has been shown that at least three feature points are required to solve for the 6D pose vector [Yuan, 1989]. However, to obtain a unique solution at least four features will be needed.

The existing solutions for the pose-estimation problem can be divided into analytic and least-squares solutions. For instance, unique analytical solutions exist for four coplanar, but not collinear, feature points. If intrinsic camera parameters need to be estimated, six or more feature points will be required for a unique solution (for further details see Further Information). Because the general least-squares solution to pose estimation is a nonlinear optimization problem with no known closed-form solution, some researchers have attempted iterative methods. These methods rely on refining the nominal pose estimation based on real-time observations. For instance, Yuan [1989] reports a general iterative solution to pose estimation independent of the number of features or their distributions. To reduce the noise effect, some sort of smoothing or averaging is usually incorporated.

Extended Kalman filtering (EKF) provides an excellent iterative solution to pose estimation. This approach has been successfully examined for 6D control of the robot endpoint using observations of image coordinates of 4 or more features [Wilson et al., 2000; Wang, 1992]. To adapt to the sudden motions of the object an adaptive Kalman filter estimation has also been formulated recently for 6D pose estimation [Ficocelli and Janabi-Sharifi, 2001]. In comparison to many techniques Kalman-filter-based solutions are less sensitive to small measurement noise. An EKF-based approach also has the following advantages. First, it provides an optimal estimation of the relative pose vector by reducing image-parameter noise. Next, EKF-based state estimations improve solution impunity against the uniqueness problem. Finally, the EKF-based approach provides feature-point locations in the image plane for the next time-step. This allows only small window areas to be processed for image parameter measurements and leads to significant reductions in image-processing time. Therefore, this section provides a brief discussion of EKF-based method that involves the following assumptions and conditions.

First, the target velocity is assumed to be constant during each sample period. This is a reasonably valid assumption for small sample periods in real-time visual servoing. Therefore, the state vector $\boldsymbol{W}$ is extended to include relative velocity as well. That is:

$$W = [X, \dot{X}, Y, \dot{Y}, Z, \dot{Z}, \ , \dot{\ }, \ , \dot{\ }, \ , \dot{\ }]^T. \tag{15.6}$$

A discrete dynamic model will then be:

$$W_k = AW_{k\ 1} + \ _k \tag{15.7}$$

with diagonal ***A*** matrix defined as follows:

$$\mathrm{A} = \begin{bmatrix} 1 & T & & & \\ 0 & 1 & & & \\ & & \ddots & & \\ & & & 1 & T \\ & & & 0 & 1 \end{bmatrix}_{12\ 12} \tag{15.8}$$

where T is the sample period, k is the sample step, and $_k$ denotes the dynamic model disturbance noise vector described by a zero mean Gaussian distribution with covariance Q_k.

The output model will be based on the projection model given by Eqs. (15.3) to (15.5) and defines the image feature locations in terms of the state vector $\boldsymbol{W}_k$ as follows:

$$\mathrm{Z}_k = G(W_k) + \ _k \tag{15.9}$$

with

$$\boldsymbol{Z}_k = [x_1^i, y_1^i, x_2^i, y_2^i, \ldots, x_p^i, y_p^i]^T \tag{15.10}$$

and

$$G(W_k) = \ F\ \frac{X_1^c}{P_X Z_1^c},\ \frac{Y_1^c}{P_Y Z_1^c}, \ldots, \frac{X_p^c}{P_X Z_p^c},\ \frac{Y_p^c}{P_Y Z_p^c}^{\ T} \tag{15.11}$$

for p features. Here, X_j^c, Y_j^c, and Z_j^c are given by Eqs. (15.3) and (15.4). $_k$ denotes the image parameter measurement noise that is assumed to be described by a zero mean Gaussian distribution with covariance $\boldsymbol{R}_k$.

The recursive EKF algorithm consists of two major parts, one for prediction and the other for updating, as follows.

Prediction:

$$\hat{W}_{k,k\ 1} = A\hat{W}_{k\ 1,k\ 1} \tag{15.12}$$

$$P_{k,k\ 1} = AP_{k\ 1,k\ 1}A^T + Q_{k\ 1} \tag{15.13}$$

Linearization:

$$H_k = \left.\frac{\partial G(W)}{\partial W}\right|_{W=\hat{W}_{k,k\ \mathbf{1}}} \tag{15.14}$$

Kalman gain:

$$K = P_{k,k\ 1}H_k^T(R_k + H_k P_{k,k\ 1}H_k^T)^{\ 1} \tag{15.15}$$

Estimation updates:

$$\hat{W}_{k,k} = \hat{W}_{k,k\ 1} + K(Z_k \quad G(\hat{W}_{k,k\ 1})) \tag{15.16}$$

$$P_{k,k} = P_{k,k\ 1} \quad KH_k P_{k,k\ 1} \tag{15.17}$$

In the above equations, $\hat{W}_{k,k\ 1}$ denotes the state predictions at time k based on measurements at time $k-1$. Also, $\hat{W}_{k,k}$ is the optimal state estimation at time k based on the measurements at time k. $\boldsymbol{P}_{k,k\ 1}$ and $\boldsymbol{P}_{k,k}$ are state prediction error and state estimation error covariance matrices, respectively. $\boldsymbol{K}$ is the Kalman gain matrix. As mentioned above, the number of features used (p) should be above four to obtain a unique solution to 6D pose-vector estimation. However, inclusion of more features will improve the performance of an EKF-based estimation, with a concomitant increase in cost for additional computations. It has been shown that the inclusion of more than six features will not improve the performance of EKF estimation significantly [Wang, 1992]. Also, the features need to be noncollinear and noncoplanar to provide good results. Consequently, 4 p 6 [Janabi-Sharifi and Wilson, 1997]. Further discussion on feature-selection issues is provided in Chapter 14.

The vision-based control system consists of fast inner joint-servo control loops. The slower outer loop is a visual servo control loop. The regulation aim in position-based control is to design a controller that computes joint-angle changes to move the robot such that the relative pose reaches the desired relative pose in an acceptable manner. Therefore, control design for visual servo-control loops requires calculation of the error vector or the control command in joint space to provide the input commands to the joint-servo loops (Figure 15.3). For this purpose first the Euler angles must be converted to the total rotation angles with respect to the endpoint frame [Wilson et al., 1996]. Assuming a slow-varying endpoint and object frames within a sample period, the total rotation angles could be obtained from:

$$\begin{matrix} & _X & & \sin + & \cos & \cos \\ = & _Y & & \cos + & \sin & \cos \\ & _Z & & & \sin & \end{matrix} \tag{15.18}$$

The orientation vector and relative position vector $\boldsymbol{T}$ are compared with the input reference trajectory point to determine the endpoint relative-pose error in Cartesian space. Also, assuming slow-varying motion of the system without commanding any large abrupt changes, the endpoint relative position and orientation changes, e.g., error equations required for controlling the robot endpoint, are related to the joint changes via:

$$q \quad J^{\ 1} \begin{matrix} R_B^E & 0 & T_E \\ 0 & R_B^E & _E \end{matrix} \tag{15.19}$$

One can regulate the joint level error $\boldsymbol{q}$ or Cartesian error $\boldsymbol{W_E} = [\ \boldsymbol{T_E},\ _E]^T$. Different control laws have been examined in the literature, e.g., PD control law to regulate $\boldsymbol{q}$ [Wilson, et al., 1996]. Note that the inner joint–servo control loops of common industrial robots operate at high sample rates, and for smooth tracking performance high sample rates such as 60 Hz are usually expected from the visual servo control loop (i.e., outer loop). Due to the load of computations involved in PBVS, distributed computing architectures have been proposed to provide reasonable sample rates for the visual servo-control loop. For instance, Wilson et al. [1996] have reported a PBVS design and implementation using a transputer-based architecture with a coordinating PC. However, recent advances in microprocessor technology have made high-speed PC-based implementation possible as well [Ficocelli and Janabi-Sharifi, 2001].

15.3.2 Image-Based Visual Servoing (IBVS)

In image-based visual servoing (Figure 15.4) the error signal and control command are calculated in the image space. The task of the control is to minimize the error of the feature-parameter vector, given by $\dot{s} = s_d$ s. An example of a visual task specified in the image space is shown in Figure 15.6. It shows the initial and desired views of an object with five hole features. The advantage of IBVS is that it does not require full pose estimation and hence is computationally less involved than PBVS. Also, it is claimed

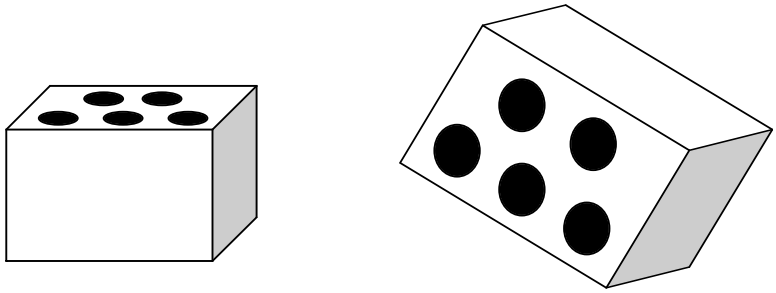

FIGURE 15.6 An example of a task specified in the image space with the initial and desired views.

that the positioning accuracy of IBVS is less sensitive to camera-calibration errors than PBVS. However, IBVS would lead to image singularities that might cause control instabilities. Another issue with IBVS is the camera-retreat problem: For the commanded pure rotations around the optical axis, the camera often moves away from the target in a normal direction and then returns. Moreover, the convergence of IBVS is ensured only in a small neighborhood of the desired camera pose. The domain of this neighborhood is analytically impossible to determine [Chaumette, 1998]. The closed-loop stability is also robust with respect to the errors of the camera-calibration and target model.

The goal of IBVS control is to find appropriate endpoint pose changes (or velocity) required to minimize error in the image space defined by $\tilde{s} = s_d \quad s$ (or its time rate). Because the robot control is usually in either Cartesian or joint space, the resultant control in image space must be converted into the Cartesian- or joint-space error. The velocity (or differential changes) of the camera $\dot{r}$ (or $\boldsymbol{r}$) or its relative pose $\dot{W}$ (or $\boldsymbol{W}$) can be related to the image feature velocities $\dot{s}$ (or $\boldsymbol{s}$) by a differential Jacobian matrix, $\boldsymbol{J_i}$, called **image Jacobian**. This matrix is also referred to as feature Jacobian matrix, feature sensitivity matrix, interaction matrix, or $\boldsymbol{B}$ matrix. Because we have assumed a hand-eye configuration, the endpoint frame (E) is coincident with the camera frame (C). Therefore, we will use endpoint and camera frames or poses interchangeably. Then, $\boldsymbol{J_i} \quad \Re^{k \ m}$ is a linear transformation from the tangent space of at $\boldsymbol{W}$ (or $\boldsymbol{r}$) to the tangent space of $\boldsymbol{S}$ at $\boldsymbol{s}$. Here, k and m are the dimensions of the image and task spaces, respectively. Depending on the choice of endpoint pose representation, there are different derivations for the image Jacobian matrix (see Further Information). For instance, if the relative pose of the object with respect to the endpoint is considered, we will have

$$s = J_i \quad W_E \tag{15.20}$$

where $\boldsymbol{W_E}$ is the differential change of the relative pose of the object defined in the endpoint (or camera) frame. The corresponding image Jacobian matrix is given in Feddema et al. [1991]. It is also possible to relate the motion of a task frame to the motion of the feature points. In hand-eye configurations, it is preferable to define endpoint (or camera) motions with respect to the endpoint (or camera) frame. Therefore, $\boldsymbol{W_E} = [\ \boldsymbol{T_E}, \quad _{\boldsymbol{E}}]^T$ in Eq. (15.20) will imply the differential corrective motion of the endpoint with respect to the endpoint frame. The image Jacobian matrix for one feature can then be written as:

$$J_i = \begin{matrix} \dfrac{F}{P_X Z^c} & 0 & \dfrac{x^i}{Z^c} & \dfrac{P_Y}{F} x^i y^i & \dfrac{F}{P_X} + \dfrac{P_X}{F} x^{i^2} & \dfrac{P_Y}{P_X} y^i \\ 0 & \dfrac{F}{P_Y Z^c} & \dfrac{y^i}{Z^c} & \dfrac{F}{P_Y} + \dfrac{P_Y}{F} y^{i^2} & \dfrac{P_X}{F} x^i y^i & \dfrac{P_X}{P_Y} x^i \end{matrix} . \tag{15.21}$$

Because several features (p) are used for 6D visual servoing, the image Jacobian matrix will have the form of:

$$J_i = \begin{matrix} (J_i)_1 \\ (J_i)_2 \\ \vdots \\ (J_i)_p \end{matrix} \tag{15.22}$$

where $(J_i)_k$ is a 2 6 Jacobian matrix for each feature given by Eq. (15.21). Therefore, one can use Eqs. (15.20) and (15.22) to calculate endpoint differential motion (or its velocity) for a given change or error expressed in the image space. That is:

$$W_E = J_i^{\mathbf{1}} \ s, \quad \text{or} \quad \dot{W}_E = J_i^{\mathbf{1}} \dot{s} \tag{15.23}$$

which assumes a nonsingular and square image Jacobian matrix. A control law can then be applied to move the robot endpoint (or camera) toward the desired feature vector. The earliest and easiest control approach is resolved-rate motion control (RRMC), which uses a proportional control $\dot{\tilde{s}} = K_p \tilde{s}$ to warrant exponential convergence $\tilde{s}$ **0**. Therefore:

$$u = K_p J_i^{\mathbf{1}} \tilde{s} \tag{15.24}$$

where $\boldsymbol{K_p}$ is a diagonal gain matrix [Feddema et al., 1991] and $\boldsymbol{u}$ is the endpoint velocity screw. At this stage a Cartesian-space controller can be applied. Because the control output will indicate endpoint changes (or velocity) with respect to the endpoint frame, transformations must be applied to calculate the equivalent joint angles changes. Under the same assumptions as Eqs. (15.19), these calculations are given by:

$$q \quad J^{1} \begin{matrix} R_B^E & \mathbf{0} \\ \mathbf{0} & R_B^E \end{matrix} K_P J_i^{\mathbf{1}} \tilde{s} \, . \tag{15.25}$$

RRMC is a simple and easy-to-implement method with a fast response time; however, it is not prone to singularity and provides highly coupled translational and rotational motions of the camera. In addition to RRMC, other approaches to IBVS exist in the literature such as optimal control techniques, model reference adaptive control (MRAC) methods, etc. (see Corke [1996]). Close examination of Eq. (15.25) reveals several problems with IBVS.

First, the image Jacobian matrix, given by Eqs. (15.21) and (15.22) depends on the depth Z^c of the feature. For a fixed camera, when the end-effector or the object held in the end-effector is tracked as the object, this is not an issue because depth can be estimated using robot forward kinematics and camera-calibration data. For an eye-in-hand configuration, however, the depth or image Jacobian matrix must be estimated during servoing. Conventional estimation techniques can be applied to provide depth estimation; however, they increase computation time. Adaptive techniques for online depth estimation can be applied, with limited success. Some proposals have been made to use an approximation to the image Jacobian matrix using the value of the image Jacobian matrix or the desired features depth computed at the desired camera position. This solution avoids local minima and online updating of the image Jacobian matrix. However, the image trajectories might be unpredictable and would leave image boundaries. Further sources are provided in Further Information.

Second, the inverse of the image Jacobian matrix in Eq. (15.25) might not be square and full-rank (nonsingular). Therefore, pseudo-inverse solutions must be sought. Because p features are used for servoing, the dimension of the image space is $k = 2p$. Two possibilities exist when the image Jacobian matrix is full rank, i.e., rank($\boldsymbol{j_i}$) = min($2p$, m) but $2p \pi m$. If $2p < m$, the pseudo-inverse Jacobian and least-squares

solution for $\dot{W}_E$ (or $\boldsymbol{W}_E$) will be:

$$\dot{W}_{\mathbf{E}} = J_i^{+}\dot{s} + (I \quad J_i^{+}J_i)v \tag{15.26}$$

$$J_i^{+} = (J_i^T J_i)^{\ 1} J_i^T \tag{15.27}$$

with $\boldsymbol{v}$ as an arbitrary vector such that $(I \quad J_i^{+}J_i)v$ lies in the null space of $\boldsymbol{J_i}$ and allows endpoint motions without changes in the object features velocity. If $2p > m$, the pseudo-inverse Jacobian and least-squares solution for $\dot{W}_E$ (or $\boldsymbol{W}_E$) will be:

$$\dot{W}_E = J_i^{+}\dot{s} \tag{15.28}$$

$$J_i^{+} = (J_i^T J_i)^{\ 1} J_i^T . \tag{15.29}$$

When the image Jacobian matrix is not full rank, the singularity problem must be dealt with. Small velocities of the image features near image singularities would lead to large endpoint velocities and result in control instabilities and task failure. This problem might be treated by singular value decomposition (SVD) approaches or via damped least-squares inverse solutions.

Finally, IBVS introduces high couplings between translational and rotational motions of the camera, leading to a typical problem of camera retreat. This problem has been resolved by the hybrid 2-1/2D approach, which will be introduced in the next section. A simple strategy for resolving the camera-retreat problem has also been introduced in Corke and Hutchinson [2000].

15.3.3 Hybrid Visual Servoing (HVS)

The advantages of both PBVS and IBVS are combined in recent hybrid approaches to visual servoing. Hybrid methods decouple control of certain degrees of freedom; for example, camera rotational degrees could be controlled by IBVS. These methods generally rely on the decomposition of an image Jacobian matrix. A homography matrix is considered to relate camera configurations corresponding to the initial and desired images. This matrix can be computed by a set of corresponding points in the initial and desired images. It is possible to decompose the homography matrix into rotational and translational components. Among hybrid approaches, 2-1/2D visual servoing is well established from an analytical point of view. Therefore, the rest of this section will be devoted to the analysis and discussion of 2-1/2D visual servoing. Further details on other approaches can be found in the sources listed in Further Information.

In 2-1/2D visual servoing [Malis et al., 1999] the controls of camera rotational and translational degrees of freedom are decoupled. That is, the control input is expressed in part in three-dimensional Cartesian space and in part in two-dimensional image space (Figure 15.5). IBVS is used to control the camera translational degrees of freedom while the orientation vector $\overline{u}$ is extracted and controlled by a Cartesian-type controller. This approach provides several advantages. First, since camera rotation and translation controls are decoupled, the problem of camera retreat is solved. Second, 2-1/2D HVS is free of image singularities and local minima. Third, this method does not require full pose estimation, and it is even possible to release the requirement for the geometric model of the object. Fourth, Cartesian camera motion and image plane trajectory can be controlled simultaneously. Finally, this method can accommodate large translational and rotational camera motions. However, in comparison with PBVS and IBVS, 2-1/2D HVS introduces some disadvantages that will be discussed after the introduction of the methodology, as follows.

For 2-1/2D HVS, first a pose estimation technique (e.g., scaled Euclidean reconstruction algorithm, SER [Malis et al., 1999]) is utilized to recover the relative pose between the current and the desired camera frames, and then the $\overline{u}$ vector is extracted. A rotation matrix $\boldsymbol{R}$ exists between the endpoint frame (or the camera frame) denoted by E and the desired endpoint frame by E . This matrix must reach identity matrix at the destination (Figure 15.7). In order to avoid workspace singularities, the rotation matrix will be

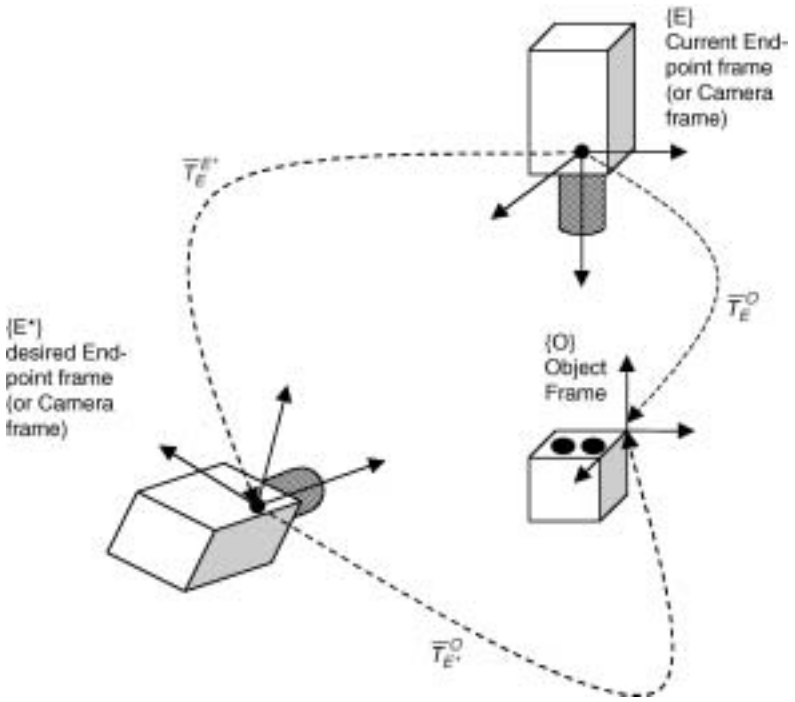

FIGURE 15.7 Modeling of the endpoint (or camera) displacement in a hand-eye configuration for 2-1/2D HVS.

represented by a vector $\bar{u}$ (a unit vector of rotation axis) and a rotation angle . For a 3 3 rotation matrix with elements r_{ij}, the $\bar{u}$ vector could be obtained from:

$$\bar{u} = \frac{1}{2\,\text{sinc}} \begin{matrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{matrix} \quad \text{for} \quad \pi \tag{15.30}$$

and

$$\text{sinc} = \begin{matrix} 1 & \text{if} \quad = 0 \\ \dfrac{\sin}{} & \text{otherwise} \end{matrix} \tag{15.31}$$

where = can be detected through the estimation of . Then the axis of rotation $\bar{u}$ will be given by the eigenvector associated with the eigenvalue of 1 of the rotation matrix. The endpoint velocity screw can be related to the time derivative of the $\bar{u}$ vector by a Jacobian matrix $\boldsymbol{L}$, given by:

$$L = I - \frac{}{2} S(\bar{u}) + \left(1 - \frac{\text{sinc}(\ )}{\text{sinc}^2(\frac{}{2})}\right) S^2(\bar{u}) \tag{15.32}$$

where $S(.)$ is the skew-symmetric matrix associated with u and $\boldsymbol{I}$ is an identity matrix. Malis et al. [1999] have shown that $\boldsymbol{L}$ is singularity free.

Next, a hybrid error vector (Figure 15.5) is defined as:

$$e = \begin{matrix} \mathrm{m}_e & \mathrm{m}_e^* \\ \bar{\mathrm{u}} & \mathbf{0} \end{matrix} \tag{15.33}$$

where m_e and m_e^* are current and desired extended image parameter vectors, respectively. Consider a reference feature point $\boldsymbol{P}$ with coordinates $[X^E, Y^E, Z^E]$ with respect to the endpoint (or camera) frame. A supplementary normalized coordinate could be defined as $z = \log Z^E$ with Z^E as depth. Then, the normalized and extended image vector will be:

$$\mathrm{m}_e = \begin{matrix} x^i \\ y^i \\ z^i \end{matrix} = \begin{matrix} \dfrac{X^E}{Z^E} \\ \dfrac{Y^E}{Z^E} \\ \log Z^E \end{matrix} \tag{15.34}$$

Let $\dot{W}_E = [v, \ ]^T \int [\dot{T}_E, \dot{\ }_E]$ represent the endpoint (or camera) velocity screw with respect to the endpoint frame. The velocity screw $\dot{W}_E$ could be related to the hybrid error velocity $\dot{e}$ by:

$$\dot{e} = \begin{matrix} \dot{m}_e \\ \dfrac{d(\bar{u}\ )}{dt} \end{matrix} = L\dot{W}_E \tag{15.35}$$

where

$$L \int \begin{matrix} L_v & L_{(v,\ )} \\ 0 & L \end{matrix} . \tag{15.36}$$

Here, $\boldsymbol{L}$ is the hybrid Jacobian matrix, with $\boldsymbol{L}$ given by Eq. (15.32),

$$L_v \int \begin{matrix} \dfrac{1}{Z^E} & 0 & \dfrac{X^E}{Z^{E^2}} \\ 0 & \dfrac{1}{Z^E} & \dfrac{Y^E}{Z^{E^2}} \\ 0 & 0 & \dfrac{1}{Z^E} \end{matrix} \tag{15.37}$$

and

$$\mathrm{L}_{(v,\ )} \int \begin{matrix} x^i & (1+ x^{i2}) & y^i \\ 1+ y^{i2} & x^i y^i & x^i \\ y^i & x^i & 0 \end{matrix} . \tag{15.38}$$

Obviously, one can consider differential changes of the endpoint (or camera) pose and error instead of their velocities. Note that Jacobian matrix $\boldsymbol{L}$ is singular only when $Z^E = 0$, or $\frac{1}{Z^E} = 0$, or with $\ = \pm 2$. These cases are exterior to the task space , so the task space is free of image-induced singularities.

Finally, the exponential convergence of $\boldsymbol{e}$ $\mathbf{0}$ is achieved by imposing a control law of the form:

$$\dot{e} = K_p e \tag{15.39}$$

or

$$\dot{W}_E = K_p L^{-1} e \tag{15.40}$$

which could be simplified to:

$$\dot{W}_E = K_p \begin{bmatrix} L_v^{-1} & L_v^{-1} L_{(v,\ )} \\ 0 & I \end{bmatrix} e \tag{15.41}$$

Note that $L^{-1} = I_{3\ 3}$ because $L^{-1}\bar{u} = \bar{u}$, as proven by Malis et al. [1999]. Also, because $\boldsymbol{L}^{-1}$ is an upper triangular matrix, the rotational control loop is decoupled from the translational one. Like Eq. (15.25), endpoint screw velocity can be expressed in terms of changes required in joint angles that will be sent to the robot joint-servo loops for the execution.

Despite numerous advantages there are a few problems associated with 2-1/2D HVS. One of the problems is the possibility of features leaving image boundaries. Some approaches have been proposed to treat this problem. Among them is the approach of Morel et al. who use a modified feature vector (see Further Information).

The second problem is related to noise sensitivity and computational expense of partial estimation in HVS. A scaled Euclidean reconstruction (SER) was originally used to estimate camera displacement between the current and desired relative poses [Malis et al., 1999]. This method does not require a geometric three-dimensional model of the object; however, SER uses recursive KF to extract the rotation matrix from the homography matrix, while KF might be sensitive to camera calibration. Also, the estimation of the homography matrix requires more feature points, especially with noncoplanar objects. This is because when the object is noncoplanar, the estimation problem becomes nonlinear and will necessitate at least eight features for homography matrix estimation at the video rate of 25 Hz. Another approach would be to estimate the full pose of the object with respect to the camera by a fast and globally convergent method such as the orthogonal iteration algorithm (OI) of Lu and Hager (see Further Information). Next, a homogeneous transformation could be applied to obtain camera-frame displacement. However, this would require a full three-dimensional model of the object.

Finally, the selection of reference feature point affects the performance of 2-1/2D HVS. A series of experiments would be required to select the best reference point for the improved performance.

15.4 Examples

Simulations and experiments were run to compare the performances of PBVS, IBVS, and HVS. A MATLAB™ environment was created to simulate a five-degrees-of-freedom CRS Plus SRS-M1A robot with an EG&G Reticon MC9000 CCD camera mounted at its endpoint (Figure 15.8). A 16-mm lens was used. Simulation parameters are shown in Table 15.1. As shown in Figure 15.8, an object with five noncoplanar hole features was considered. The initial poses of the object and robot endpoint are also shown in Table 15.1.

A number of tests were run to investigate the effect of different parameters and compare the performances of different visual servoing methods. The control methods used for each strategy are EKF-based pose estimation and PD control for PBVS, RMRC for IBVS, and 2-1/2D for HVS. The control gains were tuned by running a few simulations for stable and fast responses. The results are shown in Figures 15.9 to 15.12. For simulations, an additive noise has been considered to represent more realistic situations. The general relative motion is shown in Figures 15.9 to 15.11 for PBVS, IBVS, and HVS. Initially, the robot endpoint is located above the object, i.e., $W_E = [0,\ 0,\ 30,\ 0,\ 0,\ 0]^T$ in cm and rad, from Table 15.1.

TABLE 15.1 System Parameters for Simulations and Experiments

Parameter	Value
Focal length, F	1.723 cm
Interpixel spacings, P_x, P_y	0.006 cm/pixel
Number of pixels, N_x, N_y	128
Features coordinates in frame {O}	(5, 5, 0), (5, 5, 0), (5, 5, 0), (5, 5, 0), (0, 0, 3) cm
Initial pose of object with respect to the base frame, W_B^{∞}	[33.02, 2.64, 0.38, 0.0554, 0.0298, 3.1409] (in cm and rad)
Initial pose of the camera/endpoint with respect to the base frame, $\boldsymbol{W}_{\boldsymbol{B}}^{\boldsymbol{E}}$	[33.02, 2.64, 30.38, 0.0554, 0.0298, 3.1409] (in cm and rad)
Initial relative depth, Z_E^O	30 cm
Output measurement covariance matrix of EKF, $\boldsymbol{R}_k$	diag[0.04, 0.01, 0.04, 0.01, 0.04, 0.01, 0.04, 0.01, 0.04, 0.01] (pixel^2)
Disturbance noise covariance matrix of EKF, $\boldsymbol{Q}_k$	diag[0, 0.8, 0, 0.8, 0, 0.8, 0, 0.8, 0, 0.001] in (cm/s)^2 and (deg/sec)^2

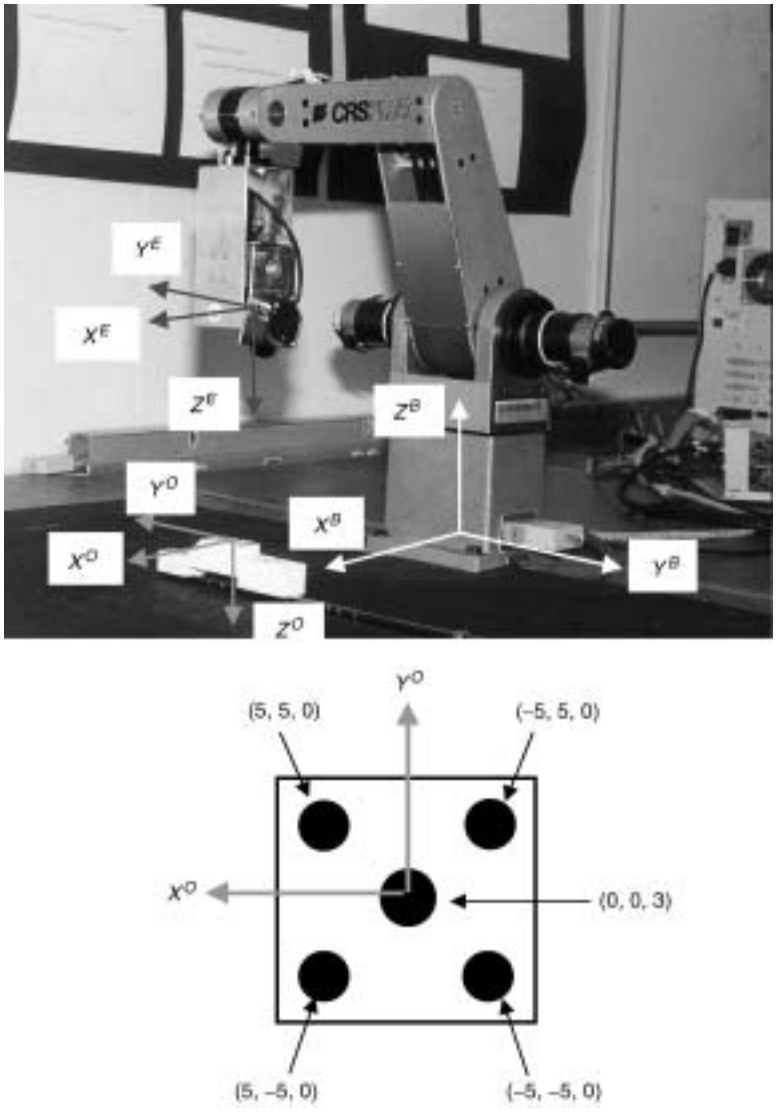

FIGURE 15.8 Simulation environment with object model and its five features.

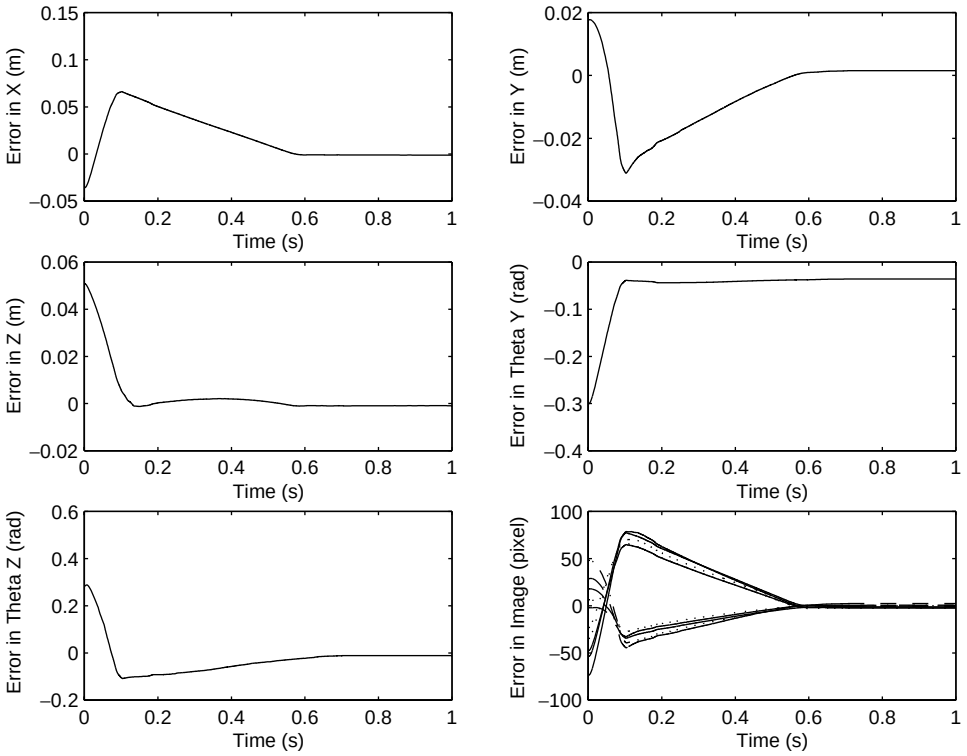

FIGURE 15.9 Simulation results for PBVS: desired relative pose of the object with respect to the endpoint frame: (3, 3, 25 cm; 0, 0.3, 0.3 rad). The relative pose errors are specified with respect to the endpoint frame {E}.

The figures show the relative pose errors for (X, Y, Z, $_X$, $_Y$, $_Z$), i.e., the current relative pose minus the desired relative pose, in the endpoint frame. The endpoint desired relative motion is indicated in the caption of each figure. That corresponds to (3, 3, 5 cm, 0, 0.3, 0.3 rad) translation and rotation along and around the X, Y, and Z axes of the base frame, respectively, for the stationary object. Because the robot had five degrees of freedom, the commanded rotation about X axis was set to zero. Therefore, no error of $_X$ is shown. Also, errors in the image plane for the X and Y positions of five feature points (in pixels) are shown in Figures 15.9 and 15.11. Figure 15.12 shows the camera-retreat problem for IBVS. Only a pure rotation around the optical Z axis of the camera was requested, but the camera moved away from the object and then returned. This problem was observed in neither PBVS nor HVS.

Some experiments were run for a different number of features and with different initial conditions and relative poses. The joint couplings in the CRS robot had negative effects on the control responses for three-dimensional relative motions. The tests showed that a reasonably larger number of features tended to improve the system response. The steady-state errors increased for all the methods with large commanded relative motions. This applied particularly to the IBVS that used a nondecoupled image Jacobian matrix leading to coupled translational and rotational motions. When the commanded motions were very close to the object, the steady-state error increased. Moreover, it was observed that PBVS and IBVS had almost the same response speed; however, HVS demonstrated a bit faster response than IBVS and PBVS, mainly due to the incorporation of the fast OI algorithm for estimation, instead of the SER used by Malis et al. [1999]. Choosing different reference points for HVS apparently had minimal effects on system performance. Finally, joint couplings of the CRS robot had considerable effect on the control responses for three-dimensional relative motions.

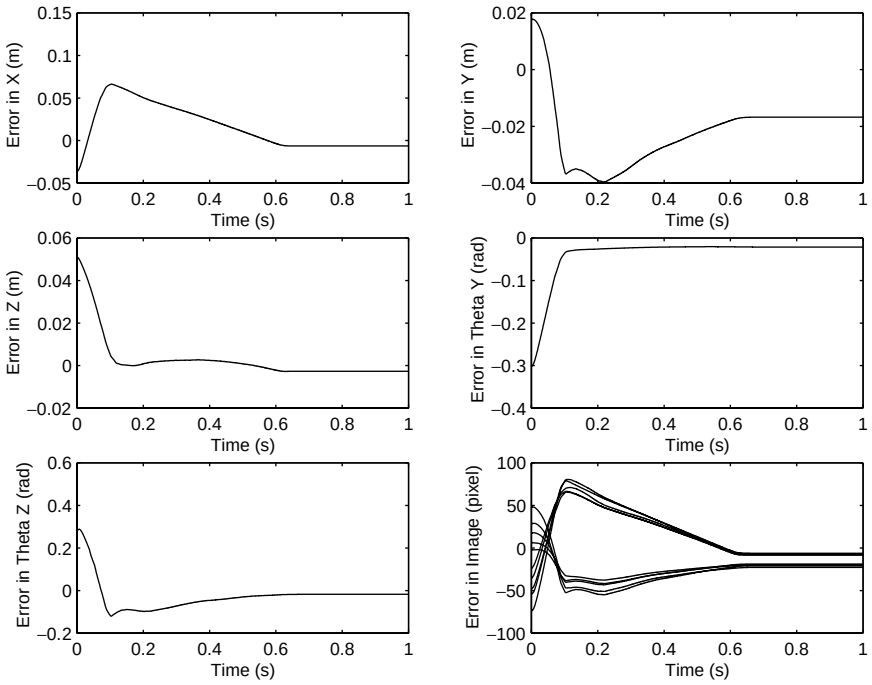

FIGURE 15.10 Simulation results for IBVS: desired relative pose of the object with respect to the endpoint frame: (3, 3, 25 cm; 0, 0.3, 0.3 rad). The relative pose errors are specified with respect to the endpoint frame {E}.

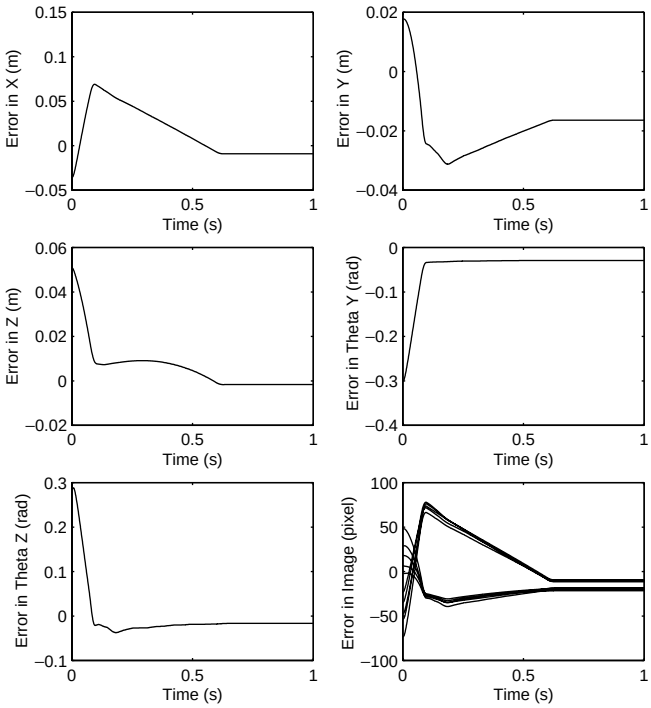

FIGURE 15.11 Simulation results for HVS: desired relative pose of the object with respect to the endpoint frame: (3, 3, 25 cm; 0, 0.3, 0.3 rad). The relative pose errors are specified with respect to the endpoint frame {E}.

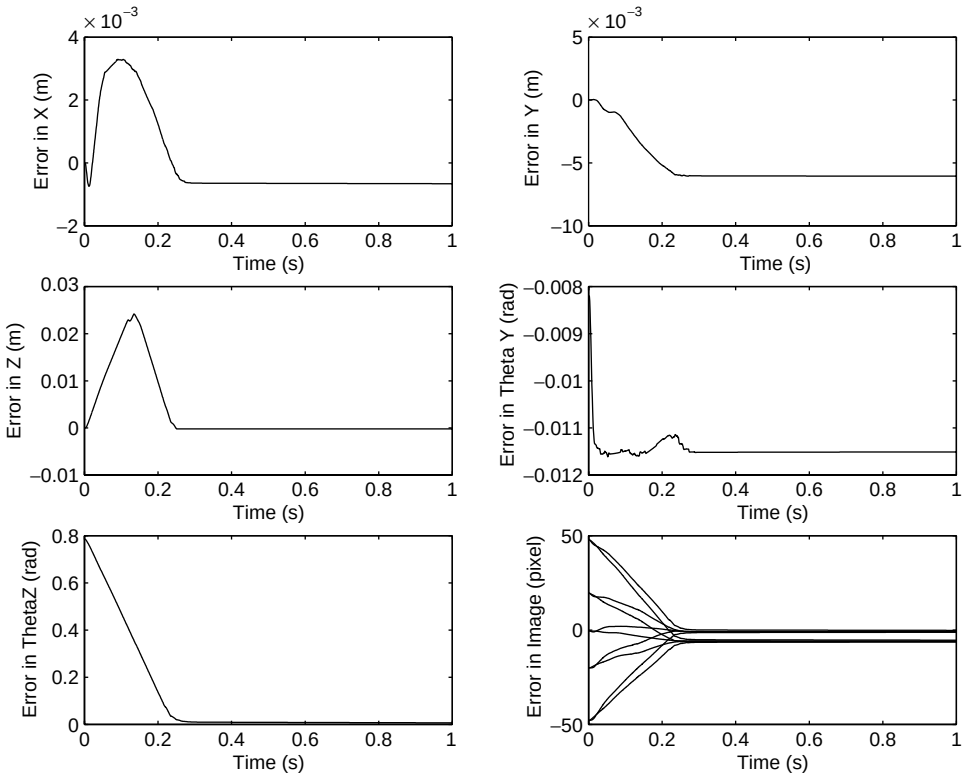

FIGURE 15.12 Simulation results for IBVS: desired relative pose of the object with respect to the endpoint frame: (0, 0, 0 cm; 0, 0, $\frac{}{4}$ rad). The relative pose errors are specified with respect to the endpoint frame {E}.

15.5 Applications

Many applications of visual servoing have been limited to the laboratory and structured environments. For instance, many visual servoing systems use markers, structured light, and artificial objects with high-contrast features. Recent advances in opto-mechatronics technology have led to significant improvements of visual servoing science and practice. A list of visual servoing achievements and applications can be found in Corke [1996]. With the recent progress, it has been possible to design visual servoing systems operating above 60 Hz, tracking and picking objects moving at 30 cm/s, applying sealants at 40 cm/s, and guiding vehicles moving about 96 km/h. In summary, the current state of visual servoing technology has the potential to support robot operation in more realistic environments than today's structured environments. This is particularly required in many emerging technologies for autonomous systems.

Table 15.2 summarizes some of the demonstrated applications of visual servoing. However, many of these applications would not justify visual servoing applications, mainly due to the effectiveness of existing traditional solutions. Commercial applications of visual servoing will occur with those applications that there are not substitute technologies. A good example is applications that require precise positioning, such as fixtureless assembly, within dynamic and uncertain environments.

Table 15.3 summarizes potential new applications of visual servoing. The main obstacle for the development of new applications is related to the robustness of vision in adapting to different and noisy environments. Research and development is underway to address vision and image-processing robustness. This topic is discussed further in Chapters 10 and 14.

TABLE 15.2 Demonstrated Applications of Visual Guidance and Servoing (Speeds Represent Those of the Objects; Numbers with Hz Denote Bandwidth of Visual Servoing System)

Application	Investigators or Organizations, Date
Bolt insertion, picking moving parts from conveyor	Rosen et al., 1976–1978 (SRI Int.)
Picking parts from fast moving conveyor (30 cm/s)	Zhang et al., 1990
Tracking and grasping a toy train (25 cm/s, 60 Hz)	Allen et al., 1991
Visual-guided motion for following and grasping	Hill and Park, 1979
Three-dimensional vision-based grasping of moving objects (61 Hz)	Janabi-Sharifi and Wilson, 1998
Fruit picking	Harrell et al., 1989
Connector acquisition	Mochizuki et al., 1987
Weld seam tracking	Clocksin et al., 1985
Sealant application (40 cm/s, 4.5 Hz)	Sawano et al., 1983
Rocket-tracking camera with pan/tilt (60 Hz)	Gilbert et al., 1980
Planar micro-positioning (300 Hz)	Webber and Hollis, 1988 (IBM Watson Research Center)
Road vehicle guidance (96 Km/h)	Dickmanns and Graefe, 1988
Aircraft landing guidance	Dickmanns and Schell, 1992
Underwater robot control	Negahdaripour and Fox, 1991
Ping-pong bouncing	Anderson, 1987
Juggling	Rizzi and Koditscek, 1991
Inverted pendulum balancing	Dickmanns and Graefe, 1988
Labyrinth game	Anderson et al., 1991
Catching ball	Bukowski et al., 1991
Catching free-flying polyhedron	Skofteland and Hirzinger, 1991
Part mating (10 Hz)	Geschke, 1981
Aircraft refuelling	Leahy et al., 1990
Mating U.S. space shuttle connector	Cyros, 1988
Telerobotics	Papanikolopoulos and Khosla, 1992
Robot hand-eye coordination	Hashimoto et al., 1989

TABLE 15.3 Potential Applications of Visual Guidance and Servoing

Potential Applications
Fixtureless assembly
Automated machining
PC board inspection and soldering
IC insertion
Remote hazardous material handling (e.g., in nuclear power plants)
Weapons disassembly
Remote mining
Textile manufacturing
Automated television and surveillance camera guidance
Remote surgery
Satellite tracking and grasping
Planetary robotic missions

15.6 Summary

Visual servoing, when compared with conventional techniques, offers many advantages for the control of motion. In particular, visual servoing supports autonomous motion control systems without requiring exact specifications of poses for the object and tracker (e.g., robot endpoint). Also, visual servoing could relax the requirement for an exact object model. Visual servoing integration with robotic environments has significant implications such as fixtureless positioning, reduced robot training, and lower robot

manufacturing costs and cycle time. In this chapter, the fundamentals of visual servoing were discussed and emphasis was placed on robotic visual servoing with eye-in-hand configurations. In particular, the emphasis was on the introduction of background theory and well-established methods of visual servoing.

An overview of the background related to visual servoing notations, coordinate transformations, image projection, and object kinematic modeling was given. Also, relevant issues of camera calibration were discussed briefly.

The main structures of visual servoing, namely, position-based, image-based, and hybrid visual servoing structures, were presented. The basic and well-established control method for each structure was given. The advantages and disadvantages of these control structures were compared.

Separation of the pose-estimation problem from the control-design problem, in position-based techniques, allows the control designer to take advantage of well-established robot Cartesian control algorithms. Also, position-based methods permit specification of the relative trajectories in a Cartesian space that provides natural expression of motion for many industrial environments, e.g., tracking and grasping a moving object on a conveyor. The interaction of image-based systems with moving objects, for example, has not been fully established. Position-based visual servoing provides no mechanism for regulating features in image space and, in order to keep the features in the field of view, must rely heavily on feature selection and switching mechanisms. Although both image-based and position-based methods demonstrate difficulties in executing large three-dimensional relative motions, image-based techniques show a response that is inferior to that of position-based methods, mainly due to the highly coupled translational and rotational motions of the camera. Hybrid systems, such as the 2-1/2D method, show superior response in comparison with their counterparts for long-range three-dimensional motions. This is mainly due to the provision of decoupling between translation and rotation of the camera in hybrid systems.

Also, two major issues with image-based methods are the presence of image singularities and camera-retreat problems. These problems do not exist with position-based and hybrid methods. One disadvantage of position-based methods over image-based and hybrid techniques is their sensitivity to camera calibration and object model errors. Furthermore, the required computation time of the position-based method is greater than that in image-based and hybrid methods. Hybrid methods usually rely on the estimation of a homography matrix. This estimation might be computationally expensive and sensitive to camera calibrations. For instance, with noncoplanar objects and conventional SER estimation of a homography matrix, more feature points might be required than those with other visual servoing methods. However, with the recent advances in microprocessor technology, the computation-time should not pose any serious problems. In all of the techniques, the visual control loop must be designed to provide higher bandwidth than that of robot position loops. Otherwise, the system control, like any other discrete feedback system with a delay, might become unstable by increasing the loop gain.

Simulations were done to demonstrate the performance of each servoing structure with the subscribed control strategy. Moreover, the effects of different design parameters were studied and some conclusions were drawn.

Finally, the demonstrated and potential applications of visual servoing techniques were summarized. Future research and development activities related to visual servoing were also highlighted.

Defining Terms

extrinsic camera parameters: Characteristics of the position and orientation of a camera, e.g., the homogeneous transform between the camera and the base frame.

feature: Any scene property that can be mapped onto and measured in the image plane. Any structural feature that can be extracted from an image is called image feature and usually corresponds to the projection of a physical feature of objects onto the image plane. Image features can be divided into region-based features, such as planes, areas, holes, and edge segment-based features, such as corners and edges.

image feature parameter: Any quantity with real value that can be obtained from image features. Examples include coordinates of image points; the length and orientation of lines connecting points in an image, region area, centroid, and moments of projected areas; parameters of lines, curves, or regular regions such as circles and ellipses.

image Jacobian matrix: Relates the velocity (or differential changes) of the camera r (or r) or its relative pose W (or W) to the image feature velocities $\dot{s}$ (or s). This matrix is also referred to as feature Jacobian matrix, feature sensitivity matrix, interaction matrix, or $\boldsymbol{B}$ matrix.

intrinsic parameters of the camera: Inner characteristics of the camera and sensor, such as focal length and radial and tangential distortion parameters, and the coordinates of the principal point, where the optical axis intersects the image plane.

visual servoing (vision-guided servoing): The use of vision in the feedback loop of the lowest level of a (usually robotic) system control with fast image processing to provide reactive behavior. The task of visual servoing for robotic manipulators (or robotic visual servoing, RVS) is to control the pose of the robot's end-effector relative to either a world coordinate frame or an object being manipulated, using real-time visual features extracted from the image. The camera can be fixed or mounted at the endpoint (eye-in-hand configuration).

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through Research Grant #203060-98. I would also like to thank my Ph.D. student Lingfeng Deng for his assistance in the preparation of the simulation results.

References

Chaumette, F., Potential problems of stability and convergence in image-based and position-based visual servoing, *The Confluence of Vision and Control*, Vol. 237 of *Lecture Notes in Control and Information Sciences*, Springer-Verlag, New York, 1998, pp. 66–78.

Corke, P. I., *Visual Control of Robots: High Performance Visual Servoing*, Research Studies, Ltd., Somerset, England, 1996.

Corke, P. I. and Hutchnison, S. A., A new hybrid image-based visual servo-control scheme, *Proc. IEEE Int. Conf. Decision and Control*, 2000, pp. 2521–2526.

Feddema, J. T., Lee, C. S. G., and Mitchell, O. R., Weighted selection of image features for resolved rate visual feedback control, *IEEE Trans. Robot. Automat.*, 7(1), 31–47, 1991.

Ficocelli, M., Camera Calibration: Intrinsic Parameters, Technical Report TR-1999-12-17-01, Robotics and Manufacturing Automation Laboratory, Ryerson University, Toronto, 1999.

Ficocelli, M. and Janabi-Sharifi, F., Adaptive Filtering for Pose Estimation in Visual Servoing, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2001*, Maui, Hawaii, 2001, pp. 19–24.

Hutchinson, S., Hager, G., and Corke, P. I., A tutorial on visual servoing, *IEEE Trans. Robot. Automat.*, 12(5), 651–670, 1996.

Janabi-Sharifi, F. and Wilson, W. J., Automatic selection of image features for visual servoing, *IEEE Trans. Robot. Automat.*, 13(6), 890–903, 1997.

Malis, E., Chaumette, F., and Boudet, S., 2-1/2D visual servoing, *IEEE Trans. Robot. Automat.*, 15(2), 238–250, 1999.

Sanderson, A. C. and Weiss, L. E., Image-based visual servo control using relational graph error signals, *Proc. IEEE*, 1980, pp. 1074–1077.

Shirai, Y. and Inoue, H., Guiding a robot by visual feedback in assembling tasks, *Pattern Recognition*, 5, 99–108, 1973.

Tsai, R. and Lenz, R., A new technique for fully autonomous and efficient three-dimensional robotic hand/eye calibration, *IEEE Trans. Robot. Automat.*, 5(3), 345–358, 1989.

Wang, J., Optimal Estimation of Three-Dimensional Relative Position and Orientation for Robot Control, M.A.Sc. dissertation, Dept. of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada, 1992.

Wilson, W. J., Williams Hulls, C. C., and Bell, G. S., Relative end-effector control using cartesian position-based visual servoing, *IEEE Trans. Robot. Automat.*, 12(5), 684–696, 1996.

Wilson, W. J., Williams-Hulls, C. C., and Janabi-Sharifi, F., Robust image processing and position-based visual servoing, in *Robust Vision for Vision-Based Control of Motion*, Vincze, M. and Hager, G. D., Eds., IEEE Press, New York, 2000, pp. 163–201.

Yuan, J. S. C., A general photogrammetric method for determining object position and orientation, *IEEE Trans. Robot. Automat.*, 5(2), 129–142, 1989.

For Further Information

A good collection of articles on visual servoing can be found in *IEEE Trans. Robot. Automat.*, 12(5), 1996. This issue includes an excellent tutorial on visual servoing. A good reference book is *Visual Control of Robots: High Performance Visual Servoing* by P. I. Corke (Research Studies, Ltd., Somerset, England, 1996), encompassing both theoretical and practical aspects related to visual servoing of robotic manipulators. *Robust Vision for Vision-Based Control of Motion*, edited by M. Vincze and G. D. Hager (IEEE Press, New York, 2000) provides recent advances in the development of robust vision for visual servo-controlled systems. The articles span issues including object modeling, feature extraction, feature selection, sensor data fusion, and visual tracking. *Robot Vision*, by B. K. Horn (McGraw-Hill, New York, 1986) is a comprehensive introductory book for the application of machine vision in robotics.

Proceedings of IEEE International Conference on Robotics and Automation, *IEEE Robotics and Automation Magazine*, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, *IEEE Transactions on Robotics and Automation*, and *IEEE Transactions on Systems, Man*, and *Cybernetics* document the latest developments in visual servoing.

Several articles have compared the performances of basic visual servoing methods. Among them are "Potential Problems of Stability and Convergence in Image-Based and Position-Based Visual Servoing," by F. Chaumette, in *The Confluence of Vision and Control*, Vol. 237 of *Lecture Notes in Control and Information Sciences* (Springer-Verlag, New York, 1998, pp. 66–78), and "Stability and Robustness of Visual Servoing Methods," by L. Deng, F. Janabi-Sharifi, and W. J. Wilson, in *Proc. IEEE Int. Conf. Robot. Automat.* (Washington, D.C., May 2002).

Good articles for camera calibration include the one by Tsai and Lenz [1989] and also "Hand-Eye Calibration," by R. Horaud and F. Dornaike, in *International Journal of Robotics Research*, 14(3), 195–210, 1995. Implementation details can also be found in Corke [1996].

Articles for analytical pose estimation include the following. For pose estimation using four coplanar but not collinear feature points, see "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," by M. A. Fischler and R. C. Bolles, in *Comm. ACM*, 24, 381–395, 1981. If camera intrinsic parameters need to be estimated, six or more feature points will be required for a unique solution. This is shown in "Decomposition of Transformation Matrices for Robot Vision," by S. Ganapathy, in *Pattern Recog. Lett.* 401–412, 1989. See also "Analysis and Solutions of the Three Point Perspective Pose Estimation Problem," by R. M. Haralick, C. Lee, K. Ottenberg, and M. Nolle, in *Proc. IEEE Conf. Comp. Vision, Pattern. Recog.*, pp. 592–598, 1991, and "An Analytic Solution for the Perspective 4-Point Problem," by R. Horaud, B. Canio, and O. Leboullenx, *Computer Vision Graphics, Image Process.*, no. 1, 33–44, 1989.

The articles for the least-squares solutions in pose estimation include an article by S. Ganapathy [1989], mentioned above, and "Determination of Camera Location from 2-D to 3-D Line and Point Correspondences," by Y. Liu, T. S. Huang, and O. D. Faugeras, in *IEEE Trans. Pat. Anal. Machine Intell.*, no. 1, 28–37, 1990. Also, "Constrained Pose Refinement of Parametric Objects," by R. Goldberg, in *Int. J. Comput. Vision*, no. 2, 181–211, 1994. A fast and globally convergent orthogonal iteration (OI) algorithm is

introduced in "Fast and Globally Convergent Pose Estimation from Video Images," by C. P. Lu and G. D. Hager, in *IEEE Trans. Patt. Analysis and Machine Intell.*, 22(6), 610–622, June 2000.

The following references provide solutions to online estimation of depth information and image Jacobian matrix for IBVS. "Manipulator Control with Image-Based Visual Servo," by K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, in *Proc. IEEE Int.Conf. Robot. Automat.*, Piscataway, NJ, 1991, pp. 2267–2272 provides an explicit depth estimation based on the feature analysis. An adaptive control-based method for depth estimation is provided in "Controlled Active Vision," by N. P. Papanikolopoulos, Ph.D. dissertation, Dept. of Electrical and Computer Engineering, Carnegie Mellon University, 1992. It is also proposed to use an approximation to the value of the image Jacobian matrix computed at the desired camera position, in "A New Approach to Visual Servoing in Robotics," by B. Espiau, F. Chaumette, and P. Rives, in *IEEE Trans. Robot. Automat.*, 8(3), 313–326, 1992.

There are different image Jacobian derived in the literature. The following sources could be studied for further details: [Feddema et al., 1991], and "Vision Resolvability for Visually Servoed Manipulation," by B. Nelson, and P. K. Khosla, in *Journal of Robotic Systems*, 13(2), 75–93, 1996. Also see "Controlled Active Vision," by N. P. Papanikolopoulos, Ph.D. dissertation, Dept. of Electrical and Computer Engineering, Carnegie Mellon University, 1992.

Other forms of image Jacobian matrices have been derived using geometrical entities, such as spheres and lines, are also available in: "A New Approach to Visual Servoing in Robotics," by B. Espiau et al., mentioned above. Also see Malis et al. [1999] for Jacobian matrix for 2-1/2D servoing.

In addition to Malis et al. [1999], the following papers are good resources for the study of HVS. For instance, the following paper proposes a solution for features leaving field-of-view in 2-1/2D HVS: "Explicit Incorporation of 2-D Constraints in Vision-Based Control of Robot Manipulators," by G. Morel, T. Liebezeit, J. Szewczyk, S. Boudet, and J. Pot, in *Experimental Robotics VI*, P. I. Corke and J. Trevelyan, Eds., Vol. 250, Springer-Verlag, 2000, pp. 99–108. The following article proposes to compute translational velocity instead of rotational one for HVS: "Optimal Motion Control for Image-Based Visual Servoing by Decoupling Translation and Rotation," by K. Deguchi, in *Proc. IEEE Int. Conf. Intel. Robotics and Systems*, 1998, pp. 705–711. Also see Corke and Hutchinson [2000] for a new partitioning of camera degrees of freedom. That includes separating optical axis z and computing z-axis velocity using two new image features.

The following sources are useful for the study of homography matrix and its decomposition used in HVS: *Three-Dimensional Computer Vision*, by O. Faugeras, MIT Press, Cambridge, MA, 1993, and "Motion and Structure from Motion in a Piecewise Planar Environment," in *Int. J. Pattern Recogn. Artificial Intelligence*, 2, 485–508, 1988.

The following sources cover the applications developed using VS techniques: Corke [1996] and "Visual Servoing: A Technology in Search of an Application," by J. T. Feddema, in *Proc. IEEE Int. Conf. Robot. Automat: Workshop on Visual Servoing: Achievements, Applications, and Open Problems*, San Diego, CA, 1994.