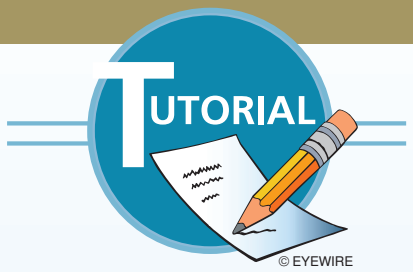
Visual servo control, Part II: Advanced approaches

François Chaumette, S. Hutchinson



Visual Servo Control

Part II: Advanced Approaches

This article is the second of a two-part tutorial on visual servo control. In Part I (*IEEE Robotics and Automation Magazine, vol. 13, no. 4*), we introduced fundamental concepts and described basic approaches. Here we discuss more advanced concepts, and present a number of recent approaches.

As described in Part I of the tutorial, visual servo schemes rely on the relationship

$$\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}\mathbf{v}_c \tag{1}$$

in which $\mathbf{s}$ is a set of geometrical features whose time derivative $\dot{\mathbf{s}}$ is linearly related to the spatial velocity $\mathbf{v}_c$ of the camera through the interaction matrix $\mathbf{L}_{\mathbf{s}}$. Using this relationship, control schemes are designed to minimize the error $\mathbf{e}$ between the current value of the visual feature $\mathbf{s}$ and its desired value $\mathbf{s}^*$: $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$.

A classical proportional control scheme is given by:

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}_{\mathbf{e}}^{+}}\mathbf{e}, \tag{2}$$

where $\mathbf{L}_{\mathbf{e}}$ is defined by

$$\dot{\mathbf{e}} = \mathbf{L}_{\mathbf{e}}\mathbf{v}_c, \tag{3}$$

and $\widehat{\mathbf{L}_{\mathbf{e}}^{+}}$ is an approximation of the pseudo-inverse of $\mathbf{L}_{\mathbf{e}}$. An approximation must be used because visual servo schemes require the values of three-dimensional (3-D) parameters that are not available directly from the image measurements. Recall that for position-based visual servo (PBVS) schemes 3-D parameters appear both in the error $\mathbf{e}$ and in the interaction matrix, while for basic image-based visual servo (IBVS), the depth of each point considered appears in the coefficients of the interaction matrix related to the translational motions. This is the case even when $\widehat{\mathbf{L}_{\mathbf{e}}^{+}} = \widehat{\mathbf{L}_{\mathbf{e}^*}^{+}}$ is used in the control scheme, although in this case only the depth Z^* of each point for the desired pose is needed, which is generally not difficult to obtain in practice. In all other cases, an estimation of the current depth must be made at each iteration of the control scheme.

We begin Part II of the tutorial by describing two approaches to estimating the interaction matrix. First we describe how epipolar geometry can be used to estimate the 3-D parameters, which can then be used to construct the interaction matrix. We then describe how it is possible to estimate directly the numerical value of $\widehat{\mathbf{L}_{\mathbf{e}}^{+}}$. With these methods in hand, we then present more advanced techniques in visual servo control. These techniques aim to compensate for the relative shortcomings of the PBVS and IBVS methods. We then consider target tracking tasks, that is, tasks for which the target object is not stationary. Finally, we present a generalization of the modeling issues that allows one to consider both eye-in-hand systems and eye-to-hand systems.

Estimation of 3-D Parameters

If a calibrated stereo vision system is used, all 3-D parameters can be easily determined by triangulation, as evoked in Part I of the tutorial. Similarly, if a 3-D model of the object is known, all 3-D parameters can be computed from a pose estimation algorithm. However, such an estimation can be quite unstable due to image noise. It is also possible to estimate 3-D parameters by using the epipolar geometry that relates the images of the same scene observed from different viewpoints. Indeed, in visual servoing, two images are generally available:

BY FRANÇOIS CHAUMETTE AND SETH HUTCHINSON

1070-9932/07/$25.00©2007 IEEE

Epipolar geometry can be used to estimate the 3-D parameters, which can then be used to construct the interaction matrix.

the current one and the desired one. In contrast, we can estimate the interaction matrix directly as image measurements are made during the visual servoing task. Here, we discuss both approaches.

Epipolar Geometry

Given a set of matches between the image measurements in the current image and in the desired one, the fundamental matrix, or the esential matrix if the camera is calibrated, can be recovered [1], and then used in visual servoing [2]. Indeed, from the essential matrix, the rotation and the translation up to a scalar factor between the two views can be estimated. However, near the convergence of the visual servo, that is when the current and desired images are similar, the epipolar geometry becomes degenerate and it is not possible to estimate accurately the partial pose between the two views. For this reason, using homography is generally prefered.

Let $\mathbf{x}_i$ and $\mathbf{x}_i^*$ denote the homogeneous image coordinates for a point in the current and desired images. Then $\mathbf{x}_i$ is related to $\mathbf{x}_i^*$ by

$$\mathbf{x}_i = \mathbf{H}_i \mathbf{x}_i^*,$$

in which $\mathbf{H}_i$ is a homography matrix.

If all feature points lie on a 3-D plane, then there is a single homography matrix $\mathbf{H}$ such that $\mathbf{x}_i = \mathbf{H}\mathbf{x}_i^*$ for all i. This homography can be estimated using the position of four matched points in the desired and the current images. If all the features points do not belong to the same 3-D plane, then three points can be used to define such a plane and five supplementary points are needed to estimate $\mathbf{H}$ [3].

Once $\mathbf{H}$ is available, it can be decomposed as

$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{t}}{d^*}\mathbf{n}^{*\top}, \tag{4}$$

in which $\mathbf{R}$ is the rotation matrix relating the orientation of the current and desired camera frames, $\mathbf{n}^*$ is the normal to the chosen 3-D plane expressed in the desired frame, d^* is the distance to the 3-D plane from the desired frame, and $\mathbf{t}$ is the translation between current and desired frames. From $\mathbf{H}$, it is thus posible to recover $\mathbf{R}$, $\mathbf{t}/d^*$, and $\mathbf{n}$. In fact, two solutions for these quantities exist [4], but it is quite easy to select the correct one using some knowledge about the desired pose. It is also possible to estimate the depth of any target point up to a common scale factor [5]. The unknown depth of each point that appears in classical IBVS can thus be expressed as a function of a single constant parameter. Similarly, the pose parameters required by PBVS can be recovered up to a scalar factor as for the translation term. The PBVS schemes described in Part I of the tutorial can thus be revisited using this approach, with the new error defined as the translation up to a scalar factor and the angle/axis parameterization of the rotation. This approach has also been used for the hybrid visual servoing schemes described in the next section.

Direct Estimation

The approach described previously can be used to estimate the unknown 3-D parameters that appear in the analytical form of the interaction matrix. It is also possible to estimate directly its numerical value using either an off-line learning step, or an on-line estimation scheme. This idea is only useful for IBVS since, for PBVS, all the coefficients of the interaction matrix are directly obtained from the features $\mathbf{s}$ used in the control scheme.

All the methods proposed to estimate numerically the interaction matrix rely on observations of the variation of features due to a known or measured camera motion. More precisely, if we measure a feature's variation $\Delta\mathbf{s}$ due to a camera motion $\Delta\mathbf{v}_c$, we have from (1)

$$\mathbf{L}_\mathbf{s}\Delta\mathbf{v}_c = \Delta\mathbf{s},$$

which provides k equations, while we have $k \times 6$ unknown values in $\mathbf{L}_\mathbf{s}$. Using a set of N independent camera motions with $N > 6$, it is thus possible to estimate $\mathbf{L}_\mathbf{s}$ by solving

$$\mathbf{L}_\mathbf{s}\mathbf{A} = \mathbf{B},$$

where the columns of $\mathbf{A} \in \mathrm{R}^{6\times N}$ and $\mathbf{B} \in \mathrm{R}^{k\times N}$ are respectively formed with the set of camera motions and the set of corresponding features variations. The least squares solution is of course given by

$$\widehat{\mathbf{L}_\mathbf{s}} = \mathbf{B}\mathbf{A}^+. \tag{5}$$

Methods based on neural networks have also been developed to estimate $\mathbf{L}_\mathbf{s}$ [6], [7]. It is also possible to estimate directly the numerical value of $\mathbf{L}_\mathbf{s}^+$, which provides in practice a better behavior [8]. In that case, the basic relation is

$$\mathbf{L}_\mathbf{s}^+\Delta\mathbf{s} = \Delta\mathbf{v}_c,$$

which provides six equations. Using a set of N measurements, with $N > k$, we now obtain

$$\widehat{\mathbf{L}_\mathbf{s}^+} = \mathbf{A}\mathbf{B}^+. \tag{6}$$

In the first case (5), the six columns of $\mathbf{L}_\mathbf{s}$ are estimated by solving six linear systems, while in the second case (6), the k columns of $\mathbf{L}_\mathbf{s}^+$ are estimated by solving k linear systems, which explains the difference in the results.

Estimating the interaction matrix online can be viewed as an optimization problem, and consequently a number of

researchers have investigated approaches that derive from optimization methods. These methods typically discretize the system equation (1), and use an iterative updating scheme to refine the estimate of $\widehat{\mathbf{L}}_{\mathbf{s}}$ at each stage. One such online and iterative formulation uses the Broyden update rule given by [9], [10]

$$\widehat{\mathbf{L}}_{\mathbf{s}}(t+1) = \widehat{\mathbf{L}}_{\mathbf{s}}(t) + \frac{\alpha}{\Delta\mathbf{v}_c^\top \Delta\mathbf{v}_c}(\Delta\mathbf{x} - \widehat{\mathbf{L}}_{\mathbf{s}}(t)\Delta\mathbf{v}_c)\Delta\mathbf{v}_c^\top,$$

where α defines the update speed. This method has been generalized to the case of moving objects in [11].

The main interest of using such numerical estimations in the control scheme is that it avoids all the modeling and calibration steps. It is particularly useful when using features whose interaction matrix is not available in analytical form. For instance, in [12], the main eigenvalues of the Principal Component Analysis of an image have been considered in a visual servoing scheme. The drawbacks of these methods is that no theoretical stability and robustness analysis can be made.

Advanced Visual Servo Control Schemes

We now describe visual servo control schemes which have been proposed to improve the behavior of basic IBVS and PBVS. The first ones combine the respective advantages of these schemes while trying to avoid their shortcomings.

Hybrid Visual Servo

Suppose we have access to a clever control law for $\boldsymbol{\omega}_c$, such as the one used in PBVS

$$\boldsymbol{\omega}_c = -\lambda\,\theta\mathbf{u}, \tag{7}$$

where $\theta\mathbf{u}$ is obtained either from a pose estimation algorithm if the 3-D model of the object is available, or from the partial pose estimated using the epipolar geometry or the homography. How could we use this in conjunction with traditional IBVS?

Considering a feature vector $\mathbf{s}_t$ and an error $\mathbf{e}_t$ devoted to control the translational degrees of freedom, we can partition the interaction matrix as follows:

$$\begin{aligned}\dot{\mathbf{s}}_t &= \mathbf{L}_{\mathbf{s}_t}\mathbf{v}_c \\ &= [\mathbf{L}_{\boldsymbol{v}} \quad \mathbf{L}_{\boldsymbol{\omega}}]\begin{bmatrix}\boldsymbol{v}_c \\ \boldsymbol{\omega}_c\end{bmatrix} \\ &= \mathbf{L}_{\boldsymbol{v}}\boldsymbol{v}_c + \mathbf{L}_{\boldsymbol{\omega}}\boldsymbol{\omega}_c.\end{aligned}$$

Now, setting $\dot{\mathbf{e}}_t = -\lambda\mathbf{e}_t$, we can solve for the desired translational control input as

$$\begin{aligned}-\lambda\mathbf{e}_t &= \dot{\mathbf{e}}_t = \dot{\mathbf{s}}_t = \mathbf{L}_{\boldsymbol{v}}\boldsymbol{v}_c + \mathbf{L}_{\boldsymbol{\omega}}\boldsymbol{\omega}_c \\ &\Rightarrow \boldsymbol{v}_c = -\mathbf{L}_{\boldsymbol{v}}^{+}(\lambda\mathbf{e}_t + \mathbf{L}_{\boldsymbol{\omega}}\boldsymbol{\omega}_c).\end{aligned} \tag{8}$$

We can think of the quantity $(\lambda\mathbf{e}_t + \mathbf{L}_{\boldsymbol{\omega}}\boldsymbol{\omega}_c)$ as a modified error term, one that combines the original error with the error that would be induced by the rotational motion due to ω_c. The translational control input $\boldsymbol{v}_c = -\mathbf{L}_{\boldsymbol{v}}^{+}(\lambda\mathbf{e}_t + \mathbf{L}_{\boldsymbol{\omega}}\boldsymbol{\omega}_c)$ will drive this error to zero.

The method known as two and one-half-dimensional (2 1/2-D) Visual Servo [13] was the first to exploit such a partitioning in combining IBVS and PBVS. More precisely, in [13], $\mathbf{s}_t$ has been selected as the coordinates of an image point, and the logarithm of its depth, so that $\mathbf{L}_{\boldsymbol{v}}$ is a triangular always invertible matrix. More precisely, we have $\mathbf{s}_t = (\mathbf{x}, \log Z)$, $\mathbf{s}_t^* = (\mathbf{x}^*, \log Z^*)$, $\mathbf{e}_t = (\mathbf{x} - \mathbf{x}^*, \log\rho_Z)$ where $\rho_Z = Z/Z^*$, and

$$\mathbf{L}_{\boldsymbol{v}} = \frac{1}{Z^*\rho_Z}\begin{bmatrix}-1 & 0 & x \\ 0 & -1 & y \\ 0 & 0 & -1\end{bmatrix}$$

$$\mathbf{L}_{\omega} = \begin{bmatrix}xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \\ -y & x & 0\end{bmatrix}.$$

Note that the ratio ρ_Z can be obtained directly from the partial pose estimation algorithm described previously.

If we come back to the usual global representation of visual servo control schemes, we have $\mathbf{e} = (\mathbf{e}_t, \theta\mathbf{u})$ and $\mathbf{L}_{\mathbf{e}}$ given by

$$\mathbf{L}_{\mathbf{e}} = \begin{bmatrix}\mathbf{L}_{\boldsymbol{v}} & \mathbf{L}_{\omega} \\ \mathbf{0} & \mathbf{L}_{\theta\mathbf{u}}\end{bmatrix},$$

from which we obtain immediately the control law (7) and (8) by applying (2).

If we consider the example chosen in Part I of the tutorial to compare the behavior of the different control schemes, the results obtained using (8) are given in Figure 1. Here, the point that has been considered in $\mathbf{s}_t$ is the center of gravity $\mathbf{x}_g$ of the target. Note that the image trajectory of this point is a straight line as expected, and that there is a nice decrease of the camera velocity components, which makes this scheme very near the first PBVS approach.

As for stability, it is clear that this scheme is globally asymptotically stable in perfect conditions (refer to the stability

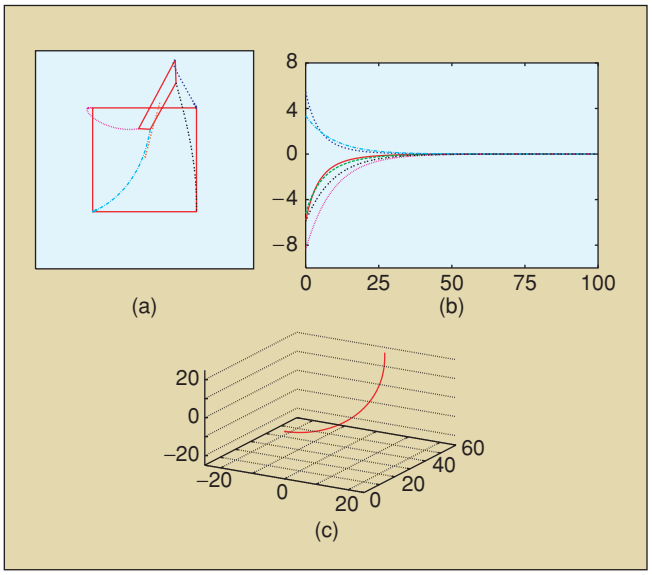


Figure 1. *System behavior using* $\mathbf{s} = (\mathbf{x}_g, \log(Z_g), \theta\mathbf{u})$.

analysis provided in the Part I of the tutorial: we are here in the simple case $k = 6$). Furthermore, thanks to the triangular form of the interaction matrix $\mathbf{L_e}$, it has been possible to analyze the stability of this scheme in the presence of calibration errors using the homography estimation approach [14]. Finally, the only unknown constant parameter involved in this scheme, that is Z^*, can be estimated on line using adaptive techniques [15].

Other hybrid schemes can be designed. For instance, in [16], the third component of $\mathbf{s}_t$ is different and has been selected so that all the target points remain in the camera field of view as far as possible. Another example has been proposed in [17]. In that case, $\mathbf{s}$ is selected as $\mathbf{s} = ({}^{c^*}\mathbf{t}_c, \mathbf{x}_g, \theta u_z)$ which provides with a block-triangular interaction matrix of the form:

$$\mathbf{L_e} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{L}'_v & \mathbf{L}'_\omega \end{bmatrix},$$

where $\mathbf{L}'_v$ and $\mathbf{L}'_\omega$ can easily be computed. This scheme is so that, in perfect conditions, the camera trajectory is a straight line (since ${}^{c^*}\mathbf{t}_c$ is a part of $\mathbf{s}$), and the image trajectory of the center of gravity of the object is also a straight line (since $\mathbf{x}_g$ is also a part of $\mathbf{s}$). The translational camera degrees of freedom are devoted to realize the 3-D straigth line, while the rotational camera degrees of freedom are devoted to realize the two-dimensional (2-D) straight line and compensate also the 2-D motion of $\mathbf{x}_g$ due to the transational motion. As can be seen in Figure 2, this scheme is particularly satisfactory in practice.

Finally, it is possible to combine differently 2-D and 3-D features. For instance, in [18], it has been proposed to use in $\mathbf{s}$ the 2-D homogeneous coordinates of a set of image points expressed in pixels multiplied by their corresponding depth: $\mathbf{s} = (u_1 Z_1, v_1 Z_1, Z_1, \ldots, u_n Z_n, v_n Z_n, Z_n)$. As for classical IBVS, we obtain in that case a set of redundant features, since at least three points have to be used to control the six camera degrees of freedom (we here have $k \geq 9$). However, it has been demonstrated in [19] that this selection of redundant features is free of attractive local minima.

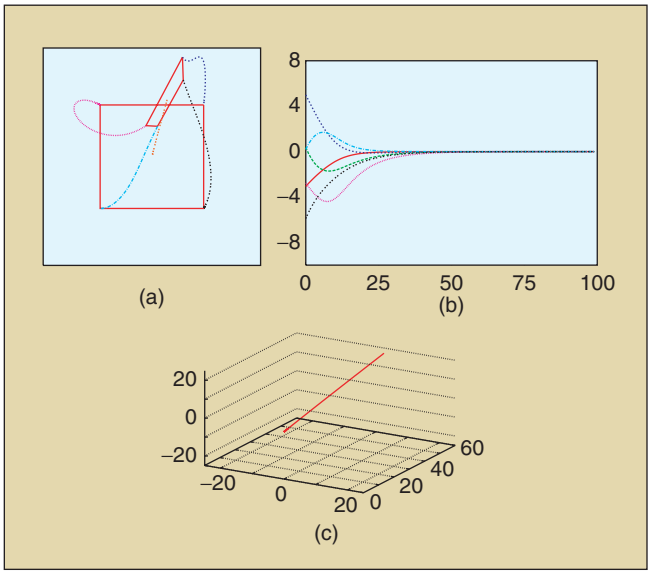


Figure 2. *System behavior using* $\mathbf{s} = ({}^{c^*}\mathbf{t}_c, \mathbf{x}_g, \theta u_z)$.

Partitioned Visual Servo

The hybrid visual servo schemes described previously have been designed to decouple the rotational motions from the translational ones by selecting adequate visual features defined in part in 2-D, and in part in 3-D (that is why they have been called 2 1/2-D visual servoing). This work has inspired some researchers to find features that exhibit similar decoupling properties but using only features expressed directly in the image. More precisely, the goal is to find six features so that each of them is related to only one degree of freedom (in which case the interaction matrix is a diagonal matrix), the Grail being to find a diagonal interaction matrix whose elements are constant, as near as possible of the identity matrix, leading to a pure, direct, and simple linear control problem.

The first work in this area partitioned the interaction matrix so as to isolate motion related to the optic axis [20]. Indeed, whatever the choice of $\mathbf{s}$, we have

$$\begin{aligned} \dot{\mathbf{s}} &= \mathbf{L_s}\mathbf{v}_c \\ &= \mathbf{L}_{xy}\mathbf{v}_{xy} + \mathbf{L}_z\mathbf{v}_z \\ &= \dot{\mathbf{s}}_{xy} + \dot{\mathbf{s}}_z, \end{aligned}$$

in which $\mathbf{L}_{xy}$ includes the first, second, fourth and fifth columns of $\mathbf{L_s}$, and $\mathbf{L}_z$ includes the third and sixth columns of $\mathbf{L_s}$. Similarly, $\mathbf{v}_{xy} = (v_x, v_y, \omega_x, \omega_y)$ and $\mathbf{v}_z = (v_z, \omega_z)$. Here, $\dot{\mathbf{s}}_z = \mathbf{L}_z\mathbf{v}_z$ gives the component of $\dot{\mathbf{s}}$ due to the camera motion along and rotation about the optic axis, while $\dot{\mathbf{s}}_{xy} = \mathbf{L}_{xy}\mathbf{v}_{xy}$ gives the component of $\dot{\mathbf{s}}$ due to velocity along and rotation about the camera x and y axes.

Proceeding as before, by setting $\dot{\mathbf{e}} = -\lambda\mathbf{e}$ we obtain

$$-\lambda\mathbf{e} = \dot{\mathbf{e}} = \dot{\mathbf{s}} = \mathbf{L}_{xy}\mathbf{v}_{xy} + \mathbf{L}_z\mathbf{v}_z,$$

which leads to

$$\mathbf{v}_{xy} = -\mathbf{L}^+_{xy}(\lambda\mathbf{e}(t) + \mathbf{L}_z\mathbf{v}_z).$$

As before, we can consider $(\lambda\mathbf{e}(t) + \mathbf{L_z}\mathbf{v}_z)$ as a modified error that incorporates the original error while taking into account the error that will be induced by $\mathbf{v}_z$.

Given this result, all that remains is to choose $\mathbf{s}$ and $\mathbf{v}_z$. As for basic IBVS, the coordinates of a collection of image points can be used in $\mathbf{s}$, while two new image features can be defined to determine $\mathbf{v}_z$.

- Define α, with $0 \leq \alpha < 2\pi$ as the angle between the horizontal axis of the image plane and the directed line segment joining two feature points. It is clear that α is closely related to the rotation around the optic axis.
- Define σ^2 to be the area of the polygon defined by these points. Similarly, σ^2 is closely related to the translation along the optic axis.

Using these features, $\mathbf{v}_z$ has been defined in [20] as

$$\begin{cases} v_z &= \lambda_{v_z} \ln \frac{\sigma^*}{\sigma} \\ \omega_z &= \lambda_{\omega_z}(\alpha^* - \alpha). \end{cases}$$

IBVS with Cylindrical Coordinates of Image Points

As proposed in [21], another option is to use the cylindrical coordinates (ρ, θ) of the image points instead of their Cartesian coordinates (x, y). They are defined by

$$\rho = \sqrt{x^2 + y^2} \quad \theta = \arctan\frac{y}{x} \tag{9}$$

as long as $\rho \neq 0$, in which case θ is not defined. We deduce from (9):

$$\dot{\rho} = (x\dot{x} + y\dot{y})/\rho \quad \dot{\theta} = (x\dot{y} - y\dot{x})/\rho^2.$$

Using the interaction matrix $\mathbf{L_x}$ given in Part I of the tutorial to express $\dot{x}$ and $\dot{y}$, and then substituting x by $\rho\cos\theta$ and y by $\rho\sin\theta$, we obtain immediately

$$\begin{aligned} \mathbf{L}_\rho &= \begin{bmatrix} \frac{-c}{Z} & \frac{-s}{Z} & \frac{\rho}{Z} & (1+\rho^2)s & -(1+\rho^2)c & 0 \end{bmatrix} \\ \mathbf{L}_\theta &= \begin{bmatrix} \frac{s}{\rho Z} & \frac{-c}{\rho Z} & 0 & c/\rho & s/\rho & -1 \end{bmatrix}, \end{aligned}$$

where $c = \cos\theta$ and $s = \sin\theta$. By looking at the third and sixth column of $\mathbf{L}_\rho$ and $\mathbf{L}_\theta$, we can note that ρ is invariant with respect to ω_z while θ is invariant with respect to v_z and linearly related to ω_z, as were the two image features σ and α chosen previously. This explains why the behavior obtained in that case is quite satisfactory. Indeed, by considering the same example as before, and using now $\mathbf{s} = (\rho_1, \theta_1, \ldots, \rho_4, \theta_4)$ and $\widehat{\mathbf{L}_\mathbf{e}^+} = \mathbf{L}_{\mathbf{e}^*}^+$ (that is a constant matrix), the system behavior shown in Figure 3 has the same nice properties as using $\widehat{\mathbf{L}_\mathbf{e}^+} = (\mathbf{L_e}/2 + \mathbf{L}_{\mathbf{e}^*}/2)^+$ with $\mathbf{s} = (x_1, y_1, \ldots, x_4, y_4)$. If we go back to the example used in the geometrical interpretation of IBVS described in Part I of the tutorial, the behavior obtained will also be as expected, thanks to the decoupling between the third and sixth columns of the interaction matrix.

Advanced IBVS Schemes

Up until now, for all IBVS schemes presented, we have primarily considered image point coordinates in $\mathbf{s}$. Other geometrical primitives can of course be used. There are several reasons to do so. First, the scene observed by the camera cannot always be described merely by a collection of points, in which case the image processing provides other types of measurements, such as a set of straight lines or the contours of an object. Second, richer geometric primitives may ameliorate the decoupling and linearizing issues that motivate the design of partitioned systems. Finally, the robotic task to be achieved may be expressed in terms of virtual linkages (or fixtures) between the camera and the observed objects [22], [23], sometimes expressed directly by constraints between primitives, such as for instance point-to-line [24] (which means that an observed point must lie on a specified line).

The main interest of using such numerical estimations in the control scheme is that it avoids all the modeling and calibration steps.

For the first point, it is possible to determine the interaction matrix related to the perspective projection of a large class of geometrical primitives, such as segments, straight lines, spheres, circles, and cylinders. The results are given in [25] and [22]. Recently, the analytical form of the interaction matrix related to any image moments corresponding to planar objects has been computed. This makes possible to consider planar objects of any shape [26]. If a collection of points is measured in the image, moments can also be used [27]. In both cases, moments allow the use of intuitive geometrical features, such as the center of gravity or the orientation of an object. By selecting an adequate combination of moments, it is then possible to determine partitioned systems with good decoupling and linearizing properties [26], [27].

Note that for all these features (geometrical primitives, moments), the depth of the primitive or of the object considered appears in the coefficients of the interaction matrix related to the translational degrees of freedom, as was the case for the image points. An estimation of this depth is thus generally still necessary. Few exceptions occur, using for instance an adequate normalization of moments, which allows in some particular cases to make only the constant desired depth appear in the interaction matrix [27].

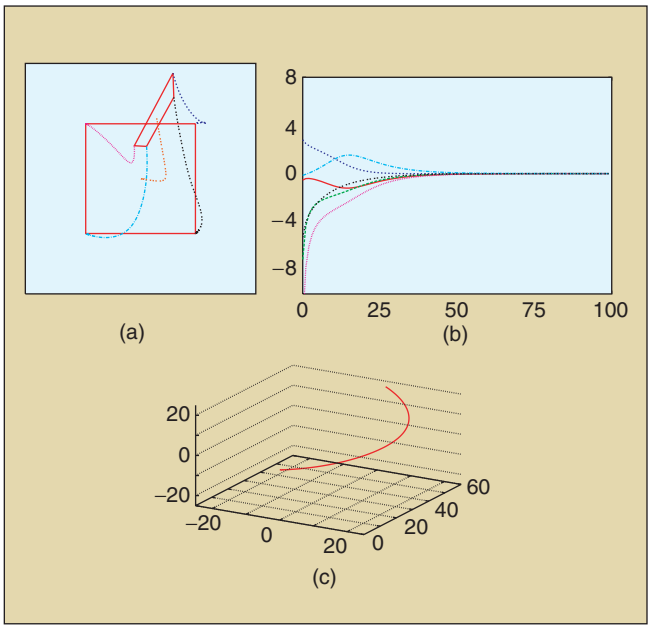


Figure 3. *System behavior using* $\mathbf{s} = (\rho_1, \theta_1, \ldots, \rho_4, \theta_4)$ *and* $\widehat{\mathbf{L}_\mathbf{e}^+} = \mathbf{L}_{\mathbf{e}^*}^+$.

Performance Optimization and Planning

In some sense, partitioned methods represent an effort to optimize system performance by assigning distinct features and controllers to individual degrees of freedom. In this way, the designer performs a sort of off-line optimization when allocating controllers to degrees of freedom. It is also possible to explicitly design controllers that optimize various system performance measures. We describe a few of these in this section.

Optimal Control and Redundancy Framework

An example of such an approach is given in [28] and [29], in which linear quadratic Gaussian (LQG) control design is used to choose gains that minimize a linear combination of state and control input. This approach explicitly balances the tradeoff between tracking errors (since the controller attempts to drive $\mathbf{s} - \mathbf{s}^*$ to zero) and robot motion. A similar control approach is proposed in [30] where joint limit avoidance is considered simultaneously with the positioning task.

It is also possible to formulate optimality criteria that explicitly express the observability of robot motion in the image. For example, the singular value decomposition of the interaction matrix reveals which degrees of freedom are most apparent and can thus be easily controlled, while the condition number of the interaction matrix gives a kind of global measure of the visibility of motion. This concept has been called resolvability in [31] and motion perceptibility in [32]. By selecting features and designing controllers that maximize these measures, either along specific degrees of freedom or globally, the performance of the visual servo system can be improved.

The constraints considered to design the control scheme using the optimal control approach may be contradictory in some cases, leading the system to fail due to local minima in the objective function to be minimized. For example, it may happen that the motion produced to move away from a robot joint limit is exactly the opposite of the motion produced to near the desired pose, which results in a zero global motion. To avoid this potential problem, it is possible to use the gradient projection method, which is classical in robotics. Applying this method to visual servoing has been proposed in [25] and [23]. The approach consists in projecting the secondary constraints $\mathbf{e}_s$ on the null space of the vision-based task $\mathbf{e}$ so that they have no effect on the regulation of $\mathbf{e}$ to 0

$$\mathbf{e}_g = \widehat{\mathbf{L}_\mathbf{e}^+}\mathbf{e} + \mathbf{P}_\mathbf{e}\,\mathbf{e}_s,$$

where $\mathbf{e}_g$ is the new global task considered and $\mathbf{P}_\mathbf{e} = (\mathbf{I}_6 - \widehat{\mathbf{L}_\mathbf{e}^+}\widehat{\mathbf{L}_\mathbf{e}})$ is such that $\widehat{\mathbf{L}_\mathbf{e}}\mathbf{P}_\mathbf{e}\mathbf{e}_s = 0, \forall \mathbf{e}_s$. Avoiding the robot joint limits using this approach has been presented in [33]. However, when the vision-based task constrains all the camera degrees of freedom, the secondary constraints cannot be considered since, when $\widehat{\mathbf{L}_\mathbf{e}}$ is of full rank 6, we have $\mathbf{P}_\mathbf{e}\mathbf{e}_s = 0, \forall \mathbf{e}_s$. In that case, it is necessary to inject the constraints in a global objective function, such as navigation functions which are free of local minima [34], [35].

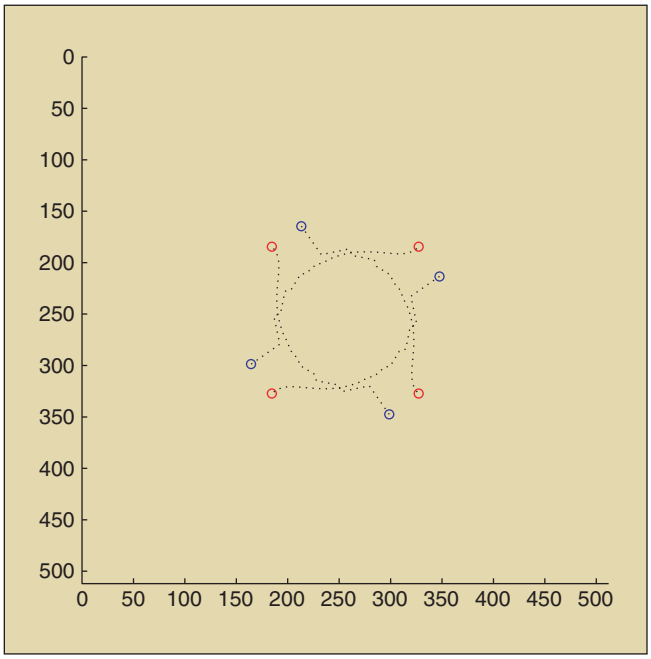


Figure 4. *Image feature trajectories for a rotation of* 160° *about the optical axis using a switched control scheme (initial points position in blue and desired points position in red).*

Switching Schemes

The partitioned methods described previously attempt to optimize performance by assigning individual controllers to specific degrees-of-freedom. Another way to use multiple controllers to optimize performance is to design switching schemes that select at each moment in time which controller to use based on criteria to be optimized.

A simple switching controller can be designed using an IBVS and a PBVS controller as follows [36]. Let the system begin by using the IBVS controller. Consider the Lyapunov function for the PBVS controller given by $\mathcal{L} = \frac{1}{2}\|\mathbf{e}(t)\|^2$, with $\mathbf{e}(t) = ({}^c\mathbf{t}_o - {}^{c^*}\mathbf{t}_o, \theta\mathbf{u})$. If at any time the value of this Lyapunov function exceeds a threshold γ_P, the system switches to the PBVS controller. While using the PBVS controller, if at any time the value of the Lyapunov function for the IBVS controller exceeds a threshold, $\mathcal{L} = \frac{1}{2}\|\mathbf{e}(t)\|^2 > \gamma_I$, the system switches to the IBVS controller. With this scheme, when the Lyapunov function for a particular controller exceeds a threshold, that controller is invoked, which in turn reduces the value of the corresponding Lyapunov function. If the switching thresholds are selected appropriately, the system is able to exploit the relative advantages of IBVS and PBVS, while avoiding their shortcomings.

An example for this system is shown in Figure 4, for the case of a rotation by 160° about the optical axis. Note that the system begins in IBVS mode and the features initially move on straight lines toward their goal positions in the image. However, as the camera retreats, the system switches to PBVS, which allows the camera to reach its desired position by combining a rotational motion around its optic axis and a forward translational motion, producing the circular trajectories observed in the image.

Other examples of temporal switching schemes can be found, such as for instance the one developed in [37] to ensure the visibility of the target observed.

Feature Trajectory Planning

It is also possible to treat the optimization problem offline, during a planning stage. In that case, several constraints can be simultaneously taken into account, such as obstacle avoidance [38], joint limit and occlusions avoidance, and ensuring the visibility of the target [5]. The feature trajectories $\mathbf{s}^*(t)$ that allow the camera to reach its desired pose while ensuring the constraints are satisfied are determined using path planning techniques, such as the well known potential field approach.

Coupling path planning with trajectory following also allows to improve significantly the robustness of the visual servo with respect to modeling errors. Indeed, modeling errors may have large effects when the error $\mathbf{s}-\mathbf{s}^*$ is large, but have few effects when $\mathbf{s}-\mathbf{s}^*$ is small. Once desired features trajectories $\mathbf{s}^*(t)$ such that $\mathbf{s}^*(0)=\mathbf{s}(0)$ have been designed during the planning stage, it is easy to adapt the control scheme to take into account the fact that $\mathbf{s}^*$ is varying, and to make the error $\mathbf{s}-\mathbf{s}^*$ remain small. More precisely, we now have

$$\dot{\mathbf{e}} = \dot{\mathbf{s}} - \dot{\mathbf{s}}^* = \mathbf{L}_\mathbf{e}\mathbf{v}_c - \dot{\mathbf{s}}^*,$$

from which we deduce, by selecting as usual $\dot{\mathbf{e}}=-\lambda\mathbf{e}$ as desired behavior

$$\mathbf{v}_c = -\lambda\widehat{\mathbf{L}_\mathbf{e}^+}\mathbf{e} + \widehat{\mathbf{L}_\mathbf{e}^+}\dot{\mathbf{s}}^*.$$

The new second term of this control law anticipates the variation of $\mathbf{s}^*$, removing the tracking error it would produce. We will see in the next section that the form of the control law is similar when tracking a moving target is considered.

Target Tracking

We now consider the case of a moving target and a constant desired value $\mathbf{s}^*$ for the features, the generalization to varying desired features $\mathbf{s}^*(t)$ being immediate. The time variation of the error is now given by

$$\dot{\mathbf{e}} = \dot{\mathbf{s}} = \mathbf{L}_\mathbf{e}\mathbf{v}_c + \frac{\partial\mathbf{e}}{\partial t}, \tag{10}$$

where the term $\frac{\partial\mathbf{e}}{\partial t}$ expresses the time variation of $\mathbf{e}$ due to the generally unknown target motion. If the control law is still designed to try to ensure an exponential decoupled decrease of $\mathbf{e}$ (that is, once again $\dot{\mathbf{e}}=-\lambda\mathbf{e}$), we now obtain using (10):

$$\mathbf{v}_c = -\lambda\widehat{\mathbf{L}_\mathbf{e}^+}\mathbf{e} - \widehat{\mathbf{L}_\mathbf{e}^+}\widehat{\frac{\partial\mathbf{e}}{\partial t}}, \tag{11}$$

where $\widehat{\frac{\partial\mathbf{e}}{\partial t}}$ is an estimation or an approximation of $\frac{\partial\mathbf{e}}{\partial t}$. This term must be introduced in the control law to compensate for the target motion.

Closing the loop, that is injecting (11) in (10), we obtain

A simple switching controller can be designed using an IBVS and a PBVS controller.

$$\dot{\mathbf{e}} = -\lambda\mathbf{L}_\mathbf{e}\widehat{\mathbf{L}_\mathbf{e}^+}\mathbf{e} - \mathbf{L}_\mathbf{e}\widehat{\mathbf{L}_\mathbf{e}^+}\widehat{\frac{\partial\mathbf{e}}{\partial t}} + \frac{\partial\mathbf{e}}{\partial t}. \tag{12}$$

Even if $\mathbf{L}_\mathbf{e}\widehat{\mathbf{L}_\mathbf{e}^+} > 0$, the error will converge to zero only if the estimation $\widehat{\frac{\partial\mathbf{e}}{\partial t}}$ is sufficiently accurate so that

$$\mathbf{L}_\mathbf{e}\widehat{\mathbf{L}_\mathbf{e}^+}\widehat{\frac{\partial\mathbf{e}}{\partial t}} = \frac{\partial\mathbf{e}}{\partial t}.$$

Otherwise, tracking errors will be observed. Indeed, by just solving the scalar differential equation $\dot{e}=-\lambda e+b$, which is a simplification of (12), we obtain $e(t)=e(0)\exp(-\lambda t)+b/\lambda$, which converges towards b/λ. On one hand, setting a high gain λ will reduce the tracking error, but on the other hand, setting the gain too high can make the system unstable. It is thus necessary to make b as small as possible.

Of course, if the system is known to be such that $\frac{\partial\mathbf{e}}{\partial t}=0$ (that is the camera observes a motionless object), no tracking error will appear with the most simple estimation given by $\widehat{\frac{\partial\mathbf{e}}{\partial t}}=0$. Otherwise, a classical method in automatic control to cancel tracking errors consists in compensating the target motion through an integral term in the control law. In that case, we have

$$\widehat{\frac{\partial\mathbf{e}}{\partial t}} = \mu\sum_j \mathbf{e}(j),$$

where μ is the integral gain, which must be tuned. This scheme allows to cancel the tracking errors only if the target has a constant velocity. Other methods, based on feedforward control, estimate directly the term $\widehat{\frac{\partial\mathbf{e}}{\partial t}}$ through the image measurements and the camera velocity, when it is available. Indeed, from (10), we obtain

$$\widehat{\frac{\partial\mathbf{e}}{\partial t}} = \widehat{\dot{\mathbf{e}}} - \widehat{\mathbf{L}_\mathbf{e}}\widehat{\mathbf{v}_c},$$

where $\widehat{\dot{\mathbf{e}}}$ can, for instance, be obtained as $\widehat{\dot{\mathbf{e}}}(t)=(\mathbf{e}(t)-\mathbf{e}(t-\Delta t))/\Delta t$, with Δt being the duration of the control loop. A Kalman filter [39] or more elaborate filtering methods [40] can then be used to improve the estimated values. If some knowledge about the target velocity or the target trajectory is available, it can of course be used to smooth or predict the motion [41]–[43]. For instance, in [44], the periodic motion of the heart and of the breath are compensated for in an application of visual servoing in medical robotics. Finally, other methods have been developed to remove as fast as possible the perturbations induced by the target motion [28], using for instance predictive controllers [45].

In the joint space, the system equations for both the eye-to-hand configuration and the eye-in-hand configuration have the same form.

Eye-in-Hand and Eye-to-Hand Systems Controlled in the Joint Space

In the previous sections, we considered the six components of the camera velocity as input of the robot controller. As soon as the robot is not able to realize this motion, for instance because it has less than six degrees of freedom, the control scheme must be expressed in the joint space. In this section, we describe how this can be done, and in the process develop a formulation for eye-to-hand systems.

In the joint space, the system equations for both the eye-to-hand configuration and the eye-in-hand configuration have the same form

$$\dot{\mathbf{s}} = \mathbf{J_s}\,\dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t}. \tag{13}$$

Here, $\mathbf{J_s} \in \mathrm{R}^{k\times n}$ is the feature Jacobian matrix, which can be linked to the interaction matrix, and n is the number of robot joints.

For an eye-in-hand system [see Figure 5(b)], $(\partial \mathbf{s}/\partial t)$ is the time variation of $\mathbf{s}$ due to a potential object motion, and $\mathbf{J_s}$ is given by

$$\mathbf{J_s} = \mathbf{L_s}\,{}^{c}\mathbf{V}_N \mathbf{J}(\mathbf{q}), \tag{14}$$

where

- ${}^{c}\mathbf{V}_N$ is the spatial motion transform matrix (as defined in Part I of the tutorial) from the vision sensor frame to the end effector frame. It is usually a constant matrix (as soon as the vision sensor is rigidly attached to the end effector). Thanks to the robustness of closed loop control schemes, a coarse approximation of this transform matrix is sufficient in visual servoing. If needed, an accurate estimation is possible through classical hand-eye calibration methods [46].
- $\mathbf{J}(\mathbf{q})$ is the robot Jacobian expressed in the end effector frame.

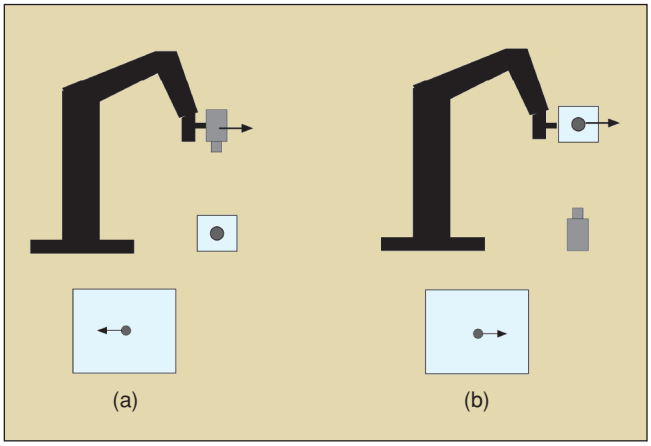


Figure 5. *Top: (a) Eye-in-hand system. (b) Eye-to-hand system. Bottom: Opposite image motion produced by the same robot motion.*

For an eye-to-hand system [see Figure 5(b)], $(\partial \mathbf{s}/\partial t)$ is now the time variation of $\mathbf{s}$ due to a potential vision sensor motion and $\mathbf{J_s}$ can be expressed as

$$\mathbf{J_s} = -\mathbf{L_s}\,{}^{c}\mathbf{V}_N\,{}^{N}\mathbf{J}(\mathbf{q}) \tag{15}$$

$$= -\mathbf{L_s}\,{}^{c}\mathbf{V}_0\,{}^{0}\mathbf{J}(\mathbf{q}). \tag{16}$$

In (15), the classical robot Jacobian ${}^{N}\mathbf{J}(\mathbf{q})$ expressed in the end effector frame is used but the spatial motion transform matrix ${}^{c}\mathbf{V}_N$ from the vision sensor frame to the end effector frame changes all along the servo, and it has to be estimated at each iteration of the control scheme, usually using pose estimation methods.

In (16), the robot Jacobian ${}^{0}\mathbf{J}(\mathbf{q})$ is expressed in the robot reference frame and the spatial motion transform matrix ${}^{c}\mathbf{x}_0$ from the vision sensor frame to that reference frame is constant as long as the camera does not move. In that case, which is convenient in practice, a coarse approximation of ${}^{c}\mathbf{V}_0$ is usually sufficient.

Once the modeling step is finished, it is quite easy to follow the procedure that has been used above to design a control scheme expressed in the joint space, and to determine sufficient condition to ensure the stability of the control scheme. We obtain, considering again $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$, and an exponential decoupled decrease of $\mathbf{e}$

$$\dot{\mathbf{q}} = -\lambda \widehat{\mathbf{J}_\mathbf{e}^{+}}\mathbf{e} - \widehat{\mathbf{J}_\mathbf{e}^{+}}\frac{\widehat{\partial \mathbf{e}}}{\partial t}. \tag{17}$$

If $k = n$, considering the Lyapunov function $\mathcal{L} = \frac{1}{2}\|\mathbf{e}(t)\|^2$, a sufficient condition to ensure the global asymptotic stability is given by

$$\mathbf{J_e}\widehat{\mathbf{J}_\mathbf{e}^{+}} > 0. \tag{18}$$

If $k > n$, we obtain by following the developments of the stability analysis of basic IBVS (see Part I of the tutorial):

$$\widehat{\mathbf{J}_\mathbf{e}^{+}}\mathbf{J_e} > 0 \tag{19}$$

to ensure the local asymptotic stability of the system. Note that the actual extrinsic camera parameters appears in $\mathbf{J_e}$, while the estimated ones are used in $\widehat{\mathbf{J}_\mathbf{e}^{+}}$. It is thus possible to analyse the robustness of the control scheme with respect to the camera extrinsic parameters. It is also possible to estimate directly the numerical value of $\mathbf{J_e}$ or $\mathbf{J_e^{+}}$ using, as described previously, the Broyden update rule.

Finally, to remove tracking errors, we have to ensure that

$$\mathbf{J_e}\widehat{\mathbf{J}_\mathbf{e}^+}\widehat{\frac{\partial \mathbf{e}}{\partial t}} = \frac{\partial \mathbf{e}}{\partial t}.$$

Finally, let us note that, even if the robot has six degrees of freedom, it is generally not equivalent to first compute $\mathbf{v}_c$ using (2) and then deduce $\dot{\mathbf{q}}$ using the robot inverse Jacobian, and to compute directly $\dot{\mathbf{q}}$ using (17). Indeed, it may occur that the robot Jacobian $\mathbf{J}(\mathbf{q})$ is singular while the feature Jacobian $\mathbf{J_s}$ is not (that may occur when $k < n$). Furthermore, the properties of the pseudo-inverse ensure that using (2), $\|\mathbf{v}_c\|$ is minimal while using (17), $\|\dot{\mathbf{q}}\|$ is minimal. As soon as $\mathbf{J}_\mathbf{e}^+ \neq \mathbf{J}^+(\mathbf{q})^N\mathbf{V}_c\mathbf{L}_\mathbf{e}^+$, the control schemes will be different and will induce different robot trajectories. The choice of the state-space is thus important.

Conclusion

In this tutorial, we have only considered velocity controllers. It is convenient for most of classical robot arms. However, the dynamics of the robot must of course be taken into account for high speed tasks, or when we deal with mobile nonholonomic or underactuated robots. As for the sensor, we have only considered geometrical features coming from a classical perspective camera. Features related to the image motion or coming from other vision sensors (fish-eye camera, catadioptric camera, echographic probes, etc.) necessitate to revisit the modeling issues to select adequate visual features. Finally, fusing visual features with data coming from other sensors (force sensor, proximetry sensors, etc.) at the level of the control scheme will allow to address new research topics. Numerous ambitious applications of visual servoing can also be considered, as well as for mobile robots in indoor or outdoor environments, for aerial, space, and submarine robots, and in medical robotics. The end of fruitful research in the field of visual servo is thus nowhere yet in sight.

Numerous ambitious applications of visual servoing can be considered.

Keywords

Visual servo, robot control.

References

[1] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. New York: Springer-Verlag, 2003.

[2] R. Basri, E. Rivlin, and I. Shimshoni, "Visual homing: Surfing on the epipoles," *Int. J. Comput. Vision*, vol. 33, pp. 117–137, Sept. 1999.

[3] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *Int. J. Comput. Vision*, vol. 37, pp. 79–97, June 2000.

[4] O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint*. Cambridge, MA: MIT Press, 1993.

[5] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. Robot. Automa.*, vol. 18, no. 4 pp. 534–549, Aug. 2002.

[6] I. Suh, "Visual servoing of robot manipulators by fuzzy membership function based neural networks," in *Visual Servoing,* K. Hashimoto, Ed. (World Scientific Series in Robotics and Automated Systems). Singapore: World Scientific Press, 1993, vol. 7, pp. 285–315.

[7] G. Wells, C. Venaille, and C. Torras, "Vision-based robot positioning using neural networks," *Image Vision Comput.*, vol. 14, pp. 75–732, Dec. 1996.

[8] J.-T. Lapresté, F. Jurie, and F. Chaumette, "An efficient method to compute the inverse jacobian matrix in visual servoing," in *Proc. IEEE Int. Conf. Robotics Automation*, New Orleans, LA, Apr. 2004, pp. 727–732.

[9] K. Hosada and M. Asada, "Versatile visual servoing without knowledge of true jacobian," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, Munchen, Germany, Sept. 1994, pp. 186–193.

[10] M. Jägersand, O. Fuentes, and R. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," in *Proc. IEEE Int. Conf. Robotics and Automation*, Albuquerque, NM, Apr. 1997, pp. 2874–2880.

[11] J. Piepmeier, G.M. Murray, and H. Lipkin, "Uncalibrated dynamic visual servoing," *IEEE Trans. Robot. Automat.*, vol. 20, no. 1, pp. 143–147, Feb. 2004.

[12] K. Deguchi, "Direct interpretation of dynamic images and camera motion for visual servoing without image feature correspondence," *J. Robot. Mechatron.*, vol. 9, no. 2, pp. 104–110, 1997.

[13] E. Malis, F. Chaumette, and S. Boudet, "2-1/2D visual servoing," *IEEE Trans. Robot. Automat.*, vol. 15, no. 2, pp. 238–250, Apr. 1999.

[14] E. Malis and F. Chaumette, "Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods," *IEEE Trans. Robot. Automat.*, vol. 18, no. 2, pp. 176–186, Apr. 2002.

[15] J. Chen, D. Dawson, W. Dixon, and A. Behal, "Adaptive homography-based visual servo tracking for fixed and camera-in-hand configurations," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 9, pp. 814–825, Sept. 2005.

[16] G. Morel, T. Leibezeit, J. Szewczyk, S. Boudet, and J. Pot, "Explicit incorporation of 2D constraints in vision-based control of robot manipulators," in *Proc. Int. Symp. Experimental Robotics*, 2000, vol. 250 LNCIS Series, pp. 99–108.

[17] F. Chaumette and E. Malis, "2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings," in *Proc. IEEE Int. Conf. Robotics Automation*, San Fransisco, CA, Apr. 2000, pp. 630–635.

[18] E. Cervera, A.D. Pobil, F. Berry, and P. Martinet, "Improving image-based visual servoing with three-dimensional features," *Int. J. Robot. Res.*, vol. 22, pp. 821–840, Oct. 2004.

[19] F. Schramm, G. Morel, A. Micaelli, and A. Lottin, "Extended-2D visual servoing," in *Proc. IEEE Int. Conf. Robotics Automation*, New Orleans, LA, Apr. 2004, pp. 267–273.

[20] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Trans. Robotics Automation*, vol. 17, no. 4, pp. 507–515, Aug. 2001.

[21] M. Iwatsuki and N. Okiyama, "A new formulation of visual servoing based on cylindrical coordinate system," *IEEE Trans. Robot. Automat.*, vol. 21, no. 2, pp. 266–273, Apr. 2005.

[22] F. Chaumette, P. Rives, and B. Espiau, "Classification and realization of the different vision-based tasks," in *Visual Servoing,* K. Hashimoto, Ed. (Robotics and Automated Systems). Singapore: World Scientific, 1993, vol. 7, pp. 199–228.

[23] A. Castano and S. Hutchinson, "Visual compliance: task directed visual servo control," *IEEE Trans. Robot. Automat.*, vol. 10, no. 3, pp. 334–342, June 1994.

[24] G. Hager, "A modular system for robust positioning using feedback from stereo vision," *IEEE Trans. Robot. Automat.*, vol. 13, no. 4, pp. 582–595, Aug. 1997.

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

[26] F. Chaumette, "Image moments: A general and useful set of features for visual servoing," *IEEE Trans. Robot. Automat.*, vol. 20, no. 4, pp. 713–723, Aug. 2004.

[27] O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1116–1127, Dec. 2005.

[28] N. Papanikolopoulos, P. Khosla, and T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: A combination of vision and control," *IEEE Trans. Robot. Automat.*, vol. 9, no. 1, pp. 14–35, Feb. 1993.

[29] K. Hashimoto and H. Kimura, "LQ optimal and nonlinear approaches to visual servoing," in *Visual Servoing,* K. Hashimoto, Ed. (Robotics and Automated Systems). Singapore: Wolrd Scientific, 1993, vol. 7, pp. 165–198.

[30] B. Nelson and P. Khosla, "Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance," *Int. J. Robot. Res.*, vol. 14, pp. 225–269, June 1995.

[31] B. Nelson and P. Khosla, "Force and vision resolvability for assimilating disparate sensory feedback," *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 714–731, Oct. 1996.

[32] R. Sharma and S. Hutchinson, "Motion perceptibility and its application to active vision-based servo control," *IEEE Trans. Robot. Automat.*, vol. 13, no. 4, pp. 607–617, Aug. 1997.

[33] E. Marchand, F. Chaumette, and A. Rizzo, "Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, Osaka, Japan, Nov. 1996, pp. 1083–1090.

[34] E. Marchand and G. Hager, "Dynamic sensor planning in visual servoing," in *Proc. IEEE Int. Conf. Robotics Automation*, Leuven, Belgium, May 1998, pp. 1988–1993.

[35] N. Cowan, J. Weingarten, and D. Koditschek, "Visual servoing via navigation functions," *IEEE Trans. Robot. Automat.*, vol. 18, no. 4, pp. 521–533, Aug. 2002.

[36] N. Gans and S. Hutchinson, "An asymptotically stable switched system visual controller for eye in hand robots," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, Las Vegas, NV, Oct. 2003, pp. 735–742.

[37] G. Chesi, K. Hashimoto, D. Prattichizio, and A. Vicino, "Keeping features in the field of view in eye-in-hand visual servoing: A switching approach," *IEEE Trans. Robot. Automat.*, vol. 20, no. 5, pp. 908–913, Oct. 2004.

[38] K. Hosoda, K. Sakamato, and M. Asada, "Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3D reconstruction," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, vol. 3, Pittsburgh, PA, Aug. 1995, pp. 29–34.

[39] P. Corke and M. Goods, "Controller design for high performance visual servoing," in *Proc. 12th World Congr. IFAC'93*, Sydney, Australia, July 1993, pp. 395–398.

[40] F. Bensalah and F. Chaumette, "Compensation of abrupt motion changes in target tracking by visual servoing," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, Pittsburgh, PA, Aug. 1995, pp. 181–187.

[41] P. Allen, B. Yoshimi, A. Timcenko, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Trans. Robot. Automat.*, vol. 9, no. 2, pp. 152–165, Apr. 1993.

[42] K. Hashimoto and H. Kimura, "Visual servoing with non linear observer," in *Proc. IEEE Int. Conf. Robotics Automation*, Nagoya, Japan, Apr. 1995, pp. 484–489.

[43] A. Rizzi and D. Koditschek, "An active visual estimator for dexterous manipulation," *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 697–713, Oct. 1996.

[44] R. Ginhoux, J. Gangloff, M. de Mathelin, L. Soler, M.A. Sanchez, and J. Marescaux, "Active filtering of physiological motion in robotized surgery using predictive control," *IEEE Trans. Robot.*, vol. 21, no. 1, pp. 67–79, Feb. 2005.

[45] J. Gangloff and M. de Mathelin, "Visual servoing of a 6 dof manipulator for unknown 3-D profile following," *IEEE Trans. Robot. Automat.*, vol. 18, no. 4, pp. 511–520, Aug. 2002.

[46] R. Tsai and R. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand-eye calibration," *IEEE Trans. Robot. Automat.*, vol. 5, no. 3, pp. 345–358, June 1989.

François Chaumette graduated from École Nationale Supérieure de Mécanique, Nantes, France, in 1987. He received the Ph.D. degree in computer science from the University of Rennes, Rennes, France, in 1990. Since 1990, he has been with IRISA in Rennes where he is now "Directeur de Recherches" INRIA and head of the Lagadic group. His research interests include robotics and computer vision, especially visual servoing and active perception. He has coauthored more than 150 papers on the topics of robotics and computer vision, and has served in the last five years on the program committees for the main conferences related to robotics and computer vision. Dr, Cahumette received the AFCET/CNRS Prize for the best French thesis in automatic control in 1991. He also received with Ezio Malis the 2002 King-Sun Fu Memorial Best IEEE Transactions on Robotics and Automation Paper Award. He was Associate Editor of *IEEE Transactions on Robotics* from 2001 to 2005.

Seth Hutchinson received the Ph.D. degree from Purdue University, West Lafayette, IN, in 1988. In 1990, he joined the faculty at the University of Illinois in Urbana-Champaign, where he is currently a Professor in the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Beckman Institute for Advanced Science and Technology. Dr. Hutchinson serves as the Editor-in-Chief for the RAS Conference Editorial Board, and on the editorial boards of the *International Journal of Robotics Research* and the *Journal of Intelligent Service Robotics*. He served as Associate and then Senior Editor for the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, now the IEEE TRANSACTIONS ON ROBOTICS, from 1997 to 2005. In 1996, he was a Guest Editor for a special section of the TRANSACTIONS devoted to the topic of visual servo control, and in 1994 he was Co-Chair of an IEEE Workshop on Visual Servoing. In 1996 and 1998, he coauthored papers that were finalists for the King-Sun Fu Memorial Best Transactions Paper Award. He was Co-Chair of IEEE Robotics and Automation Society Technical Committee on Computer and Robot Vision from 1992 to 1996, and has served on the program committees for more than fifty conferences related to robotics and computer vision. He has published more than 100 papers on the topics of robotics and computer vision, and is coauthor of the books *Principles of Robot Motion: Theory, Algorithms, and Implementations* (Cambrudge, MA: MIT Press) and *Robot Modeling and Control* (New York: Wiley).

Address for Correspondence: François Chaumette, IRISA/INRIA Rennes, Campus de Beaulieu, 35 042 Rennes cedex, France. E-mail: Francois.Chaumette@irisa.fr.