Chapter 6

Visual Servoing

6.1. Introduction

Robotic systems are more and more often equipped with exteroceptive sensors which, by definition, provide information on the environment in which they operate. These sensors are of course essential when a task has to be performed in an environment that is not completely rigid or not perfectly well known. They also make it possible to consider errors or inaccuracies that may occur in the manipulator robot's identification of geometric (and therefore kinematic) models. Aside from force sensors, the purpose and applications of which were discussed in the previous chapter, there are other sensors available that provide localization of the system in its environment, or give it a generally local perception of its surroundings. To give a few examples, road marking, passive beacon or radio-based systems, as well as GPS, all make it possible to localize a mobile robot, by determining either its absolute position or its movement. When it comes to perception, proximity sensors provide information on the distances to the closest objects. They are therefore particularly well suited for obstacle avoiding tasks. As for computer-assisted vision and telemetry systems, they have a rather wide range of applications since they can be used for localization, navigation, and exploration.

For a long time 3-D reconstruction was considered an unavoidable, independent module, a prerequisite to any movement generating module for a robot in a not perfectly well known environment. In computer-assisted vision, this state of things, which used to be justified by the prohibitive computation time required by image processing algorithms, led to a number of successful studies, notably in the field of 3-D

Chapter written by François CHAUMETTE.

vision [FAU 93, MA 03]. The algorithmic and technological progress achieved over the past 15 years has made it possible to more closely intertwine the aspects of perception with those of action, by directly integrating the measurements provided by a vision system into closed-loop control laws that deal with the extracted visual information. This approach, known as visual servoing, shares some aspects with the studies on sensory control and is the focus of this chapter.

Visual servoing techniques consist of using information provided by one or several cameras, in order to control the movements of a robotic system. This allows for a wide variety of tasks designed to locate a system with respect to its environment, or to follow mobile objects, by controlling one or as many as all of the system's n degrees of freedom. Whatever the sensor's configuration, which can range from a camera mounted on the robot's effector to several scene cameras, the objective is to select as best as possible a set of k visual data, in order to control the m desired degrees of freedom, and to develop a control law so as to make these data $\mathbf{s}(t)$ reach a desired value $\mathbf{s}^*$ that defines when a task is suitably achieved. It is also possible to track a trajectory $\mathbf{s}^*(t)$. The idea of control therefore amounts to regulating (that is to say making a value reach zero and maintaing it there) the error vector $\mathbf{s}(t) - \mathbf{s}^*(t)$.

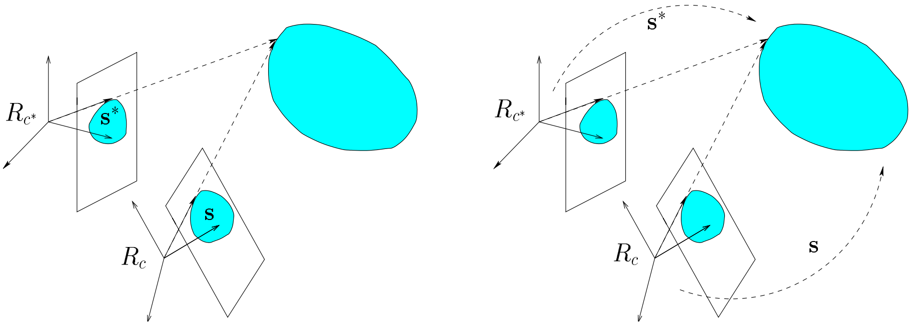


Figure 6.1. *2-D and 3-D visual servoing: 2-D visual servoing is used to bring the camera's frame of reference from R_c to R_{c^*}, based on measurements $\mathbf{s}$ extracted directly from the image (left). With 3-D visual servoing, the measurements $\mathbf{s}$ represent 3-D data estimated after an image processing phase (right).*

With a vision sensor, which is supposed to provide 2-D measurements, the nature of the potential information is extremely rich, since it is possible to imagine visual servoing of both 2-D data, such as the coordinates of characteristic points in the image for example, and 3-D data, provided by a positioning module operating on the extracted 2-D measurements (see Figure 6.1). This wide range of possibilities is the reason behind the major difficulty in visual servoing, that is to build and select as best as possible the visual data needed for a suitable behavior of the system, based on all the available measurements. A number of qualities are important: local or even global stability, robust behavior when facing measurement or modeling errors, absence of singularities and local minima, suitable trajectories for the robot, but also for the visual information in the image, and finally a maximum decoupling between the visual information and the controlled degrees of freedom.

To study the behavior of the resulting system, a model is necessary to describe the relation between the visual data $\mathbf{s}(t)$ that were chosen and the control variables. This essential phase of model design will now be described. However, in this chapter we will not be dealing with aspects of image processing, crucial to extracting useful 2-D data from a digital image and following them at each iteration of the control law. For readers interested in knowing more, we suggest turning to works specializing in this field [VIN 00, KRA 05].

6.2. Modeling visual data

6.2.1. *The interaction matrix*

In order to be taken into account in a visual servoing diagram, a set $\mathbf{s}$ of k visual data simply needs to be defined by an application differentiable from SE_3 into $\mathbb{R}^k$:

$$\mathbf{s} = \mathbf{s}(\mathbf{p}(t)) \qquad [6.1]$$

where $\mathbf{p}(t)$, an element of the space of reference frames and rigid bodies SE_3, describes the pose at the instant t between the camera and its environment. Hence only the movements of the camera, or of the objects it perceives, can modify the value of a visual datum.

The differential of $\mathbf{s}$ allows us to see how the variations in the visual data are related to the relative movements between the camera and the scene, since by differentiating [6.1], we get:

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{p}}\, \dot{\mathbf{p}} = \mathbf{L_s}\, \mathbf{v} \qquad [6.2]$$

where:

– $\mathbf{L_s}$ is a $k \times 6$ matrix, referred to as the *interaction matrix* associated with $\mathbf{s}$;

– $\mathbf{v}$ is the relative kinematic torsor between the camera and the scene, expressed in the camera's frame of reference R_c in its origin C. More accurately, if $\mathbf{v_c}$ and $\mathbf{v_o}$ are, respectively, the kinematic torsors of the camera and of the scene it perceives, both expressed in R_c and in C, then let:

$$\mathbf{v} = \mathbf{v_c} - \mathbf{v_o} \qquad [6.3]$$

From now on, except if noted otherwise, we will write, incorrectly, that a torsor expressed in a frame of reference has its value given in the origin of this frame. Also, we will denote by $\boldsymbol{\upsilon}$ the translation speed at the origin of the coordinate system, and by $\boldsymbol{\omega}$ the rotation speed, such that $\mathbf{v} = (\boldsymbol{\upsilon}, \boldsymbol{\omega})$. If ${}^{o}\mathbf{R}_{c}$ describes the rotation matrix from the frame of reference R_o bound to the object to R_c, we have by definition [SAM 91]:

$$[\boldsymbol{\omega}]_{\times} = {}^{o}\dot{\mathbf{R}}_{c}{}^{o}\mathbf{R}_{c}^{\top} = -{}^{c}\dot{\mathbf{R}}_{o}{}^{c}\mathbf{R}_{o}^{\top} = {}^{c}\mathbf{R}_{o}{}^{o}\dot{\mathbf{R}}_{c} \qquad [6.4]$$

where $[\boldsymbol{\omega}]_{\times}$ is the antisymmetric matrix of the vector preproduct associated with $\boldsymbol{\omega}$.

COMMENT.– In more formal terms [SAM 91], the transpose of the interaction matrix can be defined as the matrix representation of the subspace generated by a family of k torsors expressed in R_c. This is due to the fact that each component of $\mathbf{s}$ can be decomposed as the product of two torsors, one called the interaction torsor, and the other being of course the kinematic torsor. We will see the practical advantage of this definition in section 6.3.3.1.

6.2.2. *Mounted camera*

If we consider a camera mounted on the effector of a manipulator robot observing a fixed object, the relation between $\dot{\mathbf{s}}$ and the speed of the robot's joint variables $\dot{\mathbf{q}}$ can easily be obtained:

$$\dot{\mathbf{s}} = \mathbf{J}_{\mathbf{s}}\dot{\mathbf{q}} = \mathbf{L}_{\mathbf{s}}{}^{c}\mathbf{V}_{n}{}^{n}\mathbf{J}_{n}(\mathbf{q})\ \dot{\mathbf{q}} \qquad [6.5]$$

where $\mathbf{J}_{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}{}^{c}\mathbf{V}_{n}{}^{n}\mathbf{J}_{n}$ is the Jacobian of the visual data and where:

– ${}^{n}\mathbf{J}_{n}(\mathbf{q})$ is the robot's Jacobian expressed in the frame of reference R_n of its end-effector [KHAL 02];

– ${}^{c}\mathbf{V}_{n}$ is the kinematic torsor's transformation matrix from the camera's frame R_c to frame R_n. This matrix, which remains constant if the camera is rigidly fixed to the robot's end-effector, is given by [KHAL 02]:

$${}^{c}\mathbf{V}_{n} = \begin{bmatrix} {}^{c}\mathbf{R}_{n} & [{}^{c}\mathbf{t}_{n}]_{\times}{}^{c}\mathbf{R}_{n} \\ \mathbf{0}_{3} & {}^{c}\mathbf{R}_{n} \end{bmatrix} \qquad [6.6]$$

where ${}^{c}\mathbf{R}_{n}$ and ${}^{c}\mathbf{t}_{n}$ are, respectively, the rotation matrix and the translation vector from frame R_c to frame R_n. The elements of the transformation matrix from the camera's frame to the end-effector's frame can be estimated quite accurately by first using a technique called hand-eye calibration [TSA 89, HOR 95]. Visual servoing techniques are usually rather robust in admitting important modeling errors, both in this transformation matrix [ESP 93, MAL 02] and in the robot's Jacobian.

More generally, if a mounted camera is observing a moving object, the differential of $\mathbf{s}$ is then given by:

$$\dot{\mathbf{s}} = \mathbf{L_s}\,{}^{c}\mathbf{V}_n\,{}^{n}\mathbf{J}_n(\mathbf{q})\;\dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} \tag{6.7}$$

where the term $\frac{\partial \mathbf{s}}{\partial t}$ represents the variation of $\mathbf{s}$ caused by the object's own movement (which is usually not known). In the highly unlikely event that the object's movement is known, and given for example by the kinematic torsor $\mathbf{v_o}$ in R_c, in the end we get:

$$\dot{\mathbf{s}} = \mathbf{L_s}\,{}^{c}\mathbf{V}_n\,{}^{n}\mathbf{J}_n(\mathbf{q})\;\dot{\mathbf{q}} - \mathbf{L_s}\mathbf{v_o} \tag{6.8}$$

6.2.3. *Scene camera*

Likewise, if we now consider a scene camera observing the end-effector of a manipulator robot, the variation of the visual data rigidly related to this end-effector is expressed according to the speed of the joint coordinates:

$$\dot{\mathbf{s}} = -\mathbf{L_s}\,{}^{c}\mathbf{V}_n\,{}^{n}\mathbf{J}_n(\mathbf{q})\;\dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} \tag{6.9}$$

where $\frac{\partial \mathbf{s}}{\partial t}$ now describes the variations of $\mathbf{s}$ caused by a possible movement of the scene camera.

COMMENT.– Notice the difference in signs between Equations [6.5] and [6.9]. This difference is of course caused by the configuration change of the sensor with respect to the control variables (see Figure 6.2).

Whether the scene camera is fixed or mobile, the matrix ${}^{c}\mathbf{V}_n$ is now variable and has to be estimated at each iteration, which is usually done using a 3-D positioning technique (see section 6.2.5.1). If the camera is fixed, it is therefore better to use one of the following relations:

$$\begin{aligned} \dot{\mathbf{s}} &= -\mathbf{L_s}\,{}^{c}\mathbf{V}_\emptyset\,{}^{\emptyset}\mathbf{V}_n\,{}^{n}\mathbf{J}_n(\mathbf{q})\;\dot{\mathbf{q}} && [6.10] \\ &= -\mathbf{L_s}\,{}^{c}\mathbf{V}_\emptyset \begin{bmatrix} \mathbb{I}_3 & [{}^{\emptyset}\mathbf{t}_n]_\times \\ \mathbf{0}_3 & \mathbb{I}_3 \end{bmatrix} {}^{\emptyset}\mathbf{J}_n(\mathbf{q})\;\dot{\mathbf{q}} && [6.11] \end{aligned}$$

where ${}^{\emptyset}\mathbf{J}_n(\mathbf{q})$ is the robot's Jacobian expressed in its basic frame of reference and where the values of ${}^{\emptyset}\mathbf{V}_n$ and ${}^{\emptyset}\mathbf{t}_n$ are provided by the robot's direct geometric model. This is possible because the transformation matrix ${}^{c}\mathbf{V}_{\emptyset}$ is then constant and only has to be estimated once beforehand, usually crudely.

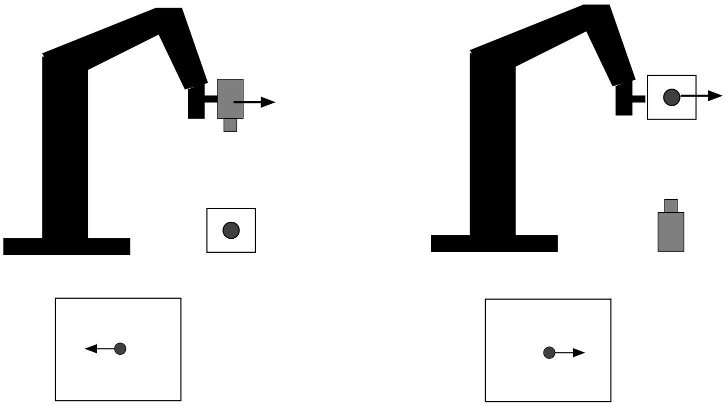


Figure 6.2. *Difference in configurations (top) and in effects produced in the image acquired by the camera (bottom)*

In the literature [HAS 93a, HUT 96], most studies are set in the context of a mounted sensor. We can however cite [ALL 93, NEL 94a, HAG 95, KEL 96, CIP 97, HOR 98, RUF 99] in which one or several scene cameras are used.

In any case, the interaction matrix plays an essential role and we will now give its analytical form for a set of visual data of geometric nature. From now on, all the necessary quantities (coordinates and speeds of points, kinematic torsor, etc.) are expressed in the camera's frame shown in Figure 6.3.

6.2.4. *2-D data interaction matrix*

6.2.4.1. *Interaction matrix of a 2-D point*

The typical mathematical model for a camera is defined by a perspective projection, such that any point M with coordinates $\mathbf{X} = (X, Y, Z)$ is projected onto the image plane in a point m with coordinates $\mathbf{x} = (x, y)$ with:

$$x = X/Z \quad , \quad y = Y/Z \qquad [6.12]$$

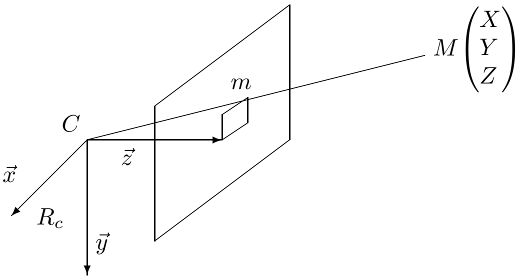


Figure 6.3. *Camera model*

By differentiating this equation, we get the variations in the image of the coordinates x and y of m with respect to the speed $\dot{\mathbf{X}}$ of the coordinates of point M:

$$\dot{\mathbf{x}} = \begin{bmatrix} 1/Z & 0 & -X/Z^2 \\ 0 & 1/Z & -Y/Z^2 \end{bmatrix} \dot{\mathbf{X}} \qquad [6.13]$$

Whatever configuration is chosen (mounted or scene camera, fixed or mobile point M), the speed $\dot{\mathbf{X}}$ of M according to the kinematic torsor $\mathbf{v}$ between the camera and its environment is given by the fundamental kinematics equation:

$$\dot{\mathbf{X}} = -\boldsymbol{v} - \boldsymbol{\omega} \times \mathbf{X} = -\boldsymbol{v} + [\mathbf{X}]_\times \, \boldsymbol{\omega} = \begin{bmatrix} -\mathbb{I}_3 & [\mathbf{X}]_\times \end{bmatrix} \mathbf{v} \qquad [6.14]$$

Equation [6.13] can then be simplifed using Equation [6.12], written in the form:

$$\dot{\mathbf{x}} = \mathbf{L_x}(\mathbf{x}, Z)\ \mathbf{v} \qquad [6.15]$$

where:

$$\mathbf{L_x}(\mathbf{x}, Z) = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \qquad [6.16]$$

Notice that the terms induced by rotation movements only depend on the measurements of x and y in the image. On the other hand, terms induced by translation movements are inversely proportional to the depth of the 3-D point in question. This effect occurs for all the visual data that can be defined in the image (and describes the classic ambiguity in computer-assisted vision between the amplitude of a translation movement and the depth of objects). In visual servoing, it is therefore necessary to insert a 3-D datum, even though it is known beforehand, whenever trying to control a robot's degrees of freedom that imply translation movements.

Image processing algorithms provide data expressed in pixels. If we ignore strongly non-linear effects of distortion, due for example to the use of short focal length lenses, the variable change when switching from the coordinates $\mathbf{x}_p = (x_p, y_p)$ of a point, expressed in pixels, to the coordinates $\mathbf{x}$ of this same point, but expressed in meters, is given by:

$$x = (x_p - x_c)/f_x \quad , \quad y = (y_p - y_c)/f_y \qquad [6.17]$$

where (x_c, y_c) represents the main point's coordinates in the image and where $f_x = f/l_x$ and $f_y = f/l_y$ are the ratios between the focal length f of the lens and the dimensions l_x and l_y of a pixel. These parameters, referred to as the intrinsic parameters of the camera, can be estimated beforehand, during a calibration phase [TSA 87, BEY 92, ZHA 00], but as with the elements of the hand-eye matrix, crude approximations are usually sufficient to maintain the stability of visual servoing systems [ESP 93, MAL 99, MAL 02, DEN 02].

It is also possible to calculate the interaction matrix associated with the coordinates of a point directly expressed in pixels. Using the reciprocal of the variable change in [6.17], given by:

$$x_p = x_c + f_x x \quad , \quad y_p = y_c + f_y y \qquad [6.18]$$

we immediately get:

$$\mathbf{L}_{\mathbf{x}_p} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \mathbf{L}_{\mathbf{x}} \qquad [6.19]$$

where the set of terms contained in $\mathbf{L}_{\mathbf{x}}$, except of course for the depth Z, can be expressed as functions of the intrinsic parameters and coordinates $\mathbf{x}_p$ using [6.17]. If need be, the same can be done for the visual data seen later on, working with data expressed in pixels. The main advantage of having an analytical form of the interaction matrix that explicitly depends on the intrinsic parameters, is that it then becomes possible to study how sensitive visual servoing systems are to errors made in the estimation or approximation of these parameters.

Finally, we should mention the studies in projective geometry described in [RUF 99] which led to a direct modeling of the Jacobian matrix $\mathbf{J_s}$ such that $\dot{\mathbf{s}} = \mathbf{J_s}\dot{\mathbf{q}}$, in the case where $\mathbf{s}$ is comprised of the coordinates of a point located on the end-effector of a robot and observed by two scene cameras: $\mathbf{s} = (x_g, y_g, x_d, y_d)$. The advantage of such an approach is that it is no longer necessary to know the Jacobian, and hence the geometric model, of the robot being used.

If we now consider a camera equipped with a controllable zoom, thus providing the system with an additional degree of freedom, we get just as simply, from [6.18]:

$$\begin{bmatrix} \dot{x_p} \\ \dot{y_p} \end{bmatrix} = \mathbf{L}_{\mathbf{x}_p}\, \mathbf{v} + \begin{bmatrix} (x_p - x_c)/f \\ (y_p - y_c)/f \end{bmatrix} \dot{f} \qquad [6.20]$$

For purely technological reasons (because for most zooms, position can be controlled, not speed), few studies have used this function, even though it provides an interesting redundancy with respect to the translation movement along the optical axis. We can still mention [HOS 95a, BEN 03].

6.2.4.2. *Interaction matrix of a configurable 2-D geometric primitive*

It is also possible to calculate the interaction matrix associated with visual data constructed from configurable geometric primitives [ESP 92]. This is done simply by defining the equations that represent:

– the primitive's nature and configuration in the scene:

$$\mathbf{h}(X, Y, Z, P_1, ..., P_n) = 0 \qquad [6.21]$$

– its projection onto the image plane:

$$\mathbf{g}(x, y, p_1, ..., p_m) = 0 \qquad [6.22]$$

– the relation between the 3-D primitive and its image (referred to as the limbo surface in the case of a volumetric primitive, see Figure 6.4):

$$1/Z = \mu(x, y, P_1, ..., P_l) = 0 \qquad [6.23]$$

As an example, if a line in space is represented by the intersection of the two following planes:

$$\mathbf{h}(X, Y, Z, A_1, ..., C_2) = \begin{cases} h_1 = A_1X + B_1Y + C_1Z + D_1 = 0 \\ h_2 = A_2X + B_2Y + C_2Z = 0 \end{cases} \qquad [6.24]$$

we immediately obtain, using the equations of perspective projection [6.12]:

– the function μ from h_1:

$$1/Z = Ax + By + C \tag{6.25}$$

with $A = -A_1/D_1$, $B = -B_1/D_1$ and $C = -C_1/D_1$;

– the equation of the 2-D line, denoted by $\mathcal{D}$, resulting from the projection onto the image of the 3-D line, from h_2:

$$ax + by + c = 0 \text{ with } a = A_2,\ b = B_2,\ c = C_2 \tag{6.26}$$

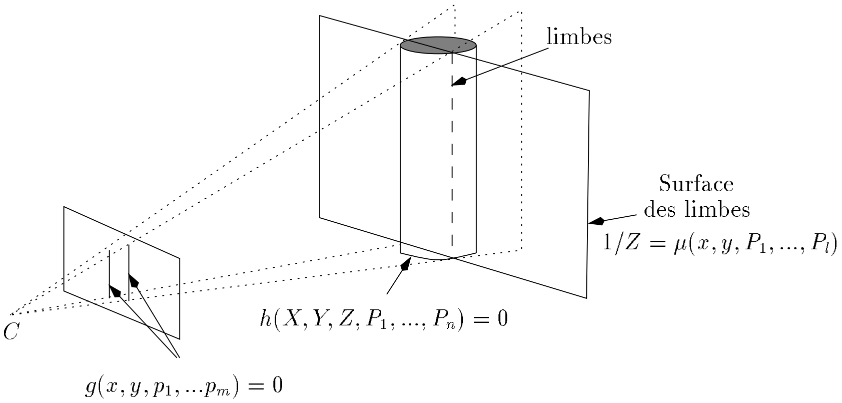

limbes
Surface
des limbes
$1/Z = \mu(x, y, P_1, ..., P_l)$
$h(X, Y, Z, P_1, ..., P_n) = 0$
C
$g(x, y, p_1, ...p_m) = 0$

Figure 6.4. *Projection of the primitive onto the image, and limbo surface in the case of cylinder*

Because the choice of parameters (a, b, c) is not minimal, it is preferable to choose the (ρ, θ) representation defined by:

$$g(x, y, \rho, \theta) = x\cos\theta + y\sin\theta - \rho = 0 \tag{6.27}$$

where $\theta = \arctan(b/a)$ and $\rho = -c/\sqrt{a^2 + b^2}$ (see Figure 6.5).

If we differentiate Equation [6.27], which corresponds to the hypothesis that the image of a line remains a line whatever the camera's motion, we get:

$$\dot{\rho} + (x\sin\theta - y\cos\theta)\,\dot{\theta} = \dot{x}\cos\theta + \dot{y}\sin\theta \ \ ,\ \ \forall (x, y) \in \mathcal{D} \tag{6.28}$$

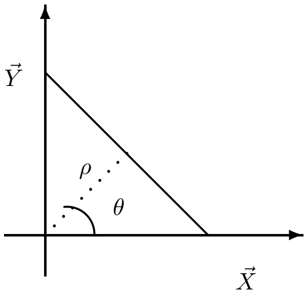


Figure 6.5. *(ρ, θ) representation of the 2-D lines*

Based on Equation [6.27], x is written according to y if $\cos\theta \neq 0$ (or y according to x if that is not the case) and Equation [6.28] can then be written, using [6.15] and [6.25]:

$$(\dot{\rho} + \rho \tan\theta \, \dot{\theta}) + y \, (-\dot{\theta}/\cos\theta) = \mathbf{K_1} \, \mathbf{v} + y \, \mathbf{K_2} \, \mathbf{v} \;\; , \;\; \forall y \in \mathbb{R} \qquad [6.29]$$

with:

$$\begin{array}{l} \mathbf{K_1} = [\; \lambda_1 \cos\theta \quad \lambda_1 \sin\theta \quad -\lambda_1 \rho \quad \sin\theta \quad -\cos\theta - \rho^2/\cos\theta \quad -\rho\tan\theta \;] \\ \mathbf{K_2} = [\; \lambda_2 \cos\theta \quad \lambda_2 \sin\theta \quad -\lambda_2 \rho \quad \rho \quad \rho\tan\theta \quad 1/\cos\theta \;] \end{array}$$

where $\lambda_1 = -A\rho/\cos\theta - C$ and $\lambda_2 = A\tan\theta - B$.

Immediately, we infer that:

$$\left\{ \begin{array}{lcl} \dot{\rho} & = & (\mathbf{K_1} + \rho \sin\theta \, \mathbf{K_2}) \, \mathbf{v} \\ \dot{\theta} & = & -\cos\theta \, \mathbf{K_2} \, \mathbf{v} \end{array} \right. \qquad [6.30]$$

hence:

$$\begin{array}{l} \mathbf{L}_\rho = [\; \lambda_\rho \cos\theta \quad \lambda_\rho \sin\theta \quad -\lambda_\rho \rho \quad (1+\rho^2)\sin\theta \quad -(1+\rho^2)\cos\theta \quad 0 \;] \\ \mathbf{L}_\theta = [\; \lambda_\theta \cos\theta \quad \lambda_\theta \sin\theta \quad -\lambda_\theta \rho \quad -\rho\cos\theta \quad -\rho\sin\theta \quad -1 \;] \end{array} \qquad [6.31]$$

with $\lambda_\rho = -A\rho\cos\theta - B\rho\sin\theta - C$ and $\lambda_\theta = -A\sin\theta + B\cos\theta$.

The same result can be obtained by applying Equation [6.28] to two points of $\mathcal{D}$, for example those with coordinates $(\rho\cos\theta, \rho\sin\theta)$ and $(\rho\cos\theta + \sin\theta, \rho\sin\theta - \cos\theta)$.

Results for more complex primitives (circles, spheres, and cylinders) are given in [CHA 93a], making it possible to use 2-D visual data associated with these primitives in visual servoing. It is also possible to infer the interaction matrix associated with data defined from several primitives (such as the orientation of a line segment or the distance from a point to a line, for example). The drawback, however, is that it is only possible to work on environments where such geometric primitives exist (hence the more frequent use of characteristic points).

6.2.4.3. *Interaction matrix for complex 2-D shapes*

Recent studies have made it possible to establish the analytical form of the interaction matrix associated with visual data representing the projection onto the image of objects with more complex shapes. In [COLO 99, DRU 99], the six terms that correspond to the affine part of the transformation between the image of a plane object in its current position and the image of the same object in the desired position are considered. More precisely, if (x, y) and (x^*, y^*) are the coordinates of a given point on the object in the current image and the desired image, respectively, then we assume that there exists a set of parameters $\boldsymbol{\theta} = (a_1, b_1, c_1, a_2, b_2, c_2)$ such that the relation:

$$\left\{ \begin{array}{lcl} x & = & a_1\, x^* + b_1\, y^* + c_1 \\ y & = & a_2\, x^* + b_2\, y^* + c_2 \end{array} \right. \quad [6.32]$$

is valid for all points of the object. This hypothesis is unfortunately not verified for a camera described by a perspective projection model. Additionally, the interaction matrix associated with $\boldsymbol{\theta}$ shows a loss in rank (from 6 to 4) when the object's plane is parallel to the image plane.

Furthermore, if we calculate the Fourier series expansion for the polar signature $\rho(\theta)$ of the contour points of an object in the image (defined such that the coordinates x and y of a contour point are written: $x = x_g + \rho(\theta)\cos\theta\ ,\ y = y_g + \rho(\theta)\sin\theta$ where x_g and y_g are the coordinates of the object's center of gravity), it is possible to calculate the interaction matrix associated with the terms of that series [COL 00]. The resulting analytical form, however, is not very inviting.

Another possibility is to calculate the interaction matrix associated with the moments m_{ij} of an object [CHA 04]. Moments are defined by:

$$m_{ij} = \int\int_{\mathcal{D}} x^i y^j \, dx\, dy \quad [6.33]$$

where $\mathcal{D}$ is the area occupied by the object in the image and where $i + j$ is the order of the moment. If we assume that the object considered is plane or has a plane limbo surface with equation $1/Z = Ax + By + C$, we obtain, for the area a $(= m_{00})$ and the coordinates x_g $(= m_{10}/m_{00})$ and y_g $(= m_{01}/m_{00})$ of the object's center of gravity:

$$\begin{array}{lcccccccc}
\mathbf{L}_a = [& -aA & -aB & a(3/Z_g - C) & 3ay_g & -3ax_g & 0 & [\\
\mathbf{L}_{x_g} = [& -1/Z_g & 0 & x_g/Z_g + \epsilon_1 & x_g y_g + 4n_{11} & -(1 + x_g^2 + 4n_{20}) & y_g &] \\
\mathbf{L}_{y_g} = [& 0 & -1/Z_g & y_g/Z_g + \epsilon_2 & 1 + y_g^2 + 4n_{02} & -x_g y_g - 4n_{11} & -x_g &]
\end{array} \quad [6.34]$$

with $1/Z_g = Ax_g + By_g + C$, $\epsilon_1 = 4(An_{20} + Bn_{11})$, $\epsilon_2 = 4(An_{11} + Bn_{02})$ and where n_{20}, n_{02} and n_{11} are the second order normalized centered moments defined by:

$$n_{ij} = \mu_{ij}/a \text{ with } \begin{cases} \mu_{20} = m_{20} - ax_g^2 \\ \mu_{02} = m_{02} - ay_g^2 \\ \mu_{11} = m_{11} - ax_g y_g \end{cases} \quad [6.35]$$

Note that the speed $\dot{a}$ is equal to zero for any movement other than the expected translation movements along the camera's optical axis if the object is centered and parallel to the image plane ($A = B = x_g = y_g = 0$). This makes area particularly interesting for controlling this degree of freedom, because of its relative decoupling compared to the other degrees of freedom.

Notice also that the results obtained for the coordinates of the object's center of gravity encompass those given in [6.15] for a purely punctual object, since for a point, we have $n_{20} = n_{11} = n_{02} = 0$ and we can assume that $A = B = 0$ in [6.34] to again obtain exactly [6.15].

More generally, the interaction matrix associated with a moment m_{ij} is given by:

$$\mathbf{L}_{m_{ij}} = \begin{bmatrix} m_{vx} & m_{vy} & m_{vz} & m_{wx} & m_{wy} & m_{wz} \end{bmatrix} \quad [6.36]$$

where:

$$\begin{cases} m_{vx} = -i(Am_{ij} + Bm_{i-1,j+1} + Cm_{i-1,j}) - Am_{ij} \\ m_{vy} = -j(Am_{i+1,j-1} + Bm_{ij} + Cm_{i,j-1}) - Bm_{ij} \\ m_{vz} = (i + j + 3)(Am_{i+1,j} + Bm_{i,j+1} + Cm_{ij}) - Cm_{ij} \\ m_{wx} = (i + j + 3)m_{i,j+1} + jm_{i,j-1} \\ m_{wy} = -(i + j + 3)m_{i+1,j} - im_{i-1,j} \\ m_{wz} = im_{i-1,j+1} - jm_{i+1,j-1} \end{cases}$$

For centered moments defined by:

$$\mu_{ij} = \iint_{\mathcal{D}} (x - x_g)^i (y - y_g)^j \, dx \, dy \qquad [6.37]$$

we get:

$$\mathbf{L}_{\mu_{ij}} = \left[\begin{array}{cccccc} \mu_{vx} & \mu_{vy} & \mu_{vz} & \mu_{wx} & \mu_{wy} & \mu_{wz} \end{array} \right] \qquad [6.38]$$

with:

$$\left\{ \begin{array}{ll} \mu_{vx} & = -(i+1)A\mu_{ij} - iB\mu_{i-1,j+1} \\ \mu_{vy} & = -jA\mu_{i+1,j-1} - (j+1)B\mu_{ij} \\ \mu_{vz} & = -A\mu_{wy} + B\mu_{wx} + (i+j+2)C\mu_{ij} \\ \mu_{wx} & = (i+j+3)\mu_{i,j+1} + ix_g\mu_{i-1,j+1} \\ & \qquad +(i+2j+3)y_g\mu_{ij} - 4in_{11}\mu_{i-1,j} - 4jn_{02}\mu_{i,j-1} \\ \mu_{wy} & = -(i+j+3)\mu_{i+1,j} - (2i+j+3)x_g\mu_{ij} \\ & \qquad -jy_g\mu_{i+1,j-1} + 4in_{20}\mu_{i-1,j} + 4jn_{11}\mu_{i,j-1} \\ \mu_{wz} & = i\mu_{i-1,j+1} - j\mu_{i+1,j-1} \end{array} \right.$$

The numerical value of the interaction matrix associated with a moment of order $i + j$ can thus be calculated from the measurement of moments with orders at most $i + j + 1$, which is convenient. The values A, B, C characterizing the plane's configuration must also be available (or at least an approximation of these values) in order to calculate the translation terms. As we have already said, this property is true for any generic visual datum defined in the image.

Based on the moments, it is possible to determine relevant geometric information, such as, as we have seen before, the area and the center of gravity of an object. The main orientation is obtained from the second order centered moments:

$$\alpha = \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \qquad [6.39]$$

and we easily get, using [6.38]:

$$\mathbf{L}_{\alpha} = \left[\begin{array}{cccccc} \alpha_{vx} & \alpha_{vy} & \alpha_{vz} & \alpha_{wx} & \alpha_{wy} & -1 \end{array} \right] \qquad [6.40]$$

where:

$$\left\{\begin{array}{lcl} \alpha_{vx} & = & aA + bB \\ \alpha_{vy} & = & -cA - aB \\ \alpha_{vz} & = & -A\alpha_{wy} + B\alpha_{wx} \\ \alpha_{wx} & = & -bx_g + ay_g + d \\ \alpha_{wy} & = & ax_g - cy_g + e \end{array}\right.$$

and:

$$\left\{\begin{array}{lcl} a & = & \mu_{11}(\mu_{20} + \mu_{02})/\Delta \\ b & = & [2\mu_{11}^2 + \mu_{02}(\mu_{02} - \mu_{20})]/\Delta \\ c & = & [2\mu_{11}^2 + \mu_{20}(\mu_{20} - \mu_{02})]/\Delta \\ d & = & 5[\mu_{12}(\mu_{20} - \mu_{02}) + \mu_{11}(\mu_{03} - \mu_{21})]/\Delta \\ e & = & 5[\mu_{21}(\mu_{02} - \mu_{20}) + \mu_{11}(\mu_{30} - \mu_{12})]/\Delta \\ \Delta & = & (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \end{array}\right.$$

We should point out that translation movements leave α unchanged when the object's plane is parallel to the image plane ($\alpha_{vx} = \alpha_{vy} = \alpha_{vz} = 0$ if $A = B = 0$). Note also the direct relation between the variation of α and the rotation movement about the optical axis ω_z, an indication, as we could have expected, that α is a good visual datum for controlling this degree of freedom.

One of the different possible strategies in visual servoing consists of directly using all of the measurements available in the image. We then have redundant visual data (that is, in higher numbers than the number of degrees of freedom we wish to control), and as we will see in section 6.3.2.2, servoing stability can only be demonstrated in the vicinity of the convergence position. Another, more promising strategy consists of determining complementary visual data, by construction or selection [COR 01, IWA 05, TAH 03], or even by finding a different way of expressing the perspective projection model (for example a spherical projection [HAM 02]). The case of an object's surface and orientation discussed earlier are simple and natural examples of such a determination. However, much remains to be done in this field.

6.2.4.4. *Interaction matrix by learning or estimation*

The use of the polar signature or of the moments allows us to consider objects with truly complex shapes, but requires a space segmentation phase in the image processing part that can turn out to be extremely difficult in textured environments. To avoid this segmentation phase and be able to process any kind of image, it is possible to conduct a principal component analysis of the desired image and select the principal eigenvectors [NAY 96, DEG 97]. The coefficients of this decomposition form the set s of the visual data. The analytical form of the associated interaction matrix is then unknown (since it is so difficult to obtain) and the servoing is based on a purely digital estimate provided by a learning technique. This technique consists of generating movements for the different degrees of freedom available and to measure the corresponding variation observed in the image.

Techniques to estimate the interaction matrix have also been used on geometric visual data such as those described in the previous sections. They are all based on the same idea and are performed either offline, by learning [WEI 84, RUF 99, LAP 04], possibly by using a neural network [SUH 93, WEL 96], or online during the servoing [KIN 94, HOS 94, CHA 96, JAG 97, PIE 04]. These studies fall into two categories, those based on purely digital estimates of the terms of the interaction matrix [WEI 84, SUH 93, WEL 96, HOS 94, JAG 97, PIE 04] or of its pseudoinverse directly [LAP 04], and those that estimate unkown parameters involved with this matrix, such as for example the structure of objects or the camera's intrinsic parameters [KIN 94, CHA 96, RUF 99]. The first case is very attractive in practice since it allows us to avoid any modeling phase. The resulting drawback is that it is impossible to demonstrate the system's stability in the presence of inevitable estimation errors. The second option is therefore more satisfactory theoretically speaking, but since it require an analytical estimation of the interaction matrix beforehand, it cannot be applied for now to servoing diagrams based on visual data as complex as those resulting from a principal component analysis of the image.

6.2.5. *3-D data interaction matrix*

As has been mentioned before, it is also possible to choose visual data no longer expressed directly from the image, but resulting from a reconstruction phase or a 3-D localization phase [WIL 96, MART 97]. These 3-D data are obtained either by a simple triangulation if a calibrated stereoscopic vision system is available, or, in the case of a monocular sensor, by dynamic vision or with a pose calculation method. Dynamic vision techniques rely on the measurement of the camera's movement and of the resulting movement in the image. They are usually rather sensitive to measurement errors [SMI 94, CHA 96]. We will now briefly describe pose calculation techniques, because they are the most commonly used in 3-D visual servoing.

6.2.5.1. *Pose calculation*

There are many methods for estimating a camera's pose with respect to an object using an image of this object. They rely on prior knowledge of a 3-D model for the object and of the camera's calibration parameters. More precisely, for an image acquired at instant t, they provide an estimate $\hat{\mathbf{p}}(t)$ of the real pose $\mathbf{p}(t)$ between the camera's frame and the object's frame based on the measurements $\mathbf{x}(t)$ extracted from the image, the camera's intrinsic parameters and the object's 3-D model, represented for example by the set $\mathbf{X}$ of the 3-D coordinates of the points that comprise it:

$$\hat{\mathbf{p}}(t) = \hat{\mathbf{p}}(\mathbf{x}(t), x_c, y_c, f_x, f_y, \mathbf{X}) \quad [6.41]$$

Most of the time, the geometric primitives behind the measurements $\mathbf{x}(t)$ are points [HOR 89, HAR 89, DEM 95], line segments [LOW 87, DHO 89], even conics [SAF 92, MA 93], or also cylindrical objects [DHO 90]. But very few methods combine different kinds of primitives (see however [PHO 95] for the combined use of points and lines).

The methods described in the literature are either purely geometric [HOR 89, DHO 89], digital and iterative linear [DEM 95] or purely non-linear [LOW 87]. Except for very peculiar cases [HOR 89], no analytical solution to the inverse problem is available.

We should point out that in the case of an error in the calibration parameters or in the object's model, the estimate $\hat{\mathbf{p}}(t)$ will be biased and, because of the absence of an analytical solution, it is unfortunately impossible to determine the value of this bias. The same goes for finding the interaction matrix associated with any datum built from $\hat{\mathbf{p}}(t)$. This is because, based on [6.41]:

$$\dot{\hat{\mathbf{p}}}(t) = \frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}}\,\dot{\mathbf{x}} = \frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}}\,\mathbf{L}_{\mathbf{x}}\,\mathbf{v} \quad [6.42]$$

hence:

$$\mathbf{L}_{\hat{\mathbf{p}}} = \frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}}\,\mathbf{L}_{\mathbf{x}} \quad [6.43]$$

The second term of this two matrix product, in other words the interaction matrix associated with $\mathbf{x}$, is therefore known if $\mathbf{x}$ is comprised of geometric primitives such as points or line segments. On the other hand, the first term, $\frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}}$, which represents the variation of the estimate of $\hat{\mathbf{p}}$ according to a variation of the measurements $\mathbf{x}$ in the image, is unknown. We can only note that it is directly related to the estimation method and depends once again on the camera's intrinsic parameters and the object's 3-D model. This is why we will assume from now on that the estimate of $\hat{\mathbf{p}}(t)$ is perfect, which is the case under the (strong) hypotheses that the camera is perfectly calibrated, that the 3-D model of the object is perfectly well known, that the measurements $\mathbf{x}(t)$

are not tainted with any errors, and that the estimation method is free of any digital instability.

The strongest hypothesis involves the estimation's assumed stability in regards to measurement errors, because if we consider for example four coplanar points, theoretically there exists only one solution to the localization problem [HOR 89], but a very small variation of the positions of the four points in the image can cause a very large variation in the estimate of $\hat{\mathbf{p}}$ (hence the matrix $(\frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}})$ is very poorly conditioned). Such an effect is illustrated by Figure 6.6. In practice, this effect decreases when considering a large number of points, or non-coplanar points, but there are currently no theoretical results available on the sensitivity of the estimation methods and of the measurements to choose, regarding what kind to use, but also how they are arranged in the image and the 3-D space.

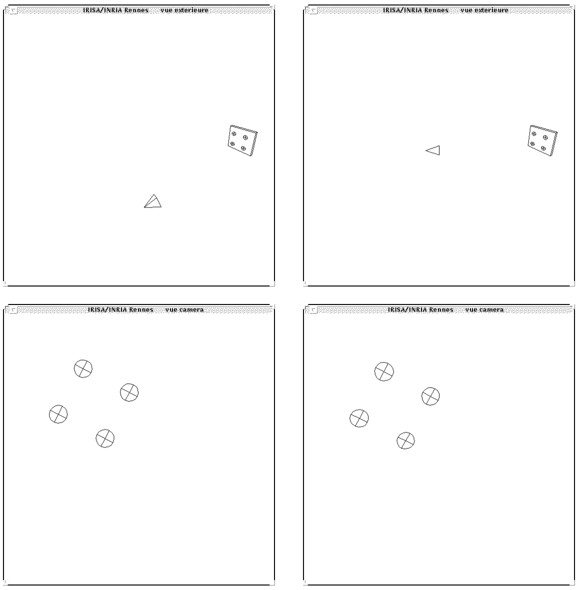

IRISA/INRIA Rennes vue exterieure
IRISA/INRIA Rennes vue exterieure
IRISA/INRIA Rennes vue camera
IRISA/INRIA Rennes vue camera

Figure 6.6. *Example of two distinct positions of the camera with respect to the object (top) that provide similar images of this object (bottom)*

Based on $\hat{\mathbf{p}}(t)$, and under the hypotheses mentioned previously, such as assuming a perfect estimate for $\hat{\mathbf{p}}(t)$ ($\hat{\mathbf{p}}(t) = \mathbf{p}(t)$), we have at our disposal the rotation ${}^{c}\mathbf{R}_{o}$ between the camera's frame in its current position R_c and the frame R_o attached to the object, as well as the translation ${}^{c}\mathbf{t}_{o}$ between theses two frames. We can then infer the position in R_c of any point attached to the object. If, additionally, in the context of a mounted camera, the pose between the camera's frame at its desired position R_{c^*} and the object's frame is known, then we can also infer the movement necessary to go from R_c and R_{c^*}. With a fixed scene camera, the same is true of course of an object attached to the robot's end-effector between its current position and its desired position.

We will now give the interaction matrix associated with the minimal representation $\theta\mathbf{u}$ of an arbitrary rotation with angle θ about an axis $\mathbf{u}$, then the one associated with the coordinates of a 3-D point.

6.2.5.2. *Interaction matrix associated with* $\theta\mathbf{u}$

Remember, first of all, that the $\theta\mathbf{u}$ representation is obtained in a unique manner from the coefficients $r_{ij(i=1..3,j=1..3)}$ of a rotation matrix $\mathbf{R}$ using the following equation [KHAL 02]:

$$\theta\mathbf{u} = \frac{1}{2\ \text{sinc}\theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \qquad [6.44]$$

where $\theta = \arccos((r_{11}+r_{22}+r_{33}-1)/2))$ and where the sine cardinal sincθ, defined by $\sin\theta = \theta\,\text{sinc}\theta$, is a function C^∞ equal to zero in $(2n+1)\pi$, $\forall n \in \mathbb{Z}$. For $\theta = \pi$, the only case not taken into account by [6.44], $\mathbf{u}$ is the eigenvector of $\mathbf{R}$ associated with the eigenvalue 1.

In the case of a mounted camera, it is possible to use for visual servoing the vector $\theta\mathbf{u}$ to represent the rotation ${}^{c^*}\mathbf{R}_c$ between R_{c^*} and R_c. If the matrices ${}^{c^*}\mathbf{R}_{n^*}$ and ${}^{c}\mathbf{R}_n$ are identical, which is usually the case, we can also consider the vector $\theta\mathbf{u}$ associated with the rotation ${}^{n^*}\mathbf{R}_n$. Likewise, with a scene camera, the vector $\theta\mathbf{u}$ can be used to represent either the rotation ${}^{o^*}\mathbf{R}_o$ between the desired frame and the current frame of the object connected to the effector, either the rotation ${}^{n^*}\mathbf{R}_n$ if the matrices ${}^{o^*}\mathbf{R}_{n^*}$ and ${}^{o}\mathbf{R}_n$ are identical (which is also usually the case).

In all of the cases mentioned above, the interaction matrix associated with $\theta\mathbf{u}$ is given by [MAL 99]:

$$\mathbf{L}_{\theta\mathbf{u}} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{L}_\omega \end{bmatrix} \qquad [6.45]$$

with:

$$\mathbf{L}_\omega = \mathbb{I}_3 - \frac{\theta}{2}\,[\mathbf{u}]_\times + (1 - \frac{\text{sinc}\theta}{\text{sinc}^2\frac{\theta}{2}})\,[\mathbf{u}]^2_\times \qquad [6.46]$$

The $\theta\mathbf{u}$ representation is therefore particularly interesting since $\mathbf{L}_\omega$ is singular only for $\theta = 2\pi$. Furthermore, we have:

$$\mathbf{L}_\omega^{-1} = \mathbb{I}_3 + \frac{\theta}{2}\,\mathrm{sinc}^2\frac{\theta}{2}\,[\mathbf{u}]_\times + (1 - \mathrm{sinc}\theta)[\mathbf{u}]_\times^2 \qquad [6.47]$$

which guarantees the following, rather convenient property:

$$\mathbf{L}_\omega^{-1}\,\theta\mathbf{u} = \theta\mathbf{u} \qquad [6.48]$$

If you would rather be considering the rotations ${}^{c}\mathbf{R}_{c^*}$, ${}^{n}\mathbf{R}_{n^*}$ or ${}^{o}\mathbf{R}_{o^*}$, we immediately infer from [6.45] that:

$$\mathbf{L}_{\theta\mathbf{u}} = \begin{bmatrix} \mathbf{0}_3 & -\mathbf{L}_\omega \end{bmatrix} \qquad [6.49]$$

and we now have:

$$\mathbf{L}_\omega^{-1}\,\theta\mathbf{u} = -\theta\mathbf{u} \qquad [6.50]$$

Note that it is not possible to directly take into account the vector $\theta\mathbf{u}$ associated with the rotation ${}^{c}\mathbf{R}_o$ and to base the argument on the difference between $\theta\mathbf{u}$ and $\theta^*\mathbf{u}^*$ (where $\theta^*\mathbf{u}^*$ represents the desired rotation ${}^{c^*}\mathbf{R}_o$). This is because $\theta\mathbf{u} - \theta^*\mathbf{u}^*$ does not represent a distance in the space SO_3 of rotations [SAM 91].

6.2.5.3. *Interaction matrix associated with a 3-D point*

Using the fundamental kinematics equation mentioned in [6.14], we immediately get for any point with coordinates $\mathbf{X}$ connected to the object:

$$\mathbf{L}_{\mathbf{X}} = \begin{bmatrix} -\mathbb{I}_3 & [\mathbf{X}]_\times \end{bmatrix} \qquad [6.51]$$

The points taken into account can be characteristic points of the object [MART 96, SCH 04], or also the origin of R_o (we then have $\mathbf{X} = {}^{c}\mathbf{t}_o$).

Thus, with a mounted camera, if we are interested in the movement it must achieve, we can also consider the origin of R_{c^*} (we then have $\mathbf{X} = {}^c\mathbf{t}_{c^*}$ and $\mathbf{X}^* = \mathbf{0}$) [MART 97]. In that case, it is even better to consider the position of the origin of the camera's frame expressed in a frame attached to the object, such as R_o, R_{c^*}, or even $R_\emptyset$ if the object is fixed (see Figure 6.7) [WIL 96].

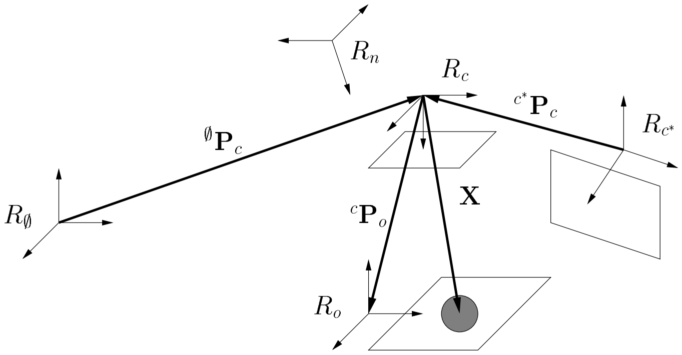


Figure 6.7. *Possible 3-D points with a mounted camera*

For example, if we choose R_o, we have:

$$^o\mathbf{t}_c = -{}^c\mathbf{R}_o^\top {}^c\mathbf{t}_o = -{}^o\mathbf{R}_c {}^c\mathbf{t}_o \qquad [6.52]$$

By differentiating this equation, we get:

$$\begin{aligned} {}^o\dot{\mathbf{t}}_c &= -{}^o\dot{\mathbf{R}}_c {}^c\mathbf{t}_o - {}^o\mathbf{R}_c {}^c\dot{\mathbf{t}}_o \\ &= -{}^o\mathbf{R}_c \left({}^c\mathbf{R}_o {}^o\dot{\mathbf{R}}_c {}^c\mathbf{t}_o + {}^c\dot{\mathbf{t}}_o\right) \end{aligned}$$

meaning that, using [6.4] and [6.51]:

$$\begin{aligned} {}^o\dot{\mathbf{t}}_c &= -{}^o\mathbf{R}_c \left([\boldsymbol{\omega}]_\times {}^c\mathbf{t}_o - \boldsymbol{v} + [{}^c\mathbf{t}_o]_\times \boldsymbol{\omega}\right) \\ &= {}^o\mathbf{R}_c \boldsymbol{v} \end{aligned}$$

We therefore have:

$$\mathbf{L}_{{}^o\mathbf{t}_c} = \left[\begin{array}{cc} {}^o\mathbf{R}_c & \mathbf{0}_3 \end{array}\right] \qquad [6.53]$$

which is independent of the camera's rotation movements. Likewise, if we choose ${}^{c^*}\mathbf{t}_c$, we get:

$$\mathbf{L}_{{}^{c^*}\mathbf{t}_c} = \left[\begin{array}{cc} {}^{c^*}\mathbf{R}_c & \mathbf{0}_3 \end{array}\right] \qquad [6.54]$$

and we will then have ${}^{c^*}\mathbf{t}_c{}^* = \mathbf{0}$.

With a scene camera (see Figure 6.8), and for the same decoupling properties, it is better to consider the position of the origin of either the frame R_o or R_n, and to express the kinematic torsor in this origin, because if we choose for example ${}^c\mathbf{t}_o$, then, using [6.51] and [6.6], we have:

$$\mathbf{L}_{{}^c\mathbf{t}_o}{}^c\mathbf{V}_o = \left[\begin{array}{cc} -\mathbb{I}_3 & [{}^c\mathbf{t}_o]_\times \end{array}\right] \left[\begin{array}{cc} {}^c\mathbf{R}_o & [{}^c\mathbf{t}_o]_\times {}^c\mathbf{R}_o \\ \mathbf{0}_3 & {}^c\mathbf{R}_o \end{array}\right] \qquad [6.55]$$

hence:

$$\mathbf{L}_{{}^c\mathbf{t}_o}{}^c\mathbf{V}_o = \left[\begin{array}{cc} -{}^c\mathbf{R}_o & \mathbf{0}_3 \end{array}\right] \qquad [6.56]$$

We can of course express the position of the origin of R_o in any frame of reference. If the robot's basic frame $R_\emptyset$ is chosen, we simply obtain:

$${}^\emptyset\dot{\mathbf{t}}_o = \left[\begin{array}{cc} \mathbb{I}_3 & \mathbf{0}_3 \end{array}\right] {}^\emptyset\mathbf{v}_o \qquad [6.57]$$

where ${}^\emptyset\mathbf{v}_o$ is the object's kinematic torsor expressed in $R_\emptyset$ and in the origin of R_o. The same result is of course achieved when considering ${}^\emptyset\mathbf{t}_n$ and ${}^\emptyset\mathbf{v}_n$.

6.2.5.4. *Interaction matrix associated with a 3-D plane*

Finally, we can also determine the interaction matrix associated with 3-D geometric primitives such as line-segments, planes, spheres, etc. For example, in the case of a plane represented by its unit normal $\mathbf{u}$ and its distance to the origin D, we get:

$$\mathbf{L}_{(\mathbf{u},D)} = \left[\begin{array}{cc} \mathbf{0}_3 & [\mathbf{u}]_\times \\ \mathbf{u}^\top & \mathbf{0} \end{array}\right] \qquad [6.58]$$

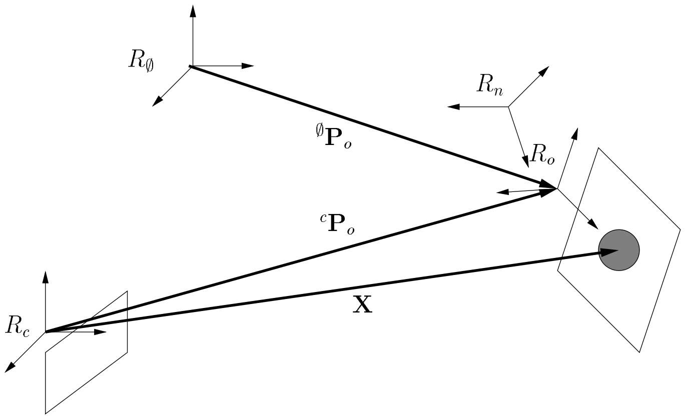


Figure 6.8. *Possible 3-D points with a scene camera*

6.3. Task function and command

Achieving a robotic task by visual servoing requires the selection of the appropriate visual data and the development of a closed-loop control law for these data. The first phase amounts to defining a task function with properties that ensure that the chosen task will be achieved [SAM 91], the second to regulating this task function. We will first be considering the case where we wish to control the system's 6 degrees of freedom, in other words to bring the frame of reference attached to the robot's effector to a unique desired pose.

If we use a set of k visual data $\mathbf{s}$, the general form of the task function $\mathbf{e}$ is:

$$\mathbf{e}(\mathbf{p}(t)) = \mathbf{C}\ (\mathbf{s}(\mathbf{p}(t)) - \mathbf{s}^*) \qquad [6.59]$$

where:

– $\mathbf{s}(\mathbf{p}(t))$ is the current value of the selected visual data;

– $\mathbf{s}^*$ is the value that $\mathbf{s}$ must reach for the task to be achieved;

– $\mathbf{C}$ is a full-rank $6 \times k$ matrix, referred to as the combination matrix, such that the 6 components of $\mathbf{e}$ are independent and control the system's 6 degrees of freedom.

6.3.1. *Obtaining the visual command* s^*

Whatever the nature of the visual data that were chosen, the command $\mathbf{s}^*$ is usually obtained, either by defining beforehand the pose that must be achieved between the robot and the object in question, or by learning:

– In the first case, if $\mathbf{s}$ includes 2-D data, their desired value can easily be obtained if a model of the object is available, simply by applying the perspective projection equations to calculate the object's position in the image. Additionally, it is also possible to specify the pose that has to be achieved between the effector and the object in question (for a gripping task for example): the calculation of the visual data (2-D or 3-D) is then immediately obtained if the basis change matrix between the effector and the camera is known. However, in any case, any modeling error in the camera's calibration parameters, in the model of the object (and possibly in the effector-camera basis change matrix) will have as a result that when the value of $\mathbf{s}$ is equal to $\mathbf{s}^*$, the pose actually reached will be different from the one that was specified, because of the bias introduced by the modeling errors.

– Obtaining the command $\mathbf{s}^*$ by learning, though less convenient to achieve in practice, is therefore preferable to ensure that the task is well achieved. It consists in a prior phase of bringing the robot to a desired position with respect to an object, then acquiring the corresponding image, and calculating the value of $\mathbf{s}^*$ exactly in the same way as for the future calculations of $\mathbf{s}(t)$. In the presence of a modeling error, we find ourselves in the paradoxical situation of having biased commands and measurements, but a pose after convergence that is accurate aside from the measurement errors.

– A third, more elegant solution consists of managing to have the camera observe the effector and the object in question simultaneously. The calculation of $\mathbf{s}^*$ can then be achieved automatically [HOR 98]. This solution has rarely been implemented, because, although it seems natural for scene cameras, it poses significant problems regarding where the sensors are placed in the case of mounted systems.

We will now give in detail the different possible choices for the combination matrix $\mathbf{C}$ by a (simple) analysis of the system's stability.

6.3.2. *Regulating the task function*

As we saw in the begining of this section, developing a control law to regulate the task function is separate from defining this function. In the literature, many types of control laws have been suggested: non-linear control laws [HAS 93b, REY 98], type LQ or LQG optimal [PAP 93, HAS 96], based on a GPC controller [GAN 02, GIN 05], even robust H_∞ [KHA 98] or by return of a non-stationary continuous return state [TSAK 98] in the case of mobile robots with nonholonomic constraints. We will simply focus on achieving a decoupled exponential decrease of the task function, that is:

$$\dot{\mathbf{e}} = -\lambda\, \mathbf{e} \qquad [6.60]$$

Using [6.59] and [6.2], if the matrix $\mathbf{C}$ is chosen constant, the differential of $\mathbf{e}$ is given by:

$$\dot{\mathbf{e}} = \mathbf{C}\,\dot{\mathbf{s}} = \mathbf{C}\,\mathbf{L_s}\,\mathbf{v} \tag{6.61}$$

We saw in sections 6.2.2 and 6.2.3 how to get from the kinematic torsor $\mathbf{v}$ to the joint variables $\dot{\mathbf{q}}$. For simpler notations, we will assume from now on that the control quantity is simply the controllable part of $\mathbf{v}$, denoted by $\mathbf{v_q}$, that is to say $\mathbf{v_q} = \mathbf{v_c}$ in the case of a mounted camera and $\mathbf{v_q} = -\mathbf{v_o}$ in the case of a scene camera (hence we will not be considering the problems caused by singularities and the robot getting blocked. Furthermore, we will not be considering the case of a robot with less than six degrees of freedom. We will just point out that, in that case, we must of course work directly in the joint space using [6.7] or [6.11], and not proceed in two steps with $\mathbf{v_q}$ then $\dot{\mathbf{q}}$). We therefore write:

$$\dot{\mathbf{e}} = \mathbf{C}\,\mathbf{L_s}\,\mathbf{v_q} + \frac{\partial \mathbf{e}}{\partial t} \tag{6.62}$$

where $\frac{\partial \mathbf{e}}{\partial t}$ represents the variations of $\mathbf{e}$ caused either by the object's movement (if we are working with a mounted camera), or by the camera's movement (if we are working with a scene camera or a mobile camera). To control the system's 6 degrees of freedom, it is necessary to at least select $\mathbf{s}$ such that $\mathbf{L_s}$ has rank 6 and we obtain as the ideal control law:

$$\mathbf{v_q} = (\mathbf{C}\,\mathbf{L_s})^{-1}\left(-\lambda\mathbf{e} - \frac{\partial \mathbf{e}}{\partial t}\right) \tag{6.63}$$

In the case where the visual data are expressed in the image, we saw that the interaction matrix depends on the values of these visual data and on the depth between the camera and the object in question. In the case of 3-D visual data, only, some rather strong hypotheses make it possible to obtain the analytical form of this matrix. In any case, measurement and estimation errors are inevitable and the exact value of $\mathbf{L_s}$ is unknown. Only an approximation $\widehat{\mathbf{L_s}}$ can therefore be considered in the control law. Also, the term $\frac{\partial \mathbf{e}}{\partial t}$ is usually unknown. Hence the control law used in practice is:

$$\mathbf{v_q} = \left(\mathbf{C}\,\widehat{\mathbf{L_s}}\right)^{-1}\left(-\lambda\mathbf{e} - \frac{\widehat{\partial \mathbf{e}}}{\partial t}\right) \tag{6.64}$$

If we assume that this command is perfectly achieved, the use of [6.64] in [6.62] leads to:

$$\dot{\mathbf{e}} = -\lambda\, \mathbf{C}\, \mathbf{L_s} \left(\mathbf{C}\, \widehat{\mathbf{L_s}}\right)^{-1} \mathbf{e} - \mathbf{C}\, \mathbf{L_s} \left(\mathbf{C}\, \widehat{\mathbf{L_s}}\right)^{-1} \widehat{\frac{\partial \mathbf{e}}{\partial t}} + \frac{\partial \mathbf{e}}{\partial t} \qquad [6.65]$$

If we assume, to make things simpler, that $\frac{\partial \mathbf{e}}{\partial t} = \widehat{\frac{\partial \mathbf{e}}{\partial t}} = 0$, then we notice that the positivity condition:

$$\mathbf{C}\, \mathbf{L_s} \left(\mathbf{C}\, \widehat{\mathbf{L_s}}\right)^{-1} > 0 \qquad [6.66]$$

is sufficient to ensure the decrease of $||\mathbf{e}||$ and therefore the system's global asymptotic stability ($||\mathbf{e}||$ is then a Lyapunov function). Also, the resulting behavior will be the same as the one specified in [6.60] under the unique condition that $\widehat{\mathbf{L_s}} = \mathbf{L_s}$ and that $\widehat{\frac{\partial \mathbf{e}}{\partial t}} = \frac{\partial \mathbf{e}}{\partial t}$. We will see in section 6.3.4 how we can estimate $\frac{\partial \mathbf{e}}{\partial t}$, which then makes it possible to reduce drag errors. We will now focus on different possible choices of $\mathbf{C}$ and $\widehat{\mathbf{L_s}}$. Therefore we will assume from now on that $\frac{\partial \mathbf{e}}{\partial t} = \widehat{\frac{\partial \mathbf{e}}{\partial t}} = 0$ so as not to complicate the notations too much.

6.3.2.1. *Case where the dimension of* $\mathbf{s}$ *is* 6 *(*$k = 6$*)*

If the dimension of $\mathbf{s}$ is 6, it is much more convenient to choose $\mathbf{C} = \mathbb{I}_6$, because the behavior of $\mathbf{s}$ will then be the same as that of $\mathbf{e}$ (meaning that, in the ideal case, both components of $\mathbf{s}$ will show a decoupled exponential decrease). In that case, we get the control law:

$$\mathbf{v_q} = -\lambda\, \widehat{\mathbf{L_s}}^{-1} \mathbf{e} = -\lambda\, \widehat{\mathbf{L_s}}^{-1} (\mathbf{s} - \mathbf{s}^*) \qquad [6.67]$$

and the stability condition:

$$\mathbf{L_s}\, \widehat{\mathbf{L_s}}^{-1} > 0 \qquad [6.68]$$

If we are able to properly measure the current value of $\mathbf{L_s}$ at each iteration of the control law, taking this estimation into account makes it possible to come closest to the ideal behavior $\dot{\mathbf{s}} = -\lambda\, \mathbf{s}$.

6.3.2.1.1. 2-D visual data

When considering 2-D visual data, it is unfortunately extremely difficult (as of now) to end up in this type of situation. The main difficulty involves not the estimate of the current value of the interaction matrix, but the selection of the six visual data.

Consider for example the case where $\mathbf{s}$ is comprised of the projection coordinates of three points in the image. The associated interaction matrix $\mathbf{L_s}$ is then a 6×6 matrix and, most of the time, is a full rank matrix. But we can show [MIC 93, PAP 95] that some configurations lead to a loss of rank of $\mathbf{L_s}$. In this case, the singularities are such that the three points are aligned in the image or that the optical center C of the camera belongs to the surface of the cylinder defined by the circumcircle of these three points (see Figure 6.9). It is therefore difficult to ensure that, for any chosen initial position, the robot's movement will avoid going through an isolated singularity (where of course the stability condition [6.68] is no longer verified).

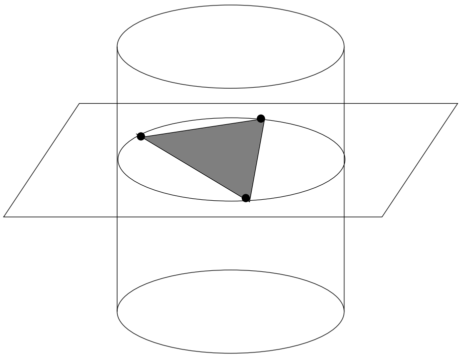


Figure 6.9. *Singularity cylinder*

Also, there are usually four distinct poses between the camera and the scene such that the image of three points is the same [DHO 89]. Minimizing $\|\mathbf{s}-\mathbf{s}^*\|$ can therefore bring the robot to one of the four global minima such that $\|\mathbf{s} - \mathbf{s}^*\| = 0$. Thus, in this case, it is quite possible to have $\mathbf{s} - \mathbf{s}^* = 0$ even if the pose that was reached is not the specified pose.

When considering visual data of different kinds (such as for example the three lines that can be defined from three non-aligned points), the same potential problems arise. A convenient solution consists of restricting the workspace to areas close to the desired pose, that include no isolated singularities, and where the minimizing of $\|\mathbf{s}-\mathbf{s}^*\|$ draws the robot's effector to its desired position. However, determining the size of these areas is a difficult problem.

Furthermore, if the six terms describing the affine deformation of an object (see 6.2.4.3), a loss of rank of the interaction matrix occurs when the considered object is parallel to the image plane [COLO 99, DRU 99], rendering servoing impossible in the vicinity of this configuration.

Very recent studies based on the search of moment combinations have allowed to determine sets of six visual data with very interesting properties (unaffected by certain movements, directly related to others) [TAH 03]. However, these results are not yet definitive and the absence of isolated singularities or local minima has not yet been demonstrated.

Because of the different reasons mentioned above, it is very common to use redundant 2-D visual data. We then have k greater than 6, a situation described in section 6.3.2.2.

6.3.2.1.2. 3-D visual data

The use of 3-D visual data makes it possible to avoid the problems mentioned earlier since three parameters $\theta\mathbf{u}$ are now available to represent the orientation and only three position parameters have to be chosen among those given in 6.2.5.3 to have $k = 6$. Remember, however, that it is still necessary to be in the ideal situation, for which the different measurement, calibration and estimation errors are negligible, to be able to express the interaction matrix. In the rest of this section, we will therefore assume that we are in this ideal case which (theoretically) ensures the specified behavior $\dot{\mathbf{s}} = -\lambda\mathbf{s}$ and the stability condition [6.68] in the entire workspace, since we then have:

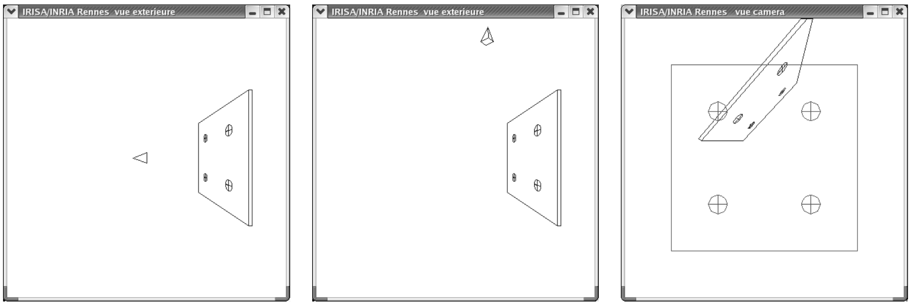
$$\mathbf{L_s}\,\widehat{\mathbf{L_s}}^{-1} = \mathbf{L_s}\,\mathbf{L_s}^{-1} = \mathbb{I}_6 > 0 \qquad [6.69]$$

If we choose to use $\theta\mathbf{u}$ to represent the rotation ${}^{c^*}\mathbf{R}_c$ and the coordinates $\mathbf{X}$ of a point attached to the object expressed in the camera's current frame of reference R_c, the global interaction matrix associated with $\mathbf{s} = (\mathbf{X}, \theta\mathbf{u})$ is given by:

$$\mathbf{L_s} = \begin{bmatrix} -\mathbb{I}_3 & [\mathbf{X}]_\times \\ \mathbf{0}_3 & \mathbf{L}_\omega \end{bmatrix} \qquad [6.70]$$

Notice how appealing this matrix is (block-triangular and not singular except in $\theta = 2\pi$), giving the system an interesting behavior, since, ideally, the trajectory of the considered point is a straight line in the image. If this point is in the camera's field of view in its initial position and in its desired position, then it will be so permanently. By properly selecting this point (in the object's center of gravity for example), we can then minimize the risk of losing a large part of the object in question during the servoing (without ensuring however that a sufficient number of characteristics necessary to the calculation remain visible). At each iteration we can also select the 3-D point that corresponds to the 2-D point closest to the limits of the image plane. But this choice implies a discontinuity in the control's translation components every time the point is changed. Additionally, without a higher level strategy (such as planning the trajectories in the image), it is always possible to display particular cases that will either cause a part of the object to fall out of view, or lead to perverse effects on the control (if two points are close to opposing edges of the image for example). Finally, the translation trajectory followed by the camera is a straight line, but only in the camera's *mobile* frame of reference. Hence it will not be an actual straight line if an orientation change is necessary.

To illustrate the behavior of this control law, we will consider a positioning task with respect to four points that form a square. As Figure 6.10 shows, the desired pose of the camera is such that it is parallel and centered with respect to the square, with the image of the four points forming a centered square with its sides parallel to the axes of the image planes. The initial pose chosen corresponds to a significant movement, particularly in rotation. The results obtained by simulation in ideal conditions (that is to say without causing measurement errors, calibration errors, or pose estimation errors) are shown in Figure 6.11. We considered, as the coordinates $\mathbf{X}$ in $\mathbf{s}$, those of the origin O in the object's frame of reference, located in the square's center. Notice how, as expected, the trajectory in the image of the projection of O, given as an illustration, forms a perfectly straight line. On the other hand, the camera's trajectory is no straight line at all. Finally, the components of the camera's kinematic torsor show a nice exponential decrease, due to the strong decoupling of this control law.


IRISA/INRIA Rennes vue exterieure
IRISA/INRIA Rennes vue exterieure
IRISA/INRIA Rennes vue camera

(a) (b) (c)

Figure 6.10. *Example of a positioning task: (a) desired pose for the camera, (b) initial pose, (c) image of the object for the initial pose and the desired pose*

With a mounted camera, we also get a block-triangular interaction matrix if we consider the vector $\mathbf{X} = {}^{c}\mathbf{t}_{c^*}$ to control the camera's position. But there is little point to this in practice. On the other hand, if we choose to consider the position of the origin of R_c expressed in a frame attached to the object, for example ${}^{c^*}\mathbf{t}_c$, we then have (see [6.54]):

$$\mathbf{L_s} = \begin{bmatrix} {}^{c^*}\mathbf{R}_c & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{L}_\omega \end{bmatrix} \qquad [6.71]$$

which is block-diagonal and therefore ensures a complete decoupling between the translation movements and the rotation movements. Additionally, the translation trajectory of the camera will be an actual straight line, something of significant practical interest. To have the robot's effector follow a straight line, we simply have to consider ${}^{n^*}\mathbf{t}_n$ in $\mathbf{s}$ instead of ${}^{c^*}\mathbf{t}_c$. Unfortunately, in both cases, there is no longer any control over the object's trajectory in the image, and if the camera's initial position is far from its desired position, there is no guarantee that the object will remain in the camera's field of view during the servoing.

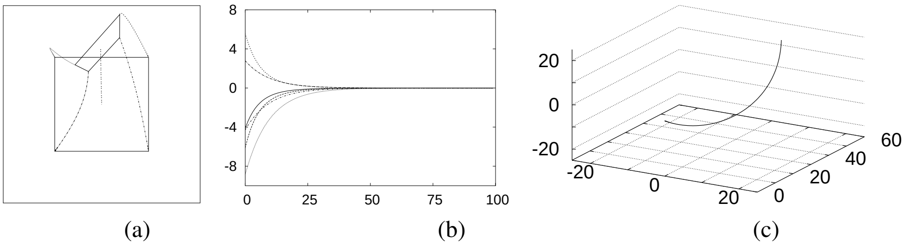


Figure 6.11. *Servoing results when choosing* $\mathbf{s} = ({}^{c}\mathbf{t}_o, \theta\mathbf{u})$*: (a) trajectories of points in the image, (b) components of* $\mathbf{v}_c$ *(in cm/s and deg/s) calculated at each iteration, (c) trajectory of the origin of the camera's frame in the frame* $\mathcal{R}_{c^*}$ *(in cm)*

The simulation results for this control law, obtained in the exact same conditions as before, are shown in Figure 6.12. They bring support to the comments stated above. Notice also that the decrease of the translation components of $\mathbf{v}_c$ are not as good, because of the coupling of these components induced by the very strong rotation that has to be performed.

Similar choices are of course also possible with a scene camera (see section 6.2.5.3). As an example, if we select in $\mathbf{s}$ the translation ${}^{\emptyset}\mathbf{t}_o$ and the vector $\theta\mathbf{u}$ associated with the rotation ${}^{o^*}\mathbf{R}_o$, we end up, by combining Equations [6.67], [6.57] and [6.48], with the following control law:

$$\begin{cases} {}^{\emptyset}\boldsymbol{v}_o = -\lambda\,({}^{\emptyset}\mathbf{t}_o - {}^{\emptyset}\mathbf{t}_{o^*}) \\ {}^{o}\boldsymbol{\omega}_o = -\lambda\,\theta\,\mathbf{u} \end{cases} \qquad [6.72]$$

where ${}^{\emptyset}\boldsymbol{v}_o$ is the translation speed of R_o expressed in $R_{\emptyset}$ and where ${}^{o}\boldsymbol{\omega}_o$ is its rotation speed expressed in R_o. This control law shows ideal decoupling properties. Additionally, it ensures a straight line trajectory for the origin of R_o both in the 3-D space and in the image. We should however point out that if there are modeling errors present in the robot's Jacobian or a calibration error in the basis change matrix from R_n to R_o, the trajectory actually performed in practice will be different from what is expected. But the closed loop that is used is robust when it comes to these calibration errors and it is possible to quantify this robustness by the analysis of the stability condition [6.68] by reasoning in the joint space.

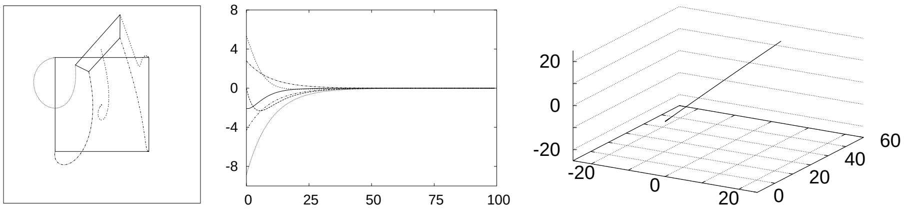


Figure 6.12. *Servoing results when choosing* $\mathbf{s} = ({}^{c^*}\mathbf{t}_c, \theta\mathbf{u})$

6.3.2.1.3. 2-D 1/2 visual data

As we have already said a number of times, taking 3-D visual data into account is based on the hypothesis that these data can be measured reliably. In practice, they are more sensitive to measurement errors than 2-D visual data, since they are obtained from these data and from a pose calculation without any particular smoothing properties. It is therefore a good idea to combine 2-D and 3-D visual data to gain robustness to measurement errors while maintaining good decoupling properties. In [MAL 99], the task function is defined as follows:

$$\mathbf{e} = \left(\ x - x^*\,,\quad y - y^*\,,\quad \log(Z/Z^*)\,,\quad \theta\mathbf{u}\ \right) \qquad [6.73]$$

where:

– (x, y) and (x^*, y^*) are the current coordinates and the desired coordinates, respectively, of a characteristic point in the image;

– Z/Z^* is the ratio of the current distance to the desired distance of this point;

– $\theta\mathbf{u}$ represents the rotation ${}^{c^*}\mathbf{R}_c$ that is to be achieved.

In this case, we get as our control law:

$$\mathbf{v_q} = -\lambda \begin{bmatrix} Z\mathbf{L}_{e_v}^{-1} & -Z\mathbf{L}_{e_v}^{-1}\mathbf{L}_{e_{v\omega}} \\ \mathbf{0}_3 & \mathbb{I}_3 \end{bmatrix} \mathbf{e} \qquad [6.74]$$

where (see [6.16] and [6.51]):

$$\mathbf{L}_{e_v} = \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \\ 0 & 0 & -1 \end{bmatrix} \quad \text{and} \quad \mathbf{L}_{e_{v\omega}} = \begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \\ -y & x & 0 \end{bmatrix}$$

Therefore, the resulting decoupling is satisfactory since the command matrix is triangular. Additionally, the trajectory of the characteristic point that was chosen will be a straight line in the image. By properly selecting this point (in the object's center of gravity for example, or as close as possible to the limits of the image, which results in the same drawbacks as those described at the begininng of the previous section), it is usually possible to keep the object inside the image. Furthermore, with the help of recent results in projective reconstruction, it is possible to use this control technique on objects whose 3-D models are unknown [MAL 00]. Because the pose calculation is no longer involved, it is then possible to determine the actual analytical form of the interaction matrix (meaning a form that does not rely on the strong hypotheses used before) and get it to display the camera's calibration errors. Thanks to the triangular form of the matrix, it is then possible to reveal the analytical conditions that ensure the system's local and global asymptotic stabilities [MAL 99, MAL 02].

In our example, the behavior resulting from the control law [6.74] and from choosing the object's center of gravity as the characteristic point is shown in Figure 6.13. Notice the straight line trajectory of this point in the image and the fact that this behavior is very similar to the one obtained when choosing $\mathbf{s} = ({}^c\mathbf{t}_o, \theta\mathbf{u})$ (go back to Figure 6.11).

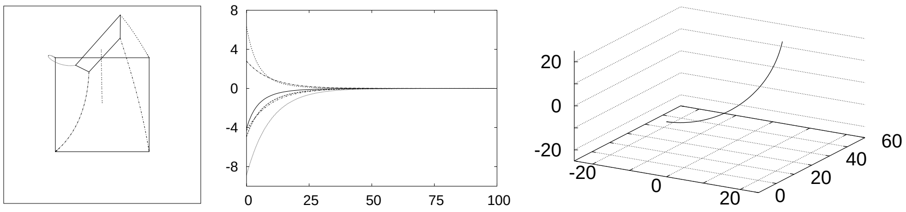


Figure 6.13. *Servoing results when choosing* $\mathbf{s} = (\mathbf{x}_g, \log(Z_g/Z_g^*), \theta\mathbf{u})$

Another version of this technique is described in [MOR 00]. The only difference involves the third component of $\mathbf{e}$ which explicitly takes into account the fact that all of the object's points must remain, as much as possible, inside the image. However, the triangular form of $\mathbf{L}_{e_v}$ is then lost, making it difficult to determine analytical stability conditions when calibration errors are present.

A second 2-D 1/2 visual servoing technique is described in [CHA 00] in the case of a mounted camera. The task function is given by:

$$\mathbf{e} = \left(\; {}^{c^*}\mathbf{t}_c, \quad x - x^*, \quad y - y^*, \quad \theta u_z \; \right) \qquad [6.75]$$

where (x, y) and (x^*, y^*) are again the current and the desired coordinates of a characteristic point in the image, and where u_z is the third component of the rotation axis $\mathbf{u}$ between R_{c^*} and R_c. Using [6.54], [6.16] and [6.45], we infer the expression of the associated control law:

$$\mathbf{v_c} = -\lambda \begin{bmatrix} {}^c\mathbf{R}_{c^*} & \mathbf{0}_3 \\ -\frac{1}{Z}\mathbf{L}_{e_\omega}^{-1}\mathbf{L}_{e_{\omega v}}\,{}^c\mathbf{R}_{c^*} & \mathbf{L}_{e_\omega}^{-1} \end{bmatrix} \mathbf{e} \qquad [6.76]$$

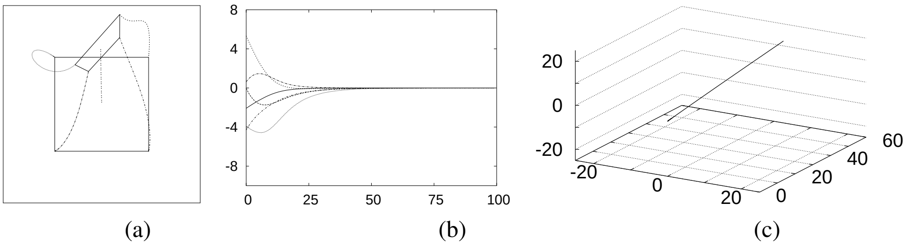
where:

$$\mathbf{L}_{e_{\omega v}} = \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{L}_{e_\omega} = \begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \\ l_1 & l_2 & l_3 \end{bmatrix}$$

$[l_1 \; l_2 \; l_3]$ being the third line of the matrix $\mathbf{L}_\omega$ given in [6.46].

Compared to the previous case, the camera translation will follow a straight line trajectory, its orientation controlled so that the trajectory of the characteristic point follows a straight line in the image (see Figure 6.14). This control law is therefore extremely useful in practice. If we select, as our characteristic point, the point of the object closest to the limits of the image, the discontinuity of the control law when changing the point will now involve the components of the rotation speed. On the other hand, note that the command matrix is no longer block-triangular, which again makes it difficult to determine analytical conditions ensuring the system's stability when calibration errors are present. The same is true if the task function θu_z is replaced by the orientation of a line segment, of a line, or of an object in the image (see sections 6.2.4.2 and 6.2.4.3). More visual data are then used and the only change that involves the modeling of the control law consists of replacing the coefficients on the last lines of $\mathbf{L}_{e_{\omega v}}$ and $\mathbf{L}_{e_\omega}$ with their new values.

Finally, in [AND 02], the Plücker coordinates of the lines are used, which also leads to a 2-D 1/2 visual servoing technique. However the resulting trajectories in space and in the image are not as satisfactory as in the two cases described previously.



(a) (b) (c)

Figure 6.14. *Servoing results when choosing* $\mathbf{s} = ({}^{c^*}\mathbf{t}_c, \mathbf{x}_g, \theta u_z)$

6.3.2.2. *Case where the dimension of* $\mathbf{s}$ *is greater than* 6 *(*$k > 6$*)*

We will now describe the different possible choices for $\mathbf{C}$ and $\widehat{\mathbf{L}_\mathbf{s}}$ (see [6.64]) when the visual data that are chosen are redundant ($k > 6$). Aside for the studies described in [MART 96, SCH 04] where the coordinates of several 3-D points are taken into account, this case only involves the choice of 2-D visual data since selecting six independent visual data is then a difficult problem.

Remember that $\mathbf{C}$ has to be a $6 \times k$ constant matrix with rank 6. The simplest choice consists of choosing as $\mathbf{C}$ the pseudoinverse of an approximate value of the interaction matrix in the desired position:

$$\mathbf{C} = \widehat{\mathbf{L}_{\mathbf{s}}}^{+}_{|\mathbf{s}=\mathbf{s}^*} \qquad [6.77]$$

As we saw in the first part of this chapter, the interaction matrix depends on the value of the visual data that are chosen and the depth between the camera and the scene's corresponding primitives. The calculation of $\mathbf{C}$ requires the value of $\mathbf{s}^*$ to be known, as well as the depth parameters in the desired position. If the 3-D model of the object is available, these parameters can easily be calculated by a pose calculation using the desired image. Otherwise, they are usually determined during the task specification itself.

Using [6.77], the control law [6.64] is expressed:

$$\mathbf{v_q} = -\lambda \left(\widehat{\mathbf{L}_{\mathbf{s}}}^{+}_{|\mathbf{s}=\mathbf{s}^*} \, \widehat{\mathbf{L}_\mathbf{s}} \right)^{-1} \mathbf{e} \qquad [6.78]$$

and, by choosing $\widehat{\mathbf{L}_{\mathbf{s}}}_{|\mathbf{s}=\mathbf{s}^*}$ to approximate $\widehat{\mathbf{L}_\mathbf{s}}$, in the end we get:

$$\mathbf{v_q} = -\lambda \, \mathbf{e} = -\lambda \, \widehat{\mathbf{L}_{\mathbf{s}}}^{+}_{|\mathbf{s}=\mathbf{s}^*} \, (\mathbf{s} - \mathbf{s}^*) \qquad [6.79]$$

We should point that, even if $\mathbf{e}$ is perfectly regulated (i.e. $\mathbf{e} = 0$), it does not necessarily imply that the visual task is achieved (i.e. $\mathbf{s} = \mathbf{s}^*$), because the set of configurations such that:

$$(\mathbf{s} - \mathbf{s}^*) \in \text{Ker}\, \mathbf{C} \qquad [6.80]$$

leads to $\mathbf{e}$ being equal to zero without $(\mathbf{s}-\mathbf{s}^*)$ being equal to zero. Hence it is important to make sure during the selection of the visual data not to create local potential minima in the workspace. As an example, consider a centered square parallel to the image plane. It is possible to show that by choosing, as the visual data, the coordinates in the image of the square's four corners, the configurations corresponding to local minima are such that the camera observes the square on its side. The four points are then aligned in the image and what we have is a degenerate case, outside the workspace of course.

Additionally, the stability condition is now written as:

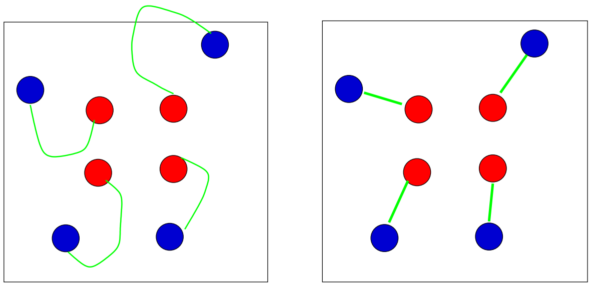
$$\widehat{\mathbf{L}_{\mathbf{s}}}^{+}_{|\mathbf{s}=\mathbf{s}^*}\, \mathbf{L}_{\mathbf{s}} > 0 \qquad [6.81]$$

Even with a perfect estimate of $\widehat{\mathbf{L}_{\mathbf{s}}}_{|\mathbf{s}=\mathbf{s}^*}$, this positivity condition is only ensured in a neighborhood around the desired position. Usually, only local asymptotic stability can be demonstrated. Likewise, the decoupled exponential behavior of $\mathbf{e}$ will only be ensured at this desired position. It is therefore possible, if the camera's initial position is far away from the desired position, that the resulting trajectories in the image turn out to be poorly satisfactory, or do not even lead to a convergence of the system (see Figure 6.15a). In practice, this only occurs if considerable rotation movements are required [CHA 98].

In our example, note the convergence of the servoing in Figure 6.16, despite the very significant rotation that has to be achieved. However, the components of $\mathbf{v}_c$ do not behave well, except near the convergence.

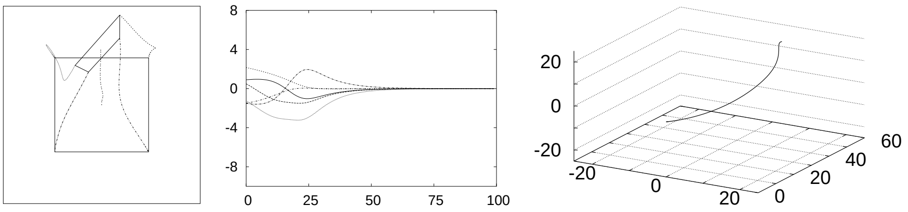
These problems are commonly solved by directly choosing for $\mathbf{C}$ the pseudo-inverse of an estimated current value of the interaction matrix, instead of a constant matrix:

$$\mathbf{C} = \widehat{\mathbf{L}_{\mathbf{s}}}^{+} \qquad [6.82]$$



(a) (b)

Figure 6.15. *(a) possible trajectory in the image when choosing* $\mathbf{C} = \widehat{\mathbf{L}_\mathbf{s}}^+_{|\mathbf{s}=\mathbf{s}^*}$*,*
(b) expected trajectory in the image when choosing $\mathbf{C} = \widehat{\mathbf{L}_\mathbf{s}}^+$



(a) (b) (c)

Figure 6.16. *Servoing results when choosing* $\mathbf{s} = (x_1, y_1, ..., x_4, y_4)$ *and* $\mathbf{C} = \mathbf{L}_\mathbf{s}^+{}_{|\mathbf{s}=\mathbf{s}^*}$

This leads us to:

$$\mathbf{v_q} = -\lambda \, \widehat{\mathbf{L}_\mathbf{s}}^+ \, (\mathbf{s} - \mathbf{s}^*) \qquad [6.83]$$

It is now necessary at each iteration of the control law to estimate either the 3-D parameters involved in the interaction matrix, or to perform an online digital estimation of the elements of this matrix (see section 6.2.4.4). In the absence of time smoothing in the calculation of this matrix, the system's behavior will therefore be less stable than in the previous case.

Also, the convergence condition [6.66] no longer applies since the calculation of $\dot{\mathbf{e}}$ would have to take into account the variations of $\mathbf{C}$ (see [6.61]), leading to virtually unfeasible calculations. Once again, only the asymptotic stability can be demonstrated. Considering the behavior of $\mathbf{s}$, we get:

$$\dot{\mathbf{s}} = -\lambda \mathbf{L}_\mathbf{s} \, \widehat{\mathbf{L}_\mathbf{s}}^+ \, (\mathbf{s} - \mathbf{s}^*) \qquad [6.84]$$

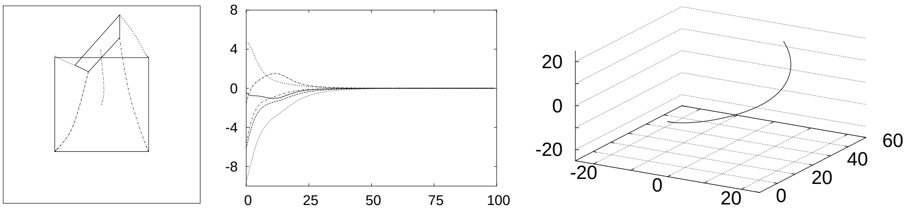
As in the previous case, it is therefore impossible to ensure the strict decrease of $\|\mathbf{s} - \mathbf{s}^*\|$ at each iteration since the $k \times k$ matrix $\mathbf{L}_\mathbf{s}\ \widehat{\mathbf{L}_\mathbf{s}}^+$ only has rank 6. All configurations such that:

$$(\mathbf{s} - \mathbf{s}^*) \in \text{Ker}\,\widehat{\mathbf{L}_\mathbf{s}}^+ \qquad [6.85]$$

correspond to attractive local minima, the existence of which is demonstrated [CHA 98] in the very simple case of a square.

The drawback of this method is that it attempts to ensure that $\dot{\mathbf{s}} = -\lambda(\mathbf{s} - \mathbf{s}^*)$ (directly providing the control law [6.83]), which implies k constraints when only 6 degrees of freedom are available. In other words, the "task function" $(\mathbf{s} - \mathbf{s}^*)$ is no longer ρ-admissible [SAM 91]. On the other hand, the advantage of this method is that, when it succeeds, it provides very interesting trajectories in the image, because if for example $\mathbf{s}$ is comprised of the coordinates of points in the image, the expected trajectories of these points will be straight lines (see Figure 6.15b). In practice, the actual trajectories will not necessarily be as inviting (since the actual behavior is given by [6.84]). Also, the robot's trajectory needed to achieve these movements in the image is not necessarily an advisable one [CHA 98].

These properties are summed-up in Figure 6.17: the trajectory of the points in the image is no longer a true straight line. Additionally, the camera's movement is not ideal given the components of $\mathbf{v}_c$ and the trajectory of the camera's optical center.



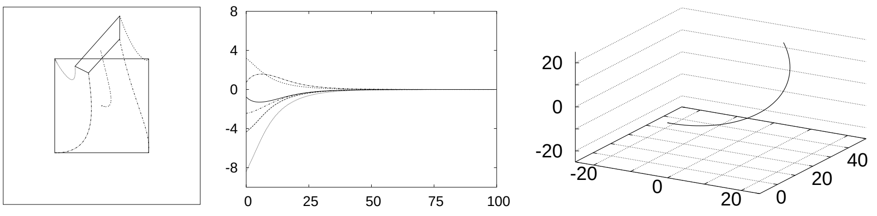
(a) (b) (c)

Figure 6.17. *Servoing results when choosing* $\mathbf{s} = (x_1, y_1, ..., x_4, y_4)$ *and* $\mathbf{C} = \mathbf{L}_\mathbf{s}^+$

Finally, other methods can be found in the literature. For example, $\widehat{\mathbf{L}_\mathbf{s}}^\top$ can be used in the control law (instead of $\widehat{\mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}}^+$ or $\widehat{\mathbf{L}_\mathbf{s}}^+$) [HAS 93b]. However, the advantages of this method compared to those described before are not obvious, since they do not show good decoupling properties. On the other hand, as was recently suggested [MAL 04], the choice of:

$$\mathbf{C} = \left(\frac{1}{2}\widehat{\mathbf{L_s}} + \frac{1}{2}\widehat{\mathbf{L_s}}_{|\mathbf{s}=\mathbf{s}^*} \right)^+ \qquad [6.86]$$

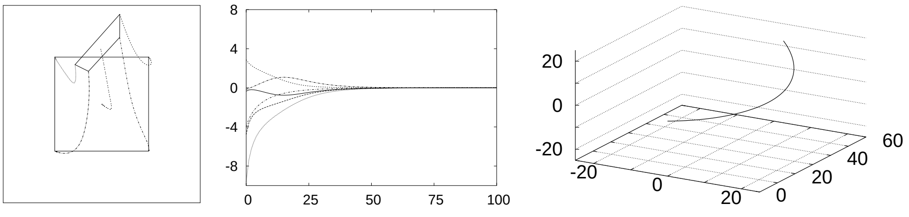
leads to satisfactory results in practice, as Figure 6.18 shows, even if the camera's trajectory is not a straight line.



(a) (b) (c)

Figure 6.18. *Servoing results when choosing* $\mathbf{s} = (x_1, y_1, ..., x_4, y_4)$ *and* $\mathbf{C} = (\frac{1}{2}\mathbf{L_s} + \frac{1}{2}\mathbf{L_s}_{|\mathbf{s}=\mathbf{s}^*})^+$

Again, we insist on the importance of the choice of the visual data inside the control law. As an illustration, Figure 6.19 shows the results obtained in our example when choosing in $\mathbf{s}$ the parameters (ρ_i, θ_i) associated with the four lines forming the sides of a square (see section 6.2.4.2). Note that these parameters can of course be directly calculated from the position of the four points in the image. No additional information is necessary. Also, the control law that was chosen is the one that uses a constant matrix for $\mathbf{C}$, namely $\mathbf{L_s^+}_{|\mathbf{s}=\mathbf{s}^*}$ which is easily obtained from [6.31]. As you can see on Figure 6.19, the behavior is quite different from Figure 6.16, even though it was also obtained using a constant matrix, and is just as satisfactory as the one obtained in Figure 6.18.



(a) (b) (c)

Figure 6.19. *Servoing results when choosing* $\mathbf{s} = (\rho_1, \theta_1, ..., \rho_4, \theta_4)$ *and* $\mathbf{C} = \mathbf{L_s^+}_{|\mathbf{s}=\mathbf{s}^*}$

To conclude this section, we should point out the importance of the conditioning of the interaction matrix and of the combination matrix in the system's behavior [FED 89, NEL 95, SHA 97]. A good conditioning of the former leads to a good sensitivity for the system, whereas a good conditioning of the latter leads to a good robustness for the control with respect to measurement errors. Even if all of the methods discussed above provide very satisfactory practical results, there is still much work to be done to determine which visual data are the most relevant.

6.3.3. *Hybrid tasks*

We will now consider the case where the k visual data that are chosen do not constrain all of the system's degrees of freedom. The visual tasks associated with the k constraints can then be categorized depending on the virtual link, an extension of the concept of links between solids, between the sensor and its environment.

6.3.3.1. *Virtual links*

The constraints $\mathbf{s}(\mathbf{p}(t)) - \mathbf{s}^* = 0$ induced by the visual data define, when they are achieved, a *virtual link* between the robot and its environment. Because $\dot{\mathbf{s}} = 0$ is an immediate consequence of $\mathbf{s}(\mathbf{p}(t)) = \mathbf{s}^*$, the set $\mathcal{S}^*$ of movements that leave $\mathbf{s}$ unchanged, or:

$$\mathcal{S}^* = \text{Ker}\ \mathbf{L_s} \qquad [6.87]$$

enables us to fully characterize this virtual link.

For a pose $\mathbf{p}$ where these constraints are satisfied, the dimension N of $\mathcal{S}^*$ is called the *class* of the virtual link in point $\mathbf{p}$. Let $m = 6 - N$. When $m = k$, the k constraints resulting from the visual data are independent. Intuitively, the size of $\mathbf{s}$ then corresponds to the number m of degrees of freedom that we can and wish to control using $\mathbf{s}$. As we have already seen previously, it is also possible to come across the case where the visual data are redundant ($k > m$).

Figure 6.20 lists the most common frictionless mechanical links. The class of each link and the number of unconstrained degrees of freedom (in translation (T) or in rotation (R)) that allow such a categorization are also indicated. With a vision sensor, it is possible to achieve all of these links. For example, the case of the rigid link (which constrains the robot's six degrees of freedom) has already been studied in detail in the previous section. For the other links, examples of designs, constructed from the most common geometric primitives (points, lines, cylinders, etc.) are given in [CHA 93a].

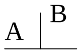
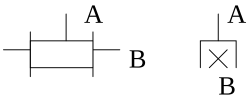
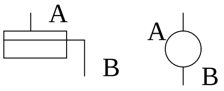
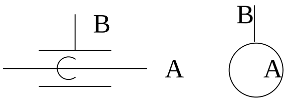
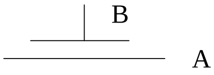
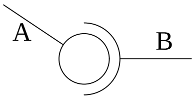
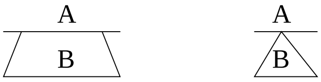
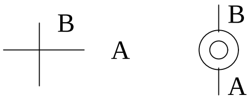
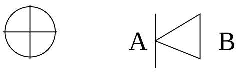
Name of the link	Class	T	R	Geometric symbol
Rigid	0	0	0	
Prismatic	1	1	0	
Rotoid	1	0	1	
Sliding pivot	2	1	1	
Planar rest	3	2	1	
Pivot ball	3	0	3	
Rectilinear	4	2	2	
Annulate linear	4	1	3	
Punctual	5	2	3	

Figure 6.20. *Mechanical links*

As an example, we will explain the case of the pivot ball link. It can be achieved if the camera is observing a sphere the center of which is located on the optical axis, and therefore has the coordinates $\mathbf{X}_0 = (0, 0, Z_0)$. The image of the sphere is then a centered circle, and if we select in $\mathbf{s}$ the area a of the circle and the coordinates (x_g, y_g) of its center of gravity ($\mathbf{s} = (a, x_g, y_g)$), the interaction matrix has the value (see [6.34]):

$$\mathbf{L_s} = \begin{bmatrix} 0 & 0 & 2a/Z_g & 0 & 0 & 0 \\ -1/Z_g & 0 & 0 & 0 & -(1+r^2) & 0 \\ 0 & -1/Z_g & 0 & 1+r^2 & 0 & 0 \end{bmatrix} \qquad [6.88]$$

where $1/Z_g = Z_0/(Z_0^2 - R^2)$ and $r^2 = R^2/(Z_0^2 - R^2)$, r being the circle's radius in the image, and R the sphere's radius. By expressing the interaction matrix in the sphere's center using the torsor frame change formula given in [6.6], we get:

$$\begin{aligned} \mathbf{L_s} &= \begin{bmatrix} 0 & 0 & 2a/Z_g & 0 & 0 & 0 \\ -1/Z_g & 0 & 0 & 0 & -(1+r^2) & 0 \\ 0 & -1/Z_g & 0 & 1+r^2 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbb{I}_3 & [\mathbf{X}_0]_\times \\ \mathbf{0}_3 & \mathbb{I}_3 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 2a/Z_g & 0 & 0 & 0 \\ -1/Z_g & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/Z_g & 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

which gives the form we wanted for $\mathcal{S}^*$:

$$\mathcal{S}^* = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad [6.89]$$

It is of course possible to select 3-D visual data to achieve this link (if 3-D measurements are available). In this case, we simply have to choose the three coordinates of the sphere's center in $\mathbf{s}$.

We will now discuss in detail how to define a hybrid task combining a visual task controlling $m(< 6)$ degrees of freedom and a secondary task.

6.3.3.2. *Hybrid task function*

Very often, regulating a visual task is not the only objective and this task must be combined with another, such as for example tracking a trajectory or avoiding joint blocking of the robot, since, aside from the case where the visual task consists of achieving a rigid link, other tasks achieve virtual links with a non-zero class, that is to say with degrees of freedom that are not constrained by the link. For example, a translation movement along the axis of a prismatic link leaves this link unchanged, and it may be advisable to put this available degree of freedom to use on a second objective.

Combining these two objectives can sometimes be done with a simple summation [NEL 94b]. The control law then reaches a compromise that can lead to none of the initial objectives being achieved. Here is another, more elegant approach, described in [SAM 91], among others. It consists of considering the visual task as the priority and to express the second objective as a cost function to minimze under the constraint that the visual task be achieved. The use of this approach is starting to be common in visual servoing. [CHA 94, COS 95, BER 00] give examples where the secondary task consists of performing trajectory tracking. This can be useful for inspection and conformity control applications, but also to perform the 3-D reconstruction of the considered objects [CHA 96] or to ensure that a system is properly positioned [COL 02]. Avoiding blocking and singularities is discussed in [MAR 96]. Secondary tasks can also be visual tasks, to try to avoid occultations for example [MAR 98], or they can be built from measurements provided by exteroceptive sensors. For example, combining a positioning task by visual servoing with an obstacle avoiding task using a laser type proximity sensor was studied in [CAD 00] in the context of mobile robotics.

Let $\mathbf{e}_1$ be the visual task function and h_s the cost function to minimize, the gradient of which is $\mathbf{e}_2^\top$. The function $\mathbf{e}_1$ has a dimension of $m = n - N \leq k$ where n is the system's number of degrees of freedom, N the class of the desired virtual link, and k the number of visual data used in $\mathbf{s}$. This task function is always written:

$$\mathbf{e}_1 = \mathbf{C}\,(\mathbf{s} - \mathbf{s}^*) \qquad [6.90]$$

where the size of $\mathbf{C}$ is now an $m \times k$ matrix that has full rank m in order for the m components of $\mathbf{e}_1$ to be independent and to control the m desired degrees of freedom. If $\mathbf{C}$ is chosen constant, the interaction matrix of $\mathbf{e}_1$, with a size of $m \times n$ and a full rank equal to m, is given by:

$$\mathbf{L}_{\mathbf{e}_1} = \mathbf{C}\mathbf{L}_{\mathbf{s}} \qquad [6.91]$$

and notice that Ker $\mathbf{L}_{\mathbf{e}_1} =$ Ker $\mathbf{L}_{\mathbf{s}}$.

A task function $\mathbf{e}$ that performs the minimizing of h_s under the constraint $\mathbf{e}_1 = 0$ can be expressed in the form [SAM 91]:

$$\mathbf{e} = \mathbf{W}^+\mathbf{e}_1 + (\mathbb{I}_n - \mathbf{W}^+\mathbf{W})\,\mathbf{e}_2 \tag{6.92}$$

where $\mathbf{W}$ is an $m \times n$ matrix with a full rank equal to m and such that Ker $\mathbf{W}$ = Ker $\mathbf{L_s}$. Hence the matrix $(\mathbb{I}_n - \mathbf{W}^+\mathbf{W})$ is therefore an orthogonal projection operator onto the kernel of $\mathbf{L_s}$ meaning that:

$$(\mathbb{I}_n - \mathbf{W}^+\mathbf{W})\,\mathbf{x} \in \text{Ker}\,\mathbf{L_s}\ ,\ \forall \mathbf{x} \in \mathbb{R}^n \tag{6.93}$$

Hence whatever the function to minimize, and therefore the secondary task $\mathbf{e}_2$, only movements that do not disrupt $\mathbf{e}_1$ will be applied, which also implies that h_s will not necessarily reach its minimum value.

However, because the exact value of the interaction matrix $\mathbf{L_s}$ is usually unknown, $\mathbf{W}$ cannot be constructed based on an approximation or an estimate $\widehat{\mathbf{L_s}}$. If the kernel of $\mathbf{W}$ is different from the kernel of $\mathbf{L_s}$, then the secondary task can lead to disruptions in the achievement $\mathbf{e}_1$. In practice, these disruptions turn out to be not too harmful, unless if the estimate of the interaction matrix is completely erroneous.

To construct the matrix $\mathbf{W}$, the simplest case occurs when $\mathbf{L_s}$ is a full rank matrix with a rank equal to $m = k$. We can then directly take $\mathbf{W} = \widehat{\mathbf{L_s}}$. Otherwise, the m lines of $\mathbf{W}$ can be comprised of the m basis vectors of the subspace generated by $\widehat{\mathbf{L_s}}$. Note that if $\mathbf{L_s}$ has a rank equal to n (in other words if the visual task constrains the system's n degrees of freedom), we can choose $\mathbf{W} = \mathbb{I}_n$. It is then impossible of course to take into account a secondary task, since in that case we have $\mathbf{e} = \mathbf{e}_1$. This also indicates that the task function discussed in [6.92] is a generalization of the previous one in [6.59].

By performing the same analysis as the one described in the beginning of section 6.3.2, we get the following control law:

$$\mathbf{v_q} = \widehat{\mathbf{L_e}}^{-1}\,(-\lambda\,\mathbf{e} - \frac{\widehat{\partial \mathbf{e}}}{\partial t}) \tag{6.94}$$

and the sufficient stability condition:

$$\mathbf{L_e}\widehat{\mathbf{L_e}}^{-1} > 0 \tag{6.95}$$

Given the difficulty of calculating $\mathbf{L_e}$ in practice, it is possible to show [SAM 91] that the condition [6.95] is usually respected if we have:

$$\mathbf{L}_{\mathbf{e}_1}\mathbf{W}^+ > 0 \qquad [6.96]$$

We can then set $\widehat{\mathbf{L_e}} = \mathbb{I}_n$, which leads to the following control law:

$$\mathbf{v_q} = -\lambda\, \mathbf{e} - \widehat{\frac{\partial \mathbf{e}}{\partial t}} \qquad [6.97]$$

As with the rigid link seen in 6.3.2, it is much better to choose $\mathbf{C} = \mathbb{I}_m$ if the visual data are not redundant ($k = m$). We should point out by the way that it is much less difficult to select non-redundant 2-D visual data for links other than the rigid link (see for example the case of the pivot ball link described in the previous section). By choosing $\mathbf{C} = \mathbb{I}_m$, Condition [6.96] can be expressed simply as $\mathbf{L_s}\mathbf{W}^+ > 0$, which is respected if the estimate used to build $\mathbf{W}$ is right. We will also have an exponential decrease for each component of $\mathbf{s}$. In the case where the visual data are redundant ($k > m$), we can choose:

$$\mathbf{C} = \mathbf{W}\widehat{\mathbf{L_s}}^+_{|\mathbf{s}=\mathbf{s}^*} \qquad [6.98]$$

If $\mathbf{W}$ is also built from $\widehat{\mathbf{L_s}}_{|\mathbf{s}=\mathbf{s}^*}$ (and therefore constant), Condition [6.96] is expressed as:

$$\mathbf{W}\widehat{\mathbf{L_s}}^+_{|\mathbf{s}=\mathbf{s}^*}\mathbf{L_s}\mathbf{W}^+ > 0 \qquad [6.99]$$

which at best can only be respected in the neighborhood of configurations such that $\mathbf{s} = \mathbf{s}^*$. In practice, it is also possible to consider matrices $\mathbf{W}$ and $\mathbf{C}$ calculated at each iteration from an estimate of the current value of the interaction matrix, but this choice no longer allows the analytical form of $\mathbf{L}_{\mathbf{e}_1}$ to be easily determined.

Finally, if the secondary task makes it possible to know $\frac{\partial \mathbf{e}_2}{\partial t}$, we can choose:

$$\widehat{\frac{\partial \mathbf{e}}{\partial t}} = \mathbf{W}^+\widehat{\frac{\partial \mathbf{e}_1}{\partial t}} + \left(\mathbb{I}_n - \mathbf{W}^+\mathbf{W}\right)\frac{\partial \mathbf{e}_2}{\partial t} \qquad [6.100]$$

where the term $\widehat{\frac{\partial \mathbf{e}_1}{\partial t}}$, if it is properly estimated, allows possible tracking errors to be reduced if the object being observed is in motion.

Figure 6.21 illustrates hybrid tasks. We considered the case of positioning a camera with respect to a sphere, thus creating a pivot ball link as we saw in section 6.3.3.1. The secondary task corresponds to a movement at a constant speed along the rotation speed's three components. Notice in Figure 6.21e the exponential decrease of the three components of $\mathbf{s}$ (the matrices $\mathbf{W}$ and $\mathbf{C}$ are also calculated at each iteration of the control law). Notice also in Figure 6.21f that the secondary task is only properly achieved after the convergence of $\mathbf{s}$ to $\mathbf{s}^*$ (except for the ω_z component which is not involved in the convergence). Finally, note on this same figure that the projection operator $\mathbb{I}_n - \mathbf{W}^+\mathbf{W}$ induces translation movements v_x and v_y to compensate for the rotation movements ω_y and ω_x caused by the secondary task, and thus preserve the sphere's image as a centered circle.

6.3.4. *Target tracking*

In this short section, we consider the case of a mounted camera observing a mobile object. However, the rules described below are also valid for a scene camera driven by an unknown movement.

A significant part of the studies conducted in visual servoing that deal with target tracking consider the object's movement as a perturbation, that must be eliminated as quickly and efficiently as possible [PAP 93, GAN 02]. Other studies use prior information on the trajectory or the type of movement of the object [ALL 93, HAS 95, RIZ 96, GIN 05]. Furthermore, the use of an integrator is very common in control theory to eliminate drag errors. Let $\mathbf{I}_k$ be the estimate of $\frac{\partial \mathbf{e}_1}{\partial t}$ at the iteration k. We then have:

$$\begin{aligned} \mathbf{I}_{k+1} &= \mathbf{I}_k + \mu\, \mathbf{e}_{1k} \ \text{ with } \ \mathbf{I}_0 = 0 \qquad [6.101] \\ &= \mu \sum_{j=0}^{k} \mathbf{e}_{1j} \end{aligned}$$

where μ is the gain of the integrator. This techinque can only function properly in cases where the object has a constant speed, since we have $\mathbf{I}_{k+1} = \mathbf{I}_k$ if and only if $\mathbf{e}_{1k} = 0$. This implies that the drag errors are not completely eliminated if the object's movement is more complex.

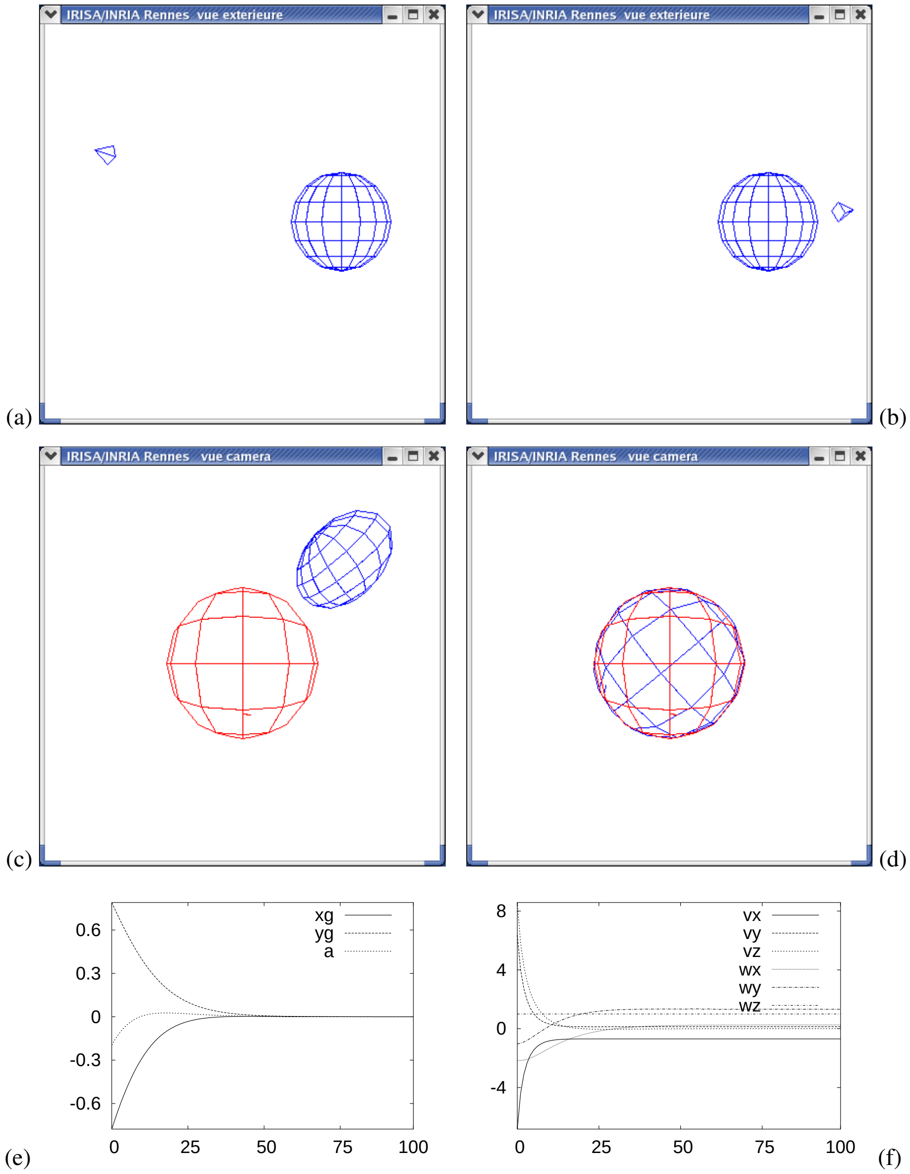

IRISA/INRIA Rennes vue exterieure
(a)
IRISA/INRIA Rennes vue exterieure
(b)
IRISA/INRIA Rennes vue camera
(c)
IRISA/INRIA Rennes vue camera
(d)
xg
yg
a
0.6
0.3
0
-0.3
-0.6
0
25
50
75
100
(e)
vx
vy
vz
wx
wy
wz
8
4
0
-4
0
25
50
75
100
(f)

Figure 6.21. *Positioning with respect to a sphere then movement around this sphere: (a) camera's initial pose with respect to the sphere, (b) camera's final pose, (c) superposition of the initial image and of the desired image, (d) superposition of the final image and of the desired image, (e) components of* $\mathbf{s}$ *calculated at each iteration, (f) components of* $\mathbf{v}_c$ *calculated at each iteration.*

Other method classes consist of estimating as reliably as possible the object's speed in the image. This is because if it is possible to measure the camera's speed, an estimate of the object's speed is given by:

$$\widehat{\frac{\partial \mathbf{e_1}}{\partial t}} = \widehat{\dot{\mathbf{e}}_1} - \widehat{\mathbf{L_{e_1}}}\mathbf{v_c} \qquad [6.102]$$

where $\widehat{\dot{\mathbf{e}}_1}$ is for example measured at the iteration k by $\widehat{\dot{\mathbf{e}}_1}_k = (\mathbf{e_1}_k - \mathbf{e_1}_{k-1})/\Delta t$, Δt being the system's sampling period. This leads us to an indirect adaptive control system and we can then use a Kalman filter (for example) to smooth this estimate. In [COR 93], such a filter based on a simple model of a constant speed state is presented. In [CHA 93b], a constant acceleration and correlated noise model was chosen. Finally, what is referred to as the GLR algorithm (Generalized Likelihood Ratio) is used in [BENS 95], to detect, estimate and compensate for possible gaps in the objects direction or speed amplitude.

6.4. Other exteroceptive sensors

All of the rules described in this chapter are valid for any exteroceptive sensor. The only characteristic involves the modeling of the interaction matrix between the sensor in question and its environment [SAM 91].

Consider for example a narrow field proximity sensor that provides the distance Z between this sensor and the closest object in the sensor's direction (see Figure 6.22). If we assume that the object's surface is perpendicular to the sensor's axis, the interaction matrix associated with Z is of the form:

$$\mathbf{L}_Z = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \qquad [6.103]$$

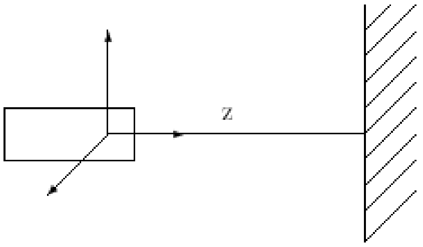


Figure 6.22. *Modeling of a proximity sensor*

Various robotic tasks can then be performed, such as obstacle avoiding or most of the virtual links that we saw previously, by selecting the appropriate number and direction of sensors. For example, a possible choice for creating a planar rest is the configuration shown in Figure 6.23. Expressed in the frame of reference $R_O = (O, \vec{x}, \vec{y}, \vec{z})$, the interaction matrix $\mathbf{L}_{Z_i}$ associated with each sensor S_i is given by (see [6.6]):

$$\mathbf{L}_{Z_i} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbb{I}_3 & -[\mathbf{X}_i]_\times \\ 0 & \mathbb{I}_3 \end{bmatrix} \qquad [6.104]$$

so in the end:

$$\mathbf{L}_{Z_i} = \begin{bmatrix} 0 & 0 & -1 & -Y_i & X_i & 0 \end{bmatrix} \qquad [6.105]$$

where $\mathbf{X}_i = (X_i, Y_i, 0)$ are the coordinates of S_i in R_O. This shows that the interaction matrix that integrates the four sensors has rank 3 and its kernel is:

$$\mathcal{S}^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad [6.106]$$

The use of force sensors has also been the subject of many studies [SAM 91]. Finally, we can mention [KHA 96] where the sensor in question is comprised of a camera rigidly attached to laser planes.

6.5. Conclusion

In this chapter, we only considered the most common case, that is controlling a system that has six degrees of freedom, for which the selection of relevant visual data to perform the task is the most difficult problem, especially if the rigid link is an objective. Many simplifications occur if the system has fewer degrees of freedom. As an example, the orientation control of a camera mounted on a pan-tilt head, for a centering or mobile object tracking task, presents no difficulties when it comes to modeling the task. No 3-D data are even necessary. The main difficulty involves the image processing aspects in order to grasp real, complex objects.

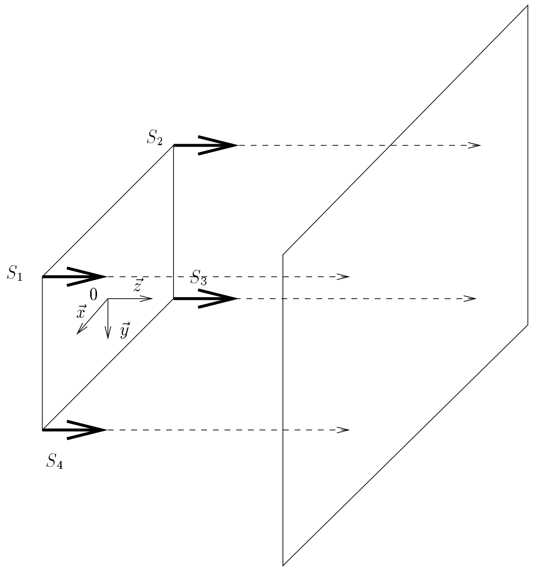


Figure 6.23. *Planar rest link*

For many years, servoing technique could actually only be applied to extremely simple objects (usually producing binary images), because of the slow processing pace for complex images. Advances in algorithms, particularly in the field of motion analysis, but most of all the enormous gains in computing power, have made it possible to consider real, video-rate applications [CRE 00, CRE 01b, TON 97, VIN 00, COM 04]. We should also mention studies where the visual data are no longer geometric, but instead describe a movement datum in the image sequence [GRO 95, CRE 01a].

The major advances to come in visual servoing involve significant progress needed in determining the optimal visual data and taking into account unknown objects, that do not require prior 3-D information. The studies in this field of projective geometry have already shown promising results [MAL 00, RUF 99]. In practice, if the learning of the desired image is impossible, obtaining the command when working with a crudely calibrated system can be quite difficult. Furthermore, aspects of trajectory planning in the sensor's space [HOS 95b] are currently the subject of successful studies [MEZ 02, COW 02, ZAN 04]. With the same basic idea, aspects of chains of visual task would deserve to be developed further. It would then be possible to broaden the very local aspect of the current techniques. Likewise, merging measurements provided by several exteroceptive sensors, which may or may not be of different types, inside control laws should make it possible to deal with many new applications [CAD 00, MAL 01].

Also, the development of omni-directional vision sensors [BAK 99, GAS 00] paves the way for many studies in the field of visual data modeling, given how peculiar and complex its projection model is [BAR 02, MEZ 04].

Finally, the use of visual servoing has expanded considerably over the past few years in several fields. We will mention of course mobile robotics, particularly applications in automatic driving of vehicles [DIC 91, PIS 95, KHA 98, TSAK 98]. In addition to the image processing problems, the main difficulty resides in designing control laws that take into account the non-holonomic constraints of this type of robots. We can also mention the control of flying machines (drones, blimps, helicopters) [ZHA 99, HAM 02, ZWA 02, RIV 04] where one of the difficulties come from the fact that these aircrafts are under-driven. Finally, the field of medical robotics is showing great promise for the application of visual servoing techniques [KRU 03, GIN 05].

6.6. Bibliography

[ALL 93] ALLEN P.K., YOSHIMI B., TIMCENKO A., MICHELMAN P., "Automated tracking and grasping of a moving object with a robotic hand-eye system", *IEEE Transactions on Robotics and Automation*, vol. 9, no. 2, p. 152-165, 1993.

[AND 02] ANDREFF N., ESPIAU B., HORAUD R., "Visual servoing from lines", *International Journal of Robotics Research*, vol. 21, no. 8, p. 679-700, 2002.

[BAK 99] BAKER S., NAYAR S.K., "A theory of single-viewpoint catadioptric image formation", *International Journal of Computer Vision*, vol. 35, no. 2, p. 1-22, 1999.

[BAR 02] BARRETO J.P., MARTIN F., HORAUD R., "Visual servoing/tracking using central catadioptric images", *Eighth International Symposium on Experimental Robotics (ISER'02)*, Bombay, p. 863-869, 2002.

[BEN 03] BENHIMANE S., MALIS E., "Vision-based control with respect to planar and nonplanar objects using a zooming camera", *IEEE International Conference on Advanced Robotics (ICAR'03)*, Coimbra, p. 991-996, 2003.

[BENS 95] BENSALAH F., CHAUMETTE F., "Compensation of abrupt motion changes in target tracking by visual servoing", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'95)*, Pittsburgh, p. 181-187, 1995.

[BER 00] BERRY F., MARTINET P, GALLICE J., "Turning around an unknown object using visual servoing", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00)*, Takamatsu, p. 257-262, 2000.

[BEY 92] BEYER H., "Accurate calibration of CCD cameras", *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'92)*, Urbana Champaign, p. 96-101, 1992.

[CAD 00] CADENAT V., SOUERES P., COURDESSES M., "Using redundancy to perform a vision-based task amidst obstacles", *Sixth International IFAC Symposium on Robot Control (SYROCO'00)*, Vienna, p .205-210, 2000.

[CHA 93a] CHAUMETTE F., RIVES P., ESPIAU B., "Classification and realization of the different vision-based tasks", In [HAS 93a], p. 199-228, 1993.

[CHA 93b] CHAUMETTE F., SANTOS A., "Tracking a moving object by visual servoing", *Twelfth World Congress IFAC*, Sydney, vol. 9, p. 643-648, 1993.

[CHA 94] CHAUMETTE F., "Visual servoing using image features defined upon geometrical primitives", *Thirty-third IEEE Conference on Decision and Control (CDC'94)*, Orlando, vol. 4, p. 3782-3787, 1994.

[CHA 96] CHAUMETTE F., BOUKIR S., BOUTHEMY P., JUVIN D., "Structure from controlled motion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, p. 492-504, 1996.

[CHA 98] CHAUMETTE F., "Potential problems of stability and convergence in image-based and position-based visual servoing", In *The Confluence of Vision and Control*, LNCIS, vol. 237, p. 66-78, Springer-Verlag, 1998.

[CHA 00] CHAUMETTE F., MALIS E., "2 1/2 D visual servoing: A possible solution to improve image-based and position-based visual servoings", *IEEE International Conference on Robotics and Automation (ICRA'00)*, San Francisco, vol. 1, p. 630-635, 2000.

[CHA 04] CHAUMETTE F., "Image moments: A general and useful set of features for visual servoing", *IEEE Transactions on Robotics*, vol. 20, no. 4, p. 713-723, 2004.

[CIP 97] CIPOLLA R., HOLLINGHURST N., "Visually guided grasping in unstructured environment", *Robotics and Autonomous Systems*, vol. 19, p. 337-346, 1997.

[COL 00] COLLEWET C., CHAUMETTE F., "A contour approach for image-based control of objects with complex shape", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00)*, Takamatsu, p. 751-756, 2000.

[COL 02] COLLEWET C., CHAUMETTE F., "Positioning a camera with respect to planar objects of unknown shape by coupling 2D visual servoing and 3D estimations", *IEEE Transactions on Robotics and Automation*, vol. 18, no. 3, p.322-333, 2002.

[COLO 99] COLOMBO C., ALLOTTA B., "Image-based robot task planning and control using a compact visual representation", *IEEE Transactions on Systems, Man and Cybernetics*, vol. A-29, no. 1, p. 92-100, 1999.

[COM 04] COMPORT A., MARCHAND E., CHAUMETTE F., "Robust model-based tracking for robot vision", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04)*, Sendai, vol. 1, p. 692-697, 2004.

[COR 93] CORKE P., GOODS M., "Controller design for high performance visual servoing", *Twelfth World Congress IFAC*, Sydney, vol. 9, p. 395-398, 1993.

[COR 01] CORKE P., HUTCHINSON S., "A new partitioned approach to image-based visual servo control", *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, p. 507-515, 2001.

[COS 95] COSTE-MANIÈRE E., COUVIGNOU P., KHOSLA P.K., "Visual servoing in the task-function framework: A contour following task", *Journal of Intelligent and Robotic Systems*, vol. 12, p. 1-21, 1995.

[COW 02] COWAN N., WEINGARTEN J., KODITSCHEK D., "Visual servoing via navigation functions", *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, p. 521-533, 2002.

[CRE 00] CRÉTUAL A., CHAUMETTE F., "Dynamic stabilization of a pan and tilt camera for sub-marine image visualization", *Computer Vision and Image Understanding*, vol. 79, no. 1, p. 47-65, 2000.

[CRE 01a] CRÉTUAL A., CHAUMETTE F., "Visual servoing based on image motion", *International Journal of Robotics Research*, vol. 20, no. 11, p. 857-877, 2001.

[CRE 01b] CRÉTUAL A., CHAUMETTE F., "Application of motion-based visual servoing to target tracking", *International Journal of Robotics Research*, vol. 20, no. 11, p. 878-890, 2001.

[DEG 97] DEGUCHI K., "Direct interpretation of dynamic images and camera motion for visual servoing without image feature correspondence", *Journal of Robotics and Mechatronics*, vol. 9, no. 2, p. 104-110, 1997.

[DEM 95] DEMENTHON D., DAVIS L.S., "Model-based object pose in 25 lines of code", *International Journal of Computer Vision*, vol. 15, no. 1/2, p. 123-141, 1995.

[DEN 02] DENG L., JANABI-SHARIFI F., WILSON W., "Stability and robustness of visual servoing methods", *IEEE International Conference on Robotics and Automation (ICRA'02)*, Washington DC, p. 1604-1609, 2002.

[DHO 89] DHOME M., RICHETIN M., LAPRESTE J.T., RIVES G., "Determination of the attitude of 3D objects from a single perspective view", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 12, p. 1265-1278, 1989.

[DHO 90] DHOME M., LAPRESTE J.T., RIVES G., RICHETIN M., "Spatial localization of modelled objects of revolution in monocular perspective vision", *European Conference on Computer Vision (ECCV'90)*, Antibes, LNCS, vol. 427, p. 475-485, Springer-Verlag, 1990.

[DIC 91] DICKMANNS E., CHRITIANS T., "Relative 3D state estimation for autonomous visual guidance of road vehicles", *Robotics and Autonomous Systems*, vol. 7, no. 2/3, p. 85-98, 1991.

[DRU 99] DRUMMOND T., CIPOLLA R., "Visual tracking and control using Lie algebra", *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'99)*, Fort Collins, vol. 2, p. 652-657, 1999.

[ESP 92] ESPIAU B., CHAUMETTE F., RIVES P., "A new approach to visual servoing in robotics", *IEEE Transactions on Robotics and Automation*, vol. 8, no. 6, p. 313-326, 1992.

[ESP 93] ESPIAU B., "Effect of camera calibration errors on visual servoing in robotics", *International Symposium on Experimental Robotics (ISER'93)*, Kyoto, 1993.

[FAU 93] FAUGERAS O., *Three-Dimensionnal Computer Vision: A Geometric Viewpoint*, MIT Press, Cambridge, 1993.

[FED 89] FEDDEMA J., LEE C., MITCHELL O., “Automatic selection of image features for visual servoing of a robot manipulator”, *IEEE International Conference on Robotics and Automation (ICRA’89)*, Scottsdale, vol. 2, p. 832-837, 1989.

[GAN 02] GANGLOFF J., DE MATHELIN M., “Visual servoing of a 6 DOF manipulator for unknown 3D profile following”, *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, p. 511-520, 2002.

[GAS 00] GASPAR J., WINTERS N., SANTOS-VICTOR J., “Vision-based navigation and environmental representations with an omni-directional camera”, *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, p. 890-898, 2000.

[GIN 05] GINHOUX R., GANGLOFF J., DE MATHELIN M., SOLER L., ARENAS SANCHEZ M., MARESCAUX J., “Active filtering of physiological motion in robotized surgery using predictive control”, *IEEE Transactions on Robotics*, vol. 21, no. 1, p. 67-79, 2005.

[GRO 95] GROSSO E., TISTARELLI M., “Active/dynamic stereo vision”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 11, p. 1117-1127, 1995.

[HAG 95] HAGER G., CHANG W., MORSE A.S., “Robot feedback control based on stereo vision: Towards calibration-free hand-eye coordination”, *IEEE Control Systems Magazine*, vol. 15, no. 1, p. hbox30-39, 1995.

[HAM 02] HAMEL T., MAHONY R., “Visual servoing of an under-actuated dynamic rigid body system: An image-based approach”, *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, p. 187-198, 2002.

[HAR 89] HARALICK R., JOO H., LEE C.N., ZHUANG X., VAIDYA V., KIM M., “Pose estimation from corresponding point data”, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 6, p. 1426-1445, 1989.

[HAS 93a] HASHIMOTO K. (ed.), *Visual Servoing: Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*, World Scientific Series in Robotics and Automated Systems, vol. 7, World Scientific Press, Singapour, 1993.

[HAS 93b] HASHIMOTO K., KIMURA H., “LQ Optimal and non linear approaches to visual servoing”, In [HAS 93a], p. 165-198, 1993.

[HAS 95] HASHIMOTO K., KIMURA H., “Visual servoing with non linear observer”, *IEEE International Conference on Robotics and Automation (ICRA’95)*, Nagoya, p. 484-489, 1995.

[HAS 96] HASHIMOTO K., EBINE T., KIMURA H., “Visual servoing with hand-eye manipulator-optimal control approach”, *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, p. 766-774, 1996.

[HOR 89] HORAUD R., CONIO B., LEBOULLEUX O., LACOLLE B., “An analytic solution for the perspective 4-points problem”, *Computer Vision, Graphics, and Image Processing*, vol. 47, no. 1, p. 33-44, 1989.

[HOR 95] HORAUD R., DORNAIKA F., “Hand-eye calibration”, *International Journal of Robotics Research*, vol. 14, no. 3, p. 195-210, 1995.

[HOR 98] HORAUD R., DORNAIKA F., ESPIAU B., "Visually guided object grasping", *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, p. 525-532, 1998.

[HOS 94] HOSODA K., ASADA M., "Versatile visual servoing without knowledge of true jacobian", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, Munich, 1994.

[HOS 95a] HOSODA K., MORIYAMA H., ASADA M., "Visual servoing utilizing zoom mechanism", *IEEE International Conference on Robotics and Automation (ICRA'95)*, Nagoya, 1995.

[HOS 95b] HOSODA K., SAKAMOTO K., ASADA M., "Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3D reconstruction", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'95)*, Pittsburgh, vol. 3, p.29-34, 1995.

[HUT 96] HUTCHINSON S., HAGER G., CORKE P., "A tutorial on visual servo control", *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, p. 651-670, 1996.

[IWA 05] IWATSUKI M., OKIYAMA N., "A new formulation of visual servoing based on cylindrical coordinate system", *IEEE Transactions on Robotics*, vol. 21, no. 2, p. 266-273, 2005.

[JAG 97] JÄGERSAND M., FUENTES O., NELSON R., "Experimental evaluation of uncalibrated visual servoing for precision manipulation", *IEEE International Conference on Robotics and Automation (ICRA'97)*, Albuquerque, vol. 3, p. 2874-2880, 1997.

[KEL 96] KELLY R., "Robust asymptotically stable visual servoing of planar robots", *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, p. 759-766, 1996.

[KHA 96] KHADRAOUI D., MOTYL G., MARTINET P., GALLICE J., CHAUMETTE F., "Visual servoing in robotics scheme using a camera/laser-stripe sensor", *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, p. 743-750, 1996.

[KHA 98] KHADRAOUI D., ROUVEURE R., DEBAIN C., MARTINET P., BONTON P., GALLICE J., "Vision-based control in driving assistance of agricultural vehicles", *International Journal of Robotics Research*, vol. 17, no. 10, p. 1040-1054, 1998.

[KHAL 02] KHALIL W., DOMBRE E., *Modeling, identification and control of robots*, Hermes Penton Sciences, London, 2002.

[KIN 94] KINOSHITA K., DEGUCHI K., "Simultaneous determination of camera pose and intrinsic parameters by visual servoing", *International Conference on Pattern Recognition (ICPR'94)*, Jerusalem, vol. 1, p. 285-290, 1994.

[KRA 05] KRAGIC D., CHRISTENSEN H., "Special issue on advances in robot vision", *Robotics and Autonomous System*, vol. 52, no. 1, 2005.

[KRU 03] KRUPA A., GANGLOFF J., DOIGNON C., DE MATHELIN M., MOREL G., LEROY J., SOLER L., MARESCAUX J., "Autonomous 3-D positioning of surgical instruments in robotized laparoscopic surgery using visual servoing", *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, p. 842-853, 2003.

[LAP 04] LAPRESTÉ J.-T., JURIE F., CHAUMETTE F., "An efficient method to compute the inverse jacobian matrix in visual servoing", *IEEE International Conference on Robotics and Automation (ICRA'04)*, New Orleans, vol. 1, p. 727-732, 2004.

[LOW 87] LOWE D.G., "Three-dimensional object recognition from single two-dimensional images", *Artificial Intelligence*, vol. 31, p. 355-394, 1987.

[MA 93] DE MA S., "Conics-based stereo, motion estimation, and pose determination", *International Journal of Computer Vision*, vol. 10, no. 1, p. 7-25, 1993.

[MA 03] MA Y., SOATTO S., KOSECKA J., SASTRY S., *An Invitation to 3-D Vision: From Images to Models*, Springer-Verlag, 2003.

[MAL 99] MALIS E., CHAUMETTE F., BOUDET S., "2 1/2 D visual servoing", *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, p. 238-250, 1999.

[MAL 00] MALIS E., CHAUMETTE F., "2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement", *International Journal of Computer Vision*, vol. 37, no. 1, p. 79-97, 2000.

[MAL 01] MALIS E., MOREL G., CHAUMETTE F., "Robot control from disparate multiple sensors", *International Journal of Robotics Research*, vol. 20, no. 5, p. 364-378, 2001.

[MAL 02] MALIS E., CHAUMETTE F., "Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods", *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, p. 176-186, 2002.

[MAL 04] MALIS E., "Improving vision-based control using efficient second-order minimization techniques", *IEEE International Conference on Robotics and Automation (ICRA'04)*, New Orleans, p. 1843-1848, 2004.

[MAR 96] MARCHAND E., CHAUMETTE F., RIZZO A., "Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'96)*, Osaka, vol. 3, p. 1083-1090, 1996.

[MAR 98] MARCHAND E., HAGER G., "Dynamic sensor planning in visual servoing", *IEEE International Conference on Robotics and Automation (ICRA'98)*, Leuven, vol. 3, p. 1988-1993, 1998.

[MART 96] MARTINET P., GALLICE J., KHADRAOUI D., "Vision-based control law using 3D visual features", *World Automation Congress (WAC'96)*, Montpellier, vol. 3, p. 497-502, 1996.

[MART 97] MARTINET P., DAUCHER N., GALLICE J., DHOME M., "Robot control using monocular pose estimation", *Workshop on New Trends in Image-based Robot Servoing (IROS'97)*, Grenoble, p. 1-12, 1997.

[MEZ 02] MEZOUAR Y., CHAUMETTE F., "Path planning for robust image-based control", *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, p. 534-549, 2002.

[MEZ 04] MEZOUAR Y., HAJ ABDELKADER H., MARTINET P., CHAUMETTE F., "Central catadioptric visual servoing from 3D straight lines", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04)*, Sendai, vol. 1, p. 343-349, 2004.

[MIC 93] MICHEL H., RIVES P., Singularities in the determination of the situation of a robot effector from the perspective view of three points, INRIA Research Report, no. 1850, 1993.

[MOR 00] MOREL G., LEIBEZEIT T., SZEWCZYK J., BOUDET S., POT J., "Explicit incorporation of 2D constraints in vision-based control of robot manipulators", *International Symposium on Experimental Robotics (ISER'00)*, Sydney, LNCIS, vol. 250, p. 99-108, Springer-Verlag, 2000.

[NAY 96] NAYAR S.K., NEME S.A., MURASE H., "Subspace methods for robot vision", *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, p. 750-758, 1996.

[NEL 94a] NELSON B., KHOSLA P.K., "Integrating sensor placement and visual tracking strategies", *IEEE International Conference on Robotics and Automation (ICRA'94)*, p. 1351-1356, 1994.

[NEL 94b] NELSON B., KHOSLA P.K., "The resolvability ellipsoid for visual servoing", *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, p. 829-832, 1994.

[NEL 95] NELSON B., KHOSLA P.K., "Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limits avoidance", *International Journal of Robotics Research*, vol. 14, no. 3, p. 255-269, 1995.

[PAP 93] PAPANIKOLOPOULOS N., KHOSLA P.K., KANADE T., "Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision", *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, p. 14-35, 1993.

[PAP 95] PAPANIKOLOPOULOS N., "Selection of features and evaluation of visual measurements during robotic visual servoing tasks", *Journal of Intelligent and Robotics Systems*, vol. 13, p. 279-304, 1995.

[PHO 95] PHONG T.Q., HORAUD R., YASSINE A., TAO P.D., "Object pose from 2D to 3D point and line correspondences", *International Journal of Computer Vision*, vol. 15, no. 3, p. 225-243, 1995.

[PIE 04] PIEPMEIER J., MC MURRAY G., LIPKIN H., "Uncalibrated dynamic visual servoing", *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, p. 143-147, 2004.

[PIS 95] PISSARD-GIBOLLET R., RIVES P., "Applying visual servoing techniques to control a mobile hand-eye system", *IEEE International Conference on Robotics and Automation (ICRA'95)*, Nagoya, 1995.

[REY 98] REYES F., KELLY R., "Experimental evaluation of fixed-camera direct visual controllers on a direct-drive robot", *IEEE International Conference on Robotics and Automation (ICRA'98)*, Leuven, vol. 3, p. 2327-2332, 1998.

[RIV 04] RIVES P., AZINHEIRA J., "Linear structures following by an airship using vanishing points and horizon line in a visual servoing scheme", *IEEE International Conference on Robotics and Automation (ICRA'04)*, New Orleans, vol. 1, p. 255-260, 2004.

[RIZ 96] RIZZI A., KODITSCHEK D., "An active visual estimator for dexterous manipulation", *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, p. 697-713, 1996.

[RUF 99] RUF A., HORAUD R., "Rigid and articulated motion seen with an uncalibrated stereo rig", *International Conference on Computer Vision (ICCV'99)*, Corfou, 1999.

[SAF 92] SAFAEE-RAD R., TCHOUKANOV I., SMITH K.C., BENHABIB B., "Three-dimensional location estimation of circular features for machine vision", *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, p. 624-639, 1992.

[SAM 91] SAMSON C., LE BORGNE M., ESPIAU B., *Robot Control: The Task Function Approach*, Clarendon Press, Oxford, 1991.

[SCH 04] SCHRAMM F., MOREL G., MICAELLI A., LOTTIN A., "Extended 2D visual servoing", *IEEE International Conference on Robotics and Automation*, New Orleans, p. 267-273, 2004.

[SHA 97] SHARMA R., HUTCHINSON S., "Motion perceptibility and its application to active vision-based servo control", *IEEE Transactions on Robotics and Automation*, vol. 13, no. 4, p. 607-616, 1997.

[SMI 94] SMITH C., PAPANIKOLOPOULOS N., "Computation of shape through controlled active exploration", *IEEE International Conference on Robotics and Automation (ICRA'94)*, San Diego, vol. 3, p. 2516-2521, 1994.

[SUH 93] SUH I.H., "Visual servoing of robot manipulators by fuzzy membership function based neural networks", In [HAS 93a], p. 285-315, 1993.

[TAH 03] TAHRI O., CHAUMETTE F., "Application of moment invariants to visual servoing", *IEEE International Conference on Robotics and Automation (ICRA'03)*, Taipeh, p. 4276-4281, 2003.

[TON 97] TONKO M., SCHÄFER K., HEIMES F., NAGEL H.-H., "Towards visually servoed manipulation of car engine parts", *IEEE International Conference on Robotics and Automation (ICRA'97)*, Albuquerque, vol. 4, p. 3166-3171, 1997.

[TSA 87] TSAI R.Y., "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses", *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, p. 323-344, 1987.

[TSA 89] TSAI R.Y., LENZ R., "A new technique for fully autonomous and efficient 3D robotics hand-eye calibration", *IEEE Transactions on Robotics and Automation*, vol. 5, p. 345-358, 1989.

[TSAK 98] TSAKIRIS D., RIVES P., SAMSON C., "Extending visual servoing techniques to nonholonomic mobile robots", In *The Confluence of Vision and Control*, *LNCIS*, vol. 237, p. 106-117, Springer-Verlag, 1998.

[VIN 00] VINCZE M., HAGER G. (eds.), *Robust Vision for Vision-based Control of Motion*, IEEE Press, Piscataway, 2000.

[WEI 84] WEISS L.E., Dynamic Visual Servo Control of Robots. An Adaptive Image-based Approach, PhD Thesis, CMU-RI-TR-84-16, Carnegie Mellon University, 1984.

[WEL 96] WELLS G., VENAILLE C., TORRAS C., "Vision-based robot positioning using neural networks", *Image and Vision Computing*, vol. 14, p. 715-732, 1996.

[WIL 96] WILSON W., HULLS C., BELL G., "Relative end-effector control using cartesian position-based visual servoing", *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, p. 684-696, 1996.

[ZAN 04] ZANNE P., MOREL G., PLESTAN F., "A robust 3D vision based control and planning", *IEEE International Conference on Robotics and Automation (ICRA'04)*, New Orleans, p. 4423-4428, 2004.

[ZHA 99] ZHANG H., OSTROWSKI J., "Visual servoing with dynamics: Control of an unmanned blimp", *IEEE International Conference on Robotics and Automation (ICRA'99)*, Detroit, p. 618-623, 1999.

[ZHA 00] ZHANG Z.-Y., "A flexible new technique for camera calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, p. 1330-1334, 2000.

[ZWA 02] VAN DER ZWAAN S., BERNARDINO A., SANTOS-VICTOR J., "Visual station keeping for floating robots in unstructured environments", *Robotics and Autonomous Systems*, vol. 39, no. 3-4, p. 145-155, 2002.