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 robot's geometric (and therefore kinematic) models. Aside from force sensors, the purpose and applications of which were discussed in the previous chapter, there are many 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 measurements on the distances to the closest objects. They are therefore particularly well suited for obstacle avoidance tasks. As for computer vision and telemetry sensors, 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 motion planning module for a robot in a not perfectly well known environment. In computer 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 vision [FAU 93, MA 03]. The algorithmic and technological progress achieved over the past 15 years

Chapter written by François CHAUMETTE.

has made it possible to more closely link the aspects of perception with those of action, by directly integrating the measurements provided by a vision sensor into closed-loop control laws. This approach, known as visual servoing, shares some aspects with the studies on sensor-based control and is the focus of this chapter.

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

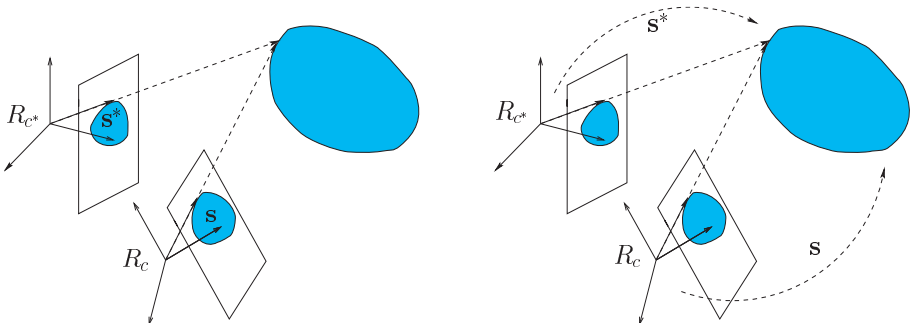

Figure 6.1. *2-D and 3-D visual servoing: in 2-D visual servoing the camera is moved from R_c to R_{c^*}, based on features $\mathbf{s}$ extracted directly from the image (left). With 3-D visual servoing, $\mathbf{s}$ is comprised of 3-D features estimated after a localization process (right)*

With a vision sensor, which provides 2-D measurements, the nature of the potential visual features is extremely rich, since it is possible to design visual servoing using both 2-D features, such as the coordinates of characteristic points in the image for example, and 3-D features, provided by a localization 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 features 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

measurements in the image, and finally a maximum decoupling between the visual features and the controlled degrees of freedom.

To study the behavior of the resulting system, a modeling phase is necessary to describe the relation between the visual features $\mathbf{s}(t)$ 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 measurements from a digital image and tracking 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 features

6.2.1. *The interaction matrix*

In order to be taken into account in a visual servoing scheme, a set $\mathbf{s}$ of k visual features needs to be defined by an application differentiable from the special Euclidean group 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 feature.

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

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

where:

– $\mathbf{L}_{\mathbf{s}}$ is a $k \times 6$ matrix, referred to as the *interaction matrix* related to $\mathbf{s}$;

– $\mathbf{v}$ is the relative instantaneous velocity (also called kinematic screw vector) between the camera and the scene, expressed in the camera's frame R_c in its origin C. More accurately, if $\mathbf{v_c}$ and $\mathbf{v_o}$ are, respectively, the kinematic screws 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 that a screw expressed in a frame of reference has its value given in the origin of this frame. Also, we will denote by $\boldsymbol{\upsilon}$ the translational velocity at the origin of the coordinate system, and by $\boldsymbol{\omega}$ the angular velocity, such that $\mathbf{v} = (\boldsymbol{\upsilon}, \boldsymbol{\omega})$. If ${}^{o}\mathbf{R}_{c}$ describes the rotation matrix from the frame 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 skew symmetric matrix defined from $\boldsymbol{\omega}$.

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 screws expressed in R_c. This is due to the fact that each component of $\mathbf{s}$ can be decomposed as the product of two screws, one called the interaction screw, and the other being of course the kinematic screw. We will see the practical advantage of this definition in section 6.3.3.1.

6.2.2. *Eye-in-hand configuration*

If we consider a camera mounted on the end-effector of a robot arm observing a static 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 features and where:

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

– ${}^{c}\mathbf{V}_{n}$ is the kinematic screw's transformation matrix from the camera's frame R_c to frame R_n. This matrix, which remains constant if the camera is rigidly attached 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 using hand-eye calibration methods [TSA 89, HOR 95]. Note that 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 the camera is observing a moving object, the differential of $\mathbf{s}$ is given by:

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

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

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

6.2.3. ~~*Hand-to-eye*~~ *configuration*

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

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

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

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

Whether the 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 localization technique (see section 6.2.5.1). If the camera is static, it is therefore more convenient to use one of the following relations:

$$\dot{\mathbf{s}} = -\mathbf{L_s}\,{}^c\mathbf{V}_\emptyset\,{}^\emptyset\mathbf{V}_n\,{}^n\mathbf{J}_n(\mathbf{q})\;\dot{\mathbf{q}} \qquad [6.10]$$

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

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 interesting because the transformation matrix ${}^{c}\mathbf{V}_{\emptyset}$ is then constant and only has to be estimated once beforehand, usually coarsely.

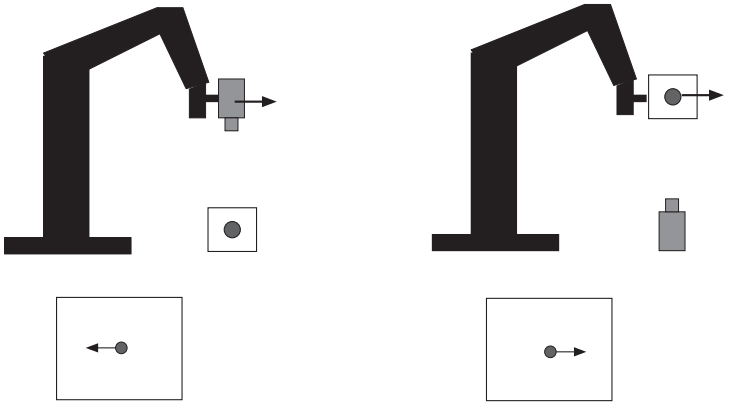

Figure 6.2. *Eye-in-hand configuration (left); eye-to-hand configuration (right)*

In the literature [HAS 93a, HUT 96], most studies focus on eye-in-hand configuration. We can however cite [ALL 93, NEL 94a, HAG 95, KEL 96, CIP 97, HOR 98, RUF 99] in which one or several cameras are used in eye-to-hand configurations.

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

6.2.4. *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:

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

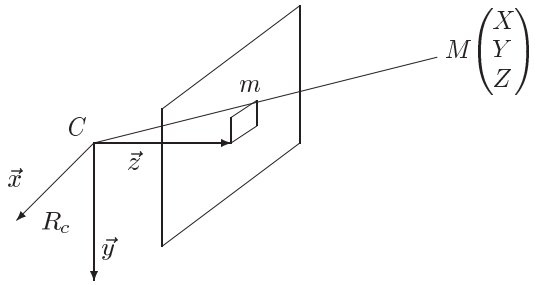

Figure 6.3. *Camera model*

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 (eye-in-hand or eye-to-hand, static or mobile point M), the speed $\dot{\mathbf{X}}$ of M according to the kinematic screw $\mathbf{v}$ between the camera and its environment is given by the fundamental kinematics equation:

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

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

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

where:

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

Notice that the terms induced by angular motions only depend on the measurements of x and y in the image. On the other hand, terms induced by translational motions are inversely proportional to the depth of the 3-D point. This effect occurs for all the visual features that can be defined in the image (and describes the classic

ambiguity in computer vision between the amplitude of a translational motion and the depth of objects). In visual servoing, it is therefore necessary to insert a 3-D knowledge, even though it is unknown beforehand, whenever trying to control a robot's degrees of freedom that imply translational motions.

Image processing algorithms provide measurements expressed in pixels. If we ignore strongly non-linear distortion effects, 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:

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

where (x_c, y_c) represents the principal 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 step [TSA 87, BEY 92, ZHA 00], but as with the elements of the hand-eye matrix, coarse approximations are usually sufficient to maintain the stability of visual servoing systems [ESP 93, MAL 99, MAL 02, DEN 02].

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

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

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 required, the same can be done for the visual features defined later on, working with features 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 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 and observed by two external 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 translational motion along the optical axis. We can still mention [HOS 95a, BEN 03].

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

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

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

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

– its projection onto the image plane:

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

– 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):

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

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

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

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

– the function μ from h_1:

$$1/Z = Ax + By + C \qquad [6.25]$$

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:

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

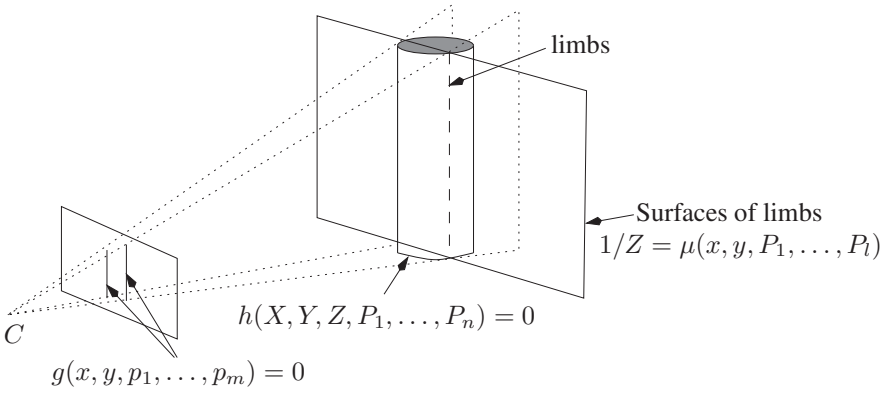

Figure 6.4. *Projection of the primitive onto the image and limb surface in the case of the cylinder*

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

$$g(x, y, \rho, \theta) = x\cos\theta + y\sin\theta - \rho = 0 \qquad [6.27]$$

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 straight line remains a straight line whatever the camera's motion, we get:

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

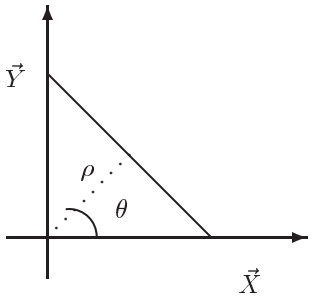

Figure 6.5. (ρ, θ) *representation of the 2-D lines*

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 features associated with these primitives in visual servoing. It is also possible to infer the interaction matrix related to features defined from several primitives (such as the orientation between two segments 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 related to visual features 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 planar 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:

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

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 related to $\boldsymbol{\theta}$ 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 related to the terms of that series [COL 00]. The resulting analytical form, however, is very complex and difficult to understand from a geometrical point of view.

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

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

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 planar or has a planar limb 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:

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

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 area speed $\dot{a}$ is equal to zero for any motion other than the expected translational motion 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 set $A = B = 0$ in [6.34] to again obtain exactly [6.15].

More generally, the interaction matrix related to 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:

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

we get:

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

with:

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

The numerical value of the interaction matrix related to 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 in practice. 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 translational terms. As we have already said, this property is true for any visual feature 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. Furthermore, 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]:

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

where:

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

and:

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

We should point out that translational motions leave α invariant 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 angular motion around the optical axis ω_z, an indication, as we could have expected, that α is a good visual feature 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 features (that is, more than the number of degrees of freedom that we wish to control), and as we will see in section 6.3.2.2, servoing stability can only be demonstrated in the neighborhood of the convergence position. Another, more promising strategy consists of determining complementary visual features, by building or selection [COR 01, IWA 05, TAH 05], 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 area 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 spatial 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 $\mathbf{s}$ of the visual features. The analytical form of the associated interaction matrix is then unknown (since it is too difficult to obtain) and the servoing is based on a purely numerical 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 for geometric visual features 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 numerical 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 the unknown parameters occurring in 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 requires an analytical determination of the interaction matrix beforehand, it cannot be applied today to servoing schemes based on visual features as complex as those resulting from a principal component analysis of the image.

6.2.5. *Interaction matrix related to 3-D visual features*

As has been mentioned before, it is also possible to choose visual features no longer expressed directly in the image, but resulting from a reconstruction phase or a 3-D localization phase [WIL 96, MART 97]. These 3-D features 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 estimation method. Dynamic vision techniques rely on the measurement of the camera's motions and of the resulting motion in the image. They are usually rather sensitive to measurement errors [SMI 94, CHA 96]. We will now briefly describe pose estimation techniques, because they are the most commonly used in 3-D visual servoing.

6.2.5.1. *Pose estimation*

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 the 3-D model of 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 constitute it:

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

Most of the time, the measurements $\mathbf{x}(t)$ are image points [HOR 89, HAR 89, DEM 95], 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], based on a numerical and iterative linear estimation [DEM 95] or based on non-linear estimation [LOW 87]. Except for very peculiar cases [HOR 89], no analytical solution to this 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 features built from $\hat{\mathbf{p}}(t)$. This is because, based on [6.41]:

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

hence:

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

The second term of this matrix product is nothing but the interaction matrix related to $\mathbf{x}$, and 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 numerical instability.

The strongest hypothesis involves the estimation's 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]; however, 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 the measurements to choose, regarding what kind to use, but also how they are arranged in the image and the 3-D space.

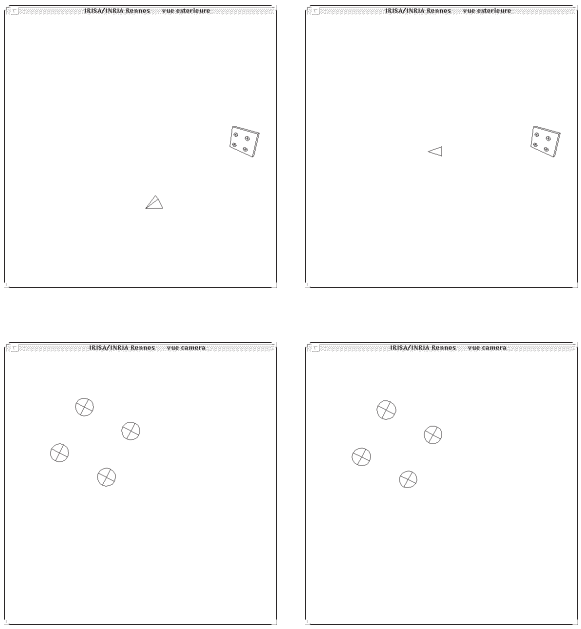

Figure 6.6. *Example of two distinct poses of the camera with respect to the object (top) that provide similar images of this object (bottom)*

Based on $\hat{\mathbf{p}}(t)$, and under the hypotheses mentioned previously, that is, 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 object's 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 object's point. If, additionally, in the context of an eye-in-hand system, 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 displacement necessary to go from R_c to R_{c^*}. With an eye-to-hand system, the same is true of course for an

object attached to the robot's end-effector between its current position and its desired position.

We will now give the interaction matrix related to 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 related to $\theta\mathbf{u}$*

Remember, first of all, that the $\theta\mathbf{u}$ representation is obtained in a unique manner from the coefficients $r_{ij\ (i=1\cdots3,j=1\cdots3)}$ of a rotation matrix $\mathbf{R}$ using the following equation [KHAL 02]:

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

where $\theta = \arccos((r_{11} + r_{22} + r_{33} - 1)/2)$ and where the sine cardinal $\mathrm{sinc}\theta$, defined by $\sin\theta = \theta\,\mathrm{sinc}\theta$, 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 an eye-in-hand system, it is possible to use 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 an eye-to-hand system, 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 mounted on the effector, or 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 related to $\theta\mathbf{u}$ is given by [MAL 99]:

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

with:

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

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

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

which guarantees the following, rather convenient property:

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

If it would be preferable to consider 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 wise to directly take into account the vector $\theta\mathbf{u}$ associated with the rotation ${}^c\mathbf{R}_o$ and to use 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 related to a 3-D point*

Using the fundamental kinematics equation given in [6.14], we immediately get for any point of the object 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 an eye-in-hand system, if we are interested in the displacement 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 rigidly fixed frame, such as R_o, or even R_{c^*} or $R_\emptyset$ if the object is static (see Figure 6.7) [WIL 96].

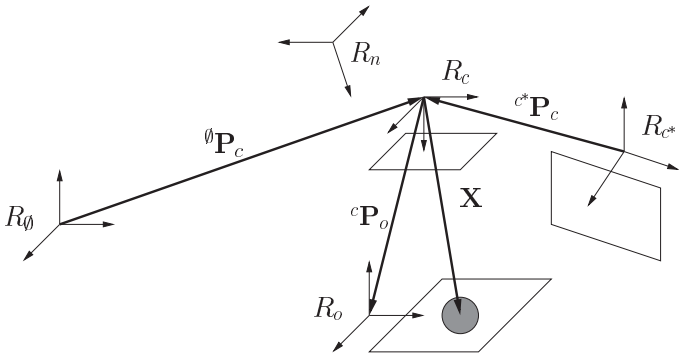

Figure 6.7. *Possible 3-D points with an eye-in-hand system*

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]:

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

We therefore have:

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

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

$$\mathbf{L}_{c^*\mathbf{t}_c} = \begin{bmatrix} {}^{c^*}\mathbf{R}_c & \mathbf{0}_3 \end{bmatrix} \quad [6.54]$$

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

With an eye-to-hand system (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 screw in this origin. This is because if we choose for example ${}^c\mathbf{t}_o$, then, using [6.51] and [6.6], we have:

$$\mathbf{L}_{c\mathbf{t}_o}{}^c\mathbf{V}_o = \begin{bmatrix} -\mathbb{I}_3 & [{}^c\mathbf{t}_o]_\times \end{bmatrix} \begin{bmatrix} {}^c\mathbf{R}_o & [{}^c\mathbf{t}_o]_\times {}^c\mathbf{R}_o \\ \mathbf{0}_3 & {}^c\mathbf{R}_o \end{bmatrix} \quad [6.55]$$

hence:

$$\mathbf{L}_{c\mathbf{t}_o}{}^c\mathbf{V}_o = \begin{bmatrix} -{}^c\mathbf{R}_o & \mathbf{0}_3 \end{bmatrix} \quad [6.56]$$

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

$${}^\emptyset\dot{\mathbf{t}}_o = \begin{bmatrix} \mathbb{I}_3 & \mathbf{0}_3 \end{bmatrix} {}^\emptyset\mathbf{v}_o \quad [6.57]$$

where ${}^\emptyset\mathbf{v}_o$ is the object's kinematic screw 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 related to a 3-D plane*

Finally, we can also determine the interaction matrix related to 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:

$$\mathbf{L}_{(\mathbf{u},D)} = \begin{bmatrix} \mathbf{0}_3 & [\mathbf{u}]_\times \\ \mathbf{u}^\top & \mathbf{0} \end{bmatrix} \quad [6.58]$$

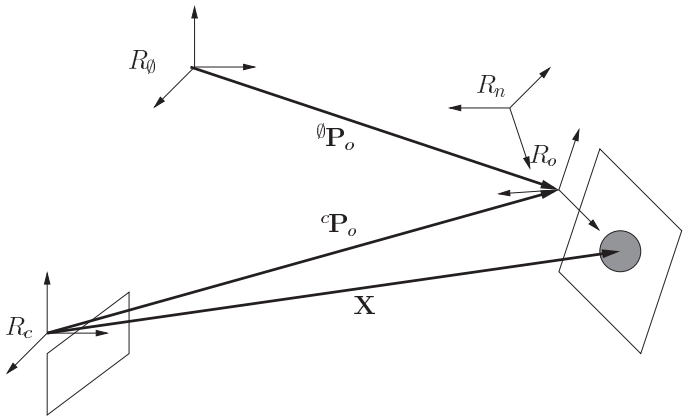

Figure 6.8. *Possible 3-D points with an eye-to-hand system*

6.3. Task function and control scheme

Achieving a robotic task by visual servoing requires the selection of the appropriate visual features and the design of a closed-loop control law. 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 consider the case where we wish to control the 6 degrees of freedom of the robot, in other words to bring the end-effector's frame to a unique desired pose.

If we use a set of k visual features $\mathbf{s}$, 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 features;

– $\mathbf{s}^*$ 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 robot's 6 degrees of freedom.

6.3.1. *Obtaining the desired value* $\mathbf{s}^*$

Whatever the nature of the visual features that were chosen, the value $\mathbf{s}^*$ 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 features, their desired value can easily be obtained if a 3-D 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 end-effector and the object of interest (for a grasping task for example): the calculation of the visual features (2-D or 3-D) is then immediately obtained if the transformation matrix between the end-effector frame and the camera frame 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 end-effector-camera transform 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 $\mathbf{s}^*$ 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 the 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 modeling errors, we find ourselves in the paradoxical situation of having biased desired and current values of visual features, 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 end-effector and the object of interest 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 eye-to-hand systems, it poses significant problems regarding where the camera is placed in the case of eye-in-hand systems.

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

6.3.2. *Regulating the task function*

As we saw in the beginning 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], LQ or LQG optimal control [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 feedback [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:

$$\dot{\mathbf{e}} = \mathbf{C}\,\dot{\mathbf{s}} = \mathbf{C}\,\mathbf{L_s}\,\mathbf{v} \qquad [6.61]$$

We saw in sections 6.2.2 and 6.2.3 how to pass from the kinematic screw $\mathbf{v}$ 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 an eye-in-hand system and $\mathbf{v_q} = -\mathbf{v_o}$ in the case of an eye-to-hand system (hence we will not be considering the problems caused by singularities of the robot and its joint limits. 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:

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

where $\frac{\partial \mathbf{e}}{\partial t}$ represents the variations of $\mathbf{e}$ caused either by the object's motion (if an eye-in-hand system is used), or by the camera's motion (if an eye-to-hand system is used). To control the robot's 6 degrees of freedom, it is at least necessary to select $\mathbf{s}$ such that $\mathbf{L_s}$ has rank 6 and we obtain as an ideal control law:

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

In the case where the visual features are expressed in the image, we saw that the interaction matrix depends on the values of these visual features and on the depth between the camera and the object in question. In the case of 3-D visual features, 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:

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

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

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

If we assume that $\frac{\partial \mathbf{e}}{\partial t} = \frac{\widehat{\partial \mathbf{e}}}{\partial t} = 0$, 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 $\frac{\widehat{\partial \mathbf{e}}}{\partial t} = \frac{\partial \mathbf{e}}{\partial t}$. 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 tracking 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 $\frac{\partial \mathbf{e}}{\partial t} = \frac{\widehat{\partial \mathbf{e}}}{\partial t} = 0$ 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, all components of $\mathbf{s}$ will have 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 sufficient 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 $\dot{\mathbf{s}} = -\lambda\, (\mathbf{s} - \mathbf{s}^*)$.