

Visual Servoing Based on a Task Function Approach

Patrick Rives*, François Chaumette**, Bernard Espiau***

* *INRIA-centre de Sophia Antipolis, 2004 Route des Lucioles,
06565 Valbonne, France*

** *IRISA-INRIA campus de Beaulieu 35042 Rennes-cedex, France*

*** *ISIA-ENSMP, rue Claude Daunesse, Sophia Antipolis,
06565 Valbonne, France*

Abstract

Recent advances in vision sensors technology and image processing authorize to hope that the use of vision data directly into the control loop of a robot is no more an utopic way. Commonly, the general approach in robot vision is the following: processing vision data into the frame linked to the sensor, converting data into the frame linked to the scene by mean of inverse calibration matrix, computing, with respect to the robot task, the control vector of the robot into the frame linked to the scene, controlling the robot by using the inverse kinematic model. This scheme works in open loop with respect to vision data and cannot take into account inaccuracies and uncertainties occuring during the processing. Such an approach needs to perfectly overcome the constraints of the problem: geometry of the sensor (for example, in a stereovision method), the model of the environment and the model of the robot. In some cases, this approach is the only one possible but, in many cases, an alternative way consists to specify the robot task in terms of control directly into the sensor frame. This approach is often referred as visual servoing [8], [3] or sensor based control [2]; in this case, a closed loop can be really performed from the vision data and allows to compensate for the perturbations by a robust control scheme. The work described in this paper deals with such an approach using a mobile vision sensor mounted on the end effector of a robot manipulator. It is characterized by two main points: the use of vision sensor as *local* sensor providing relatively poor instantaneous information but at a rate consistent with the bandwidth of the robot controller and the exploitation of the vision data into a robust control scheme based on a task function approach [7].

1 The Framework of Sensor-based Control

1.1 Background

We are interested in this paper in the design of control systems which work in closed loop with regard to the environment, with the point of view of automatic control. This means that sensors exclusively devoted to supplying of symbolic information will not be considered, and that only high data rate sensors are used. This excludes, for example,

high-level visual functions while this includes force, proximity, local range, and local visual sensing. In all cases, we shall therefore consider a sensor as a device rigidly linked to a mobile body which provides continuously with a p -dimensional vector of data denoted s . These sensor data are assumed to be only dependent of the *interactions* of the sensor and the environment.

Before designing the related control loops, modelling and analysis of these interactions are necessary steps, which will be done in the following subsection. The next one will briefly show how to realize a sensor-based task and will exhibit the needed related properties. All extended developments concerning the described approach may be found in [7].

1.2 Modelling the Interactions Sensor/ Environment

Basic Notations. A sensor (S) is linked to a rigid body (B), with related frame F_S . (S) interacts with an environment (objects) to which is associated a frame F_T . A fixed reference frame F_0 is also given. F_S and F_T are mobile with respect to F_0 . The position (location and attitude) of (B) is an element \bar{r} of the Lie group of the displacements, called SE_3 , which is a 6-dimensional differential manifold. Its tangent space is called se_3 and its dual se_3^* . A screw, H , is an element of se_3 , which is also defined by its vector and the value of its field in some point O . Shortly, $H = (H(O), u)$. When expressed in a given frame, H will be represented by a vector in \mathbf{R}^6 .

The velocity screw of a frame F_k with respect to a frame F_i is denoted T_{ki} .

The Concept of Elementary Signal. Let us consider a one-dimensional component s_j of s . Our basic assumption is that s_j is in a mathematical sense a mapping from SE_3 to \mathbf{R} ; more precisely, we assume that, given (S) and an observed environment (T), s_j is a function of the relative position of (S) with respect to (T) only. Therefore:

$$s_j = s_j(F_s, F_T) \in \mathbf{R} \quad (1)$$

This function will be assumed to be twice differentiable. Equation (1) defines a so-called *elementary signal*. This is for example the output of an ultrasonic sensor, an optical reflectance sensor, a strain gauge, or in the present case any 'relevant' parameter computed from an image.

Now, if (S) belongs to a mechanical system with several degrees of freedom (rigid manipulator, mobile robot), it may occur that the joint coordinates q constitutes a local chart of SE_3 . In another way, the sensed object (T) may move independently with respect to F_0 ; its motion is then parametrizable by the time variable t . From these two arguments, (1) may also be written:

$$s_j = s_j(q, t) \quad (2)$$

Note that when q is not a chart of SE_3 (case of a plane mobile robot for example), the system evolves inside a submanifold with dimension smaller than 6. Nevertheless, the presented approach may often be applied in this smaller configuration space.

The Concept of Interaction Screw. Owing to its definition s_j admits a derivative represented by an element of se_3 . The differential of s_j is a mapping from se_3^* to \mathbf{R} which may shortly be written:

$$\dot{s}_j = \frac{\partial s_j}{\partial \bar{r}} \bullet \frac{d\bar{r}}{dt} = H_j \bullet T_{ST} \quad (3)$$

where \bullet denotes the screw product. Note that H and T_{ST} are both functions of q and t . Let us now suppose that (S) belongs to the last body, (B_6) , of a six-jointed robot. Equation (3) may then be expressed, for example in F_0 , giving:

$$\dot{s}_j = [u_j^T H_j^T(O_6)](q, t) J_6(q) \dot{q} + [u_j^T H_j^T(T)](q, t) \begin{pmatrix} V_T(t) \\ \Omega_{F_T}(t) \end{pmatrix} \quad (4)$$

where J_6 is the Jacobian matrix associated with the frame F_6 linked to (B_6) , with origin O_6 , and T a fixed point of (T) , all elements being expressed in F_0 .

From now, we assume without loss of generality that (T) is motionless in F_0 ; we then finally define the *interaction screw* between the elementary sensor with output s_j and the environment as the screw H_j such that:

$$\dot{s}_j = H_j \bullet T_{SO} \quad (5)$$

H_j may be understood as a kind of Jacobian of the elementary sensor relative to its displacements with regard to the environment. It should be emphasized that all the information about the interactions between a sensor and its environment is contained in H_j . Unfortunately, the knowledge of H_j is generally partial, because of the existing uncertainties on environment and sensors. Consequently, only models of H_j may be used in practice; such models will be denoted \hat{H}_j in the following.

The Concept of Virtual Linkage. Let us now search for a velocity screw T^* such that $\dot{s}_j = 0$, i.e. *the elementary sensor output is invariant with respect to the motion defined by T^** . T^* is a solution of:

$$H_j \bullet T^* = 0 \quad (6)$$

i.e. is a screw reciprocal to H_j . Considering now the full sensor (S) with *vector* output s shows that the motions T^* leaving s unchanged belong to a subspace reciprocal to the subspace \mathcal{S} spanned by the set $H_1, \dots, H_j, \dots, H_p$.

The set $(S); H_1, \dots, H_j, \dots, H_p$ constitutes a *virtual linkage*.

This concept is an extension of the classical one used in the description of mechanical contacts between rigid bodies. Its *class, N* (number of "allowed motions"), is the dimension of \mathcal{S} . When a frame and a basis are chosen, the virtual linkage is fully defined by the properties of the $6 \times p$ matrix

$$L = \begin{pmatrix} u_1 & \dots & u_p \\ H_1(P) & \dots & H_p(P) \end{pmatrix} \quad (7)$$

with rank $6 - N$, evaluated at some point P . Recall that $H_j(P) = H_j(P') + \Delta_{PP'}u_j$, where $\Delta_{PP'}$ is the antisymmetric matrix associated with the cross product $P\bar{P}'\times$.

The concept of virtual linkage is an easy way to the user to *specify* the task he wishes to realize through the sensors. It allows to determine the motions which may be sensor-controlled and the ones which remain free. In the case of physical contact, and with some assumptions on the used force sensors, this virtual linkage reduces to an *actual* linkage. An important fact to be emphasized is that a *virtual stiffness* may also be associated with the linkage, through the performed control. All these points are discussed in [7].

1.3 Sensor-based Task Functions

General Background. Let us consider a rigid robot, the dynamic equation of which is

$$\Gamma = M(q)\ddot{q} + N(q, \dot{q}, t); \dim(q) = n \quad (8)$$

where M is the kinetic energy matrix and N gathers Coriolis, centrifugal, gravity and friction terms. From the point of view of automatic control, (8) is the *state* equation of the system (with natural state vector (q, \dot{q})), which fully describes its dynamics and involves only its intrinsic properties. The user's requirements, i.e. the task to be performed, have only to be specified as an *output function* associated with equation (8). The problem will be considered as well-posed if the 'passage' from the output space to the space where the control is actually performed is regular in some sense.

Further considerations ([2], [6]) allows finally to state that, in robotics applications, most of the user's objectives may be defined through a C^2 n -dimensional function depending both of q and t , $e(q, t)$, to be servoed to zero, associated with an initial condition q_0 and a time horizon $[0, T]$. Provided that a solution exist, some other conditions among them the regularity of the task-Jacobian $\frac{\partial e}{\partial q}$ in a certain domain, are also required.

A Major Property Needed for Stability. Given equation (8), and a task function $e(q, t)$ to be servoed, a control scheme has to be designed. A general form is the approximated decoupling/linearization scheme performed on a transformation of model (8) in the task space ([6]). "Approximated" means that *models* \hat{M} , \hat{N} , $\frac{\partial \hat{e}}{\partial q}$, $\frac{\partial \hat{e}}{\partial t}$, are used instead of the true expressions often practically unavailable. In [6] and [7], C. Samson has exhibited some sufficient conditions ensuring the closed-loop system to be stable. One of the most important is certainly:

$$\frac{\partial e}{\partial q} \left(\frac{\partial \hat{e}}{\partial q} \right)^{-1} > 0 \quad (9)$$

in the sense that a $n \times n$ matrix A is positive if $x^T A x > 0$ for any nonzero $x \in \mathbf{R}^n$.

Clearly, this condition involves the task itself. More precisely, it states the minimal knowledge which is needed about the task for ensuring a good behaviour of the system. This property will be largely used in the following.

Case of Sensor-based Tasks. A common example of task function is the tracking of a trajectory in a given space, for instance $e(q, t) = \begin{pmatrix} x(q) - x_r(t) \\ d(r(q), r_d(t)) \end{pmatrix}$ where x is the location of O_6 , x_r the related trajectory to be tracked, and $d(\cdot)$ a 3-dimensional vector representing the error between a desired rotation $r_d(t)$ and the actual one $r(q)$. The use of sensors is also a way of constituting a task function, when the needed assumptions are satisfied. However, due to the fact that the virtual linkage which is desired to be realized is often of class N greater than 0, the related sensor based task may have an intrinsic dimension less than the required n . It is therefore necessary to complete it, provided that the 'added' objective is *independent and compatible* with the sensor-based task. It may be shown ([7],[2]) that an efficient way for doing this is the following: starting from s , a first step is to build a vector function $e_1(s)$, generally linear ($e_1 = Ds - e^*$) with the right dimension, i.e. the one of S , and characterizing in a simple manner the desired virtual linkage (the example of visual-based tasks will be presented later). The second step consists in defining a secondary objective, to be realized under the constraint of achieving $e_1 = 0$. This secondary goal is expressed as the minimization of a cost function h_s . It may then be shown that the resulting (global) task function takes the general form:

$$e = W^+ W e_1 + \alpha (I - W^+ W) \frac{\partial h_s}{\partial x} \quad (10)$$

where x is an element of any working space such that $\frac{\partial x}{\partial q}$ is nonsingular, α is a positive scalar function and W a matrix function such that, for decoupling requirements, $\text{range}(W^T) = \text{range}(\frac{\partial e_1}{\partial x})^T$.

A particularly interesting case, used in all the following, is $x = \bar{r}$, with the frame of the sensor system used as working frame. Then, $\frac{\partial e_1}{\partial r}$ is directly related to the matrix L of the interaction screws; furthermore, it is sometimes possible to choose W such that $W^+ W$ is diagonal, with entries 0 or 1, then called a selection matrix. In that case, and if $\frac{\partial h_s}{\partial r}$ has the meaning of a trajectory tracking in SE_3 , the realized task is called a *hybrid task*, by analogy with the wellknown *hybrid control*.

A last thing is that the important positivity condition (9), when applied to equation (10), may be sometimes easily satisfied by a simple choice of the model $\frac{\partial \hat{e}}{\partial q}$: it is shown in [7] that, when W may be chosen such that $\frac{\partial e}{\partial r} W^T > 0$, and if α is not too large, then $\frac{\partial \hat{e}}{\partial r}$ is *itself* positive. Condition (9) is therefore satisfied with the simple choice $\frac{\partial \hat{e}}{\partial q} = \frac{\partial r}{\partial q}$, i.e. the basic Jacobian matrix of the robot. This is the reason why some hybrid control schemes may work even with a bad knowledge of the true interactions.

In the case of visual sensors, it is very difficult to know precisely the interaction screws, because they will be shown later to depend on an unknown parameter, the depth. In opposition to force or range measurements, it is thus not possible to find a priori what are the right *models* to be used, and, more, what is the allowed "amount of uncertainty". From an other point of view, given certain simple choices of models, it is not easy to determine their domain of validity. This is why, in the case of visual servoing, often only an experimental analysis may allow to validate the choices done in the control schemes. This way of applying the previous theory is the topic of the following developments.

2 Application to a visual servoing approach

2.1 Problem statement

Let us consider a vision sensor moving across a three dimensional environment (3D-scene). We assume that the motion of the sensor is fully controllable and can be characterized by its velocity screw. In practice, this vision sensor can be mounted on the end effector of a manipulator or carried by a mobile robot. We use the classical "pinhole" approximation for modelling the perspective geometry of the sensor, and we assume the focal length equal to unity (figure 1). Using the same definition as above for the diverse frames, we can state that, due to the motion of sensor and objects, a point $P_i = (x_i; y_i; z_i)^T$ linked to an object , moves with a relative velocity with regard to F_S . This velocity can be expressed by the velocity screw T_{ST} by means of:

$$\dot{P}_i = V_T + \Omega_{F_T} \times SP_i \quad (11)$$

At each instant, the point P_i projects onto the image plane as a point p_i with coordinates $(X_i; Y_i)^T$

$$X_i = \frac{x_i}{z_i} ; Y_i = \frac{y_i}{z_i} \quad (12)$$

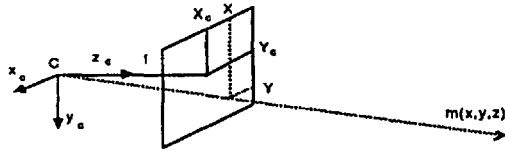


Figure 1: Perspective model

A velocity field projected onto the image plane corresponds to the motions of the sensor and/or objects in the 3D-scene ; this velocity field is often called as the Optical Flow field. By differentiating (12) and using (11) , we can derive the wellknown equation relating optical flow measurement to 3D structure and motion in the scene. When expressed in the F_S frame, we obtain:

$$\begin{pmatrix} \dot{X}_i \\ \dot{Y}_i \end{pmatrix} = \begin{pmatrix} \frac{1}{z_i} & 0 & \frac{X_i}{z_i} & X_i Y_i & -(1 + X_i^2) & Y_i \\ 0 & \frac{1}{z_i} & \frac{Y_i}{z_i} & 1 + Y_i^2 & X_i Y_i & X_i \end{pmatrix} \cdot \begin{pmatrix} V_T \\ \Omega_{F_T} \end{pmatrix} \quad (13)$$

This equation shows the basic structure of the *interaction* between the vision sensor and its environment. As we underlined in the first part, the two interaction screws appearing as the lines of the matrix given in (13) depend on the inverse of the depth z_i expressed in the F_S frame. Generally, without some *a priori* knowledge about the environment, it is not possible to know the true value of the interaction screws and only some rough approximation will be available for the control purpose. As we shall show later, we will be able to compensate for this lack of accuracy by a robust control scheme.

Now, let us consider more complex geometrical primitives than points. In a general way, we assume that a 3D geometrical primitive can be represented as a vectorial function $h(x(t), y(t), z(t), Q(t)) = 0$ which projects onto the image plane under the form $g(X(t), Y(t), R(t)) = 0$ where $Q(t) = Q_i(t); i = 1, m$ and $R(t) = R_i(t); i = 1, n$ are the parameters of the primitives respectively in the 3D scene and in the image plane.

From these assumptions, an important part of the visual servoing problem will be devoted to find, for a given 3D primitive, a wellsuited parametrization of h and g (i.e. without singularity of representation) and to establish the interaction screw $H_i(R, Q)$ such that:

$$\dot{R}_i = H_i(R, Q) \bullet T_{ST} \quad (14)$$

According to virtual linkage defined in section 2.2, we can search for a velocity screw T^* such that $\dot{R}_i = 0$ leaving R unchanged which can be formulated by:

$$(\forall T^* \in \mathcal{S}^* \subset \mathcal{S}) \text{ if } T^* \in \text{Ker}(H_i) \iff \dot{R}_i = 0 \quad (15)$$

The following section is devoted to the definition a set of *elementary visual signals* and their associated virtual linkage from a set of low level geometrical primitives.

2.2 Modelling elementary visual signals

Assuming that g is a C^2 function on a time interval $[0, T]$, let us derive a general solution for computing $H_i(R, Q)$:

$$g(X(t), Y(t), R(t)) = 0; \forall t \in [0, T] \Rightarrow \dot{g}(X(t), Y(t), R(t)) = 0; \forall t \in [0, T] \quad (16)$$

after developments, we obtain:

$$\sum_{i=1}^n \frac{\partial g}{\partial R_i} \dot{R}_i = -\frac{\partial g}{\partial X} \dot{X} - \frac{\partial g}{\partial Y} \dot{Y}; \forall (X, Y) \text{ such that } g(X, Y, R) = 0 \quad (17)$$

Eq.(16) allows us to relate the variation of the parameters R_i characterizing the 2D primitive in the image plane to the optic flow components and thus, to the velocity screw of the camera by means of eq.(13). An *elementary visual signal* will be defined as a function $s_j = f(R_i, \dots, R_k)$, $i, \dots, k \in n$ characterizing usual geometrical properties in the image like distance between two points, orientation between two lines, surface, mass centroid and so on.

Case of points

Using one point. The equation of h and g are trivial, and we find again the classic optical flow equation:

$$h : \begin{cases} x - x_1 = 0 \\ y - y_1 = 0 \\ z - z_1 = 0 \end{cases} \longrightarrow g : \begin{cases} X - X_1 = 0 \\ Y - Y_1 = 0 \end{cases} \quad (18)$$

From the eq.(17), we have:

$$\dot{j} = 0 \Rightarrow \begin{cases} \dot{X}_1 = [\frac{1}{z_1} & 0 & \frac{X_1}{z_1} & X_1 Y_1 & -(1 + X_1^2) & Y_1] \cdot T_{ST} \\ \dot{Y}_1 = [0 & \frac{1}{z_1} & \frac{Y_1}{z_1} & 1 + Y_1^2 & X_1 Y_1 & X_1] \cdot T_{ST} \end{cases} \quad (19)$$

Finally, we can define two elementary visual signal $s_1 = X_1$ and $s_2 = Y_1$; the set ($S = [s_1 s_2]^T$); H_1, H_2 constitutes a virtual linkage the class of which is $4 = \dim(S)$ (S is the subspace spanned by H_1, H_2). A basis of S can be easily found, for example:

$$B : \begin{pmatrix} X_1 & 0 & z_1(1 + X_1^2 + Y_1^2) & 0 \\ Y_1 & 0 & 0 & z_1(1 + X_1^2 + Y_1^2) \\ 1 & 0 & 0 & 0 \\ 0 & X_1 & -X_1 Y_1 & 1 + X_1^2 \\ 0 & Y_1 & -(1 + Y_1^2) & X_1 Y_1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (20)$$

Each velocity screw T_{ST}^* which belongs to S leaves unchanged the projection of the 3D point in the image plane.

Using several points. The case of two different points in the image frame leads to the following equation:

$$\begin{pmatrix} \dot{s}_1 \\ \dot{s}_2 \\ \dot{s}_3 \\ \dot{s}_4 \end{pmatrix} = \begin{pmatrix} \dot{X}_1 \\ \dot{Y}_1 \\ \dot{X}_2 \\ \dot{Y}_2 \end{pmatrix} = H \cdot T_{ST} \quad (21)$$

the class of the linkage is 2, i.e. the dimension of the subspace corresponding to the allowed motions without changing the nature of the image.

The case of three points is more interesting. As it is wellknown, the inverse perspective problem (recovering location and orientation of 3D objects from one 2D image) has not an unique solution. By this fact, several relative attitudes between the camera and the object can lead to the same image. These attitudes correspond to the solving of a 8-degrees analytic equation. In the most cases, a continuous motion does not exist from one solution to another one which leaves unchanged the image. In consequence, for a particular attitude between object and camera, the dimension of the subspace S is 0 (i.e. $\text{rank}(H) = 6$). However, for some particular cases, like, for example, when the three points are colinear in the 3D space, singularities may be occurred (i.e. $\text{rank}(H) < 6$). For a number of points greater than three (four or more), we are in a case of redundancy: the rank of H is always equal to 6 and the dimension of T_{ST}^* equal to 0. We will see later how we can use these properties to perform a task of target tracking.

Other elementary visual signal based on points In the previous paragraph, we only presented the case where the application $s_j = f(R_i, \dots, R_k)$, $i, \dots, k \in n$ is the identity. In many cases, it can be fruitful to consider more relevant signals with regard to the task. For instance, let us consider the problem of positioning an end effector with

regard to an object which is characterized by three points. A natural way for specifying this task is to control, on the first hand, the location of a particular point of the object, by example the mass centroid of the three points and on the other hand some geometrical characteristics like the distance between the three points. For doing that, we choose the following signals vector:

$$s = \begin{pmatrix} L_{12} \\ L_{13} \\ L_{23} \\ X_G \\ Y_G \end{pmatrix} \quad (22)$$

with

$$s_i = L_{jk} = (X_j - X_k)^2 + (Y_j - Y_k)^2 ; i = 1 \dots 3; j = 1 \dots 3, k = j + 1 \dots 3 \quad (23)$$

and

$$s_4 = X_G = \frac{\sum_{i=1}^3 X_i}{3} ; s_5 = Y_G = \frac{\sum_{i=1}^3 Y_i}{3} \quad (24)$$

Then it is possible to compute the interaction screw $H(X_1, X_2, X_3, Y_1, Y_2, Y_3, z_1, z_2, z_3)$ and the associated virtual linkage. Let us remark that this choice of s lets a rotation around the mass centroid to be free (ie $rankH = 5$); The full control of this rotation from the image needs another component characterizing the orientation in the image plane like, for instance, the orientation of the main inertial axis with regard to the image frame has to be add to s .

Case of lines. For many reasons (accuracy, robustness with regard to the noise...) it is often interesting to use in image analysis more structured primitives than simple points. In the case of visual servoing, using lines as primitives seems to be natural. A 3D line will be represented by two planes which intersect:

$$h(x, y, z, Q) : \begin{cases} a_1x + b_1y + c_1z + d_1 = 0 \\ a_2x + b_2y + c_2z + d_2 = 0 \end{cases}, \text{ with } d_1, d_2 \neq 0 \quad (25)$$

A 3D line in the scene projects onto the image plane as a 2D line (except on some degenerate cases). Some attention has to be taken concerning the parametrization of the 2D line. As we underline in the first part, an elementary visual signal has to be twice differentiable. Let us consider the classical representation of a 2D line: $Y = aX + b$. This parametrization needs two charts for representing 2D lines into the cartesian space (i.e. the lines: $X = 1$ and $Y = 1$ belongs to two different charts). If we use the parameters a and b as elementary visual signals, the condition of differentiability is not preserved during the passage from one chart to the other. For this reason, we choose a representation ρ, θ and we obtain:

$$D : g(X, Y, R) = X \cos \theta + Y \sin \theta - \rho = 0, \theta \in] -\frac{\pi}{2}, \frac{\pi}{2}] \quad (26)$$

deriving this equation,

$$\dot{g} = 0 \Rightarrow \dot{\rho} + (X \sin \theta - Y \cos \theta) \dot{\theta} = \cos \theta \dot{X} + \sin \theta \dot{Y}, \forall (X, Y) \in \mathcal{D} \quad (27)$$

finally, using eqs.(13), (25), (26), we obtain:

$$\dot{\theta} = -\left(\frac{b_1}{d_1} \cos \theta - \frac{a_1}{d_1} \sin \theta\right)(\cos \theta \sin \theta - \rho) \cdot (V_T) - (\rho \cos \theta \sin \theta - 1) \cdot (\Omega_{F_T}) \quad (28)$$

and

$$\dot{\rho} = \left(\frac{c_1}{d_1} + \rho \frac{a_1}{d_1} \cos \theta + \rho \frac{b_1}{d_1} \sin \theta\right)(\cos \theta \sin \theta - \rho) \cdot (V_T) + ((1 + \rho^2) \sin \theta - (1 + \rho^2) \cos \theta) \cdot (\Omega_{F_T}) \quad (29)$$

Eqs.(28)(29) define the associated interaction screws. Typically, the vector $s = (\rho, \theta)^T$ will be used to characterize a 2D line, but if necessary, we will be able to control some other characteristics like, for instance, orientation between two lines ($s = |\theta_1 - \theta_2|$) or all other measurement built from θ and ρ .

Case of circles. In a similar manner, let us consider a circle in the 3D scene. It projects onto the image plane as an ellipse:

$$h(x, y, z, Q) : \begin{cases} (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0 \\ (z - z_0) - \alpha(x - x_0) - \beta(y - y_0) = 0 \end{cases} \quad (30)$$

$$\Rightarrow g(X, Y, R) : \frac{(X - X_C + e(Y - Y_C))^2}{a^2(1 + e^2)} + \frac{(Y - Y_C + e(X - X_C))^2}{b^2(1 + e^2)} - 1 = 0 \quad (31)$$

After some tedious developments, we can relate the variation of the parameters of the ellipse to the motion into the 3D scene by means of the interaction screws such that:

$$\begin{pmatrix} \dot{X}_C \\ \dot{Y}_C \\ \dot{e} \\ \dot{a} \\ \dot{b} \end{pmatrix} = H \bullet T_{ST} \quad (32)$$

2.3 Visual sensing and task function

At this step, we have defined a set of elementary vision signals; in this section, we will investigate some ways of using these signals in a robust control scheme based on a task function approach, as defined in section 3.1. The problem can be stated as follows: is it possible to specify a robotic task in term of reaching a particular configuration of a set of features in the image frame? If so, from a running observation of these features in the image, are we able to perform this task? This means that we have to design a control scheme which allows us to reach this particular configuration (target image). Let

us consider an example: a task for a mobile robot consists in going in front of a door at a given distance. This task may be translated in the image frame by the following characteristics:

- matching in the image a set of polygonal lines which can be associated with the model of the door.
- controlling the motion of the robot for bringing these lines to form a rectangle centred on the image frame with two parallel and vertical lines and the surface of which is equal to a given value.

With regard to the general form of sensor based task functions presented in the first part, we define a task vector $e(q(t), t)$, associated to a visual task function, where q is a local chart of SE_3 , (obviously, in the case of a camera mounted on an end effector of a manipulator q is a chart of the configuration space) and such that :

$$e(q(t), t) = \begin{pmatrix} s_1(q, t) - s_1^*(t) \\ \vdots \\ s_p(q, t) - s_p^*(t) \end{pmatrix} \quad (33)$$

where $s^*(t)$ can be considered as a reference trajectory to be tracked in the image frame. If the robotic task just consists in controlling a given attitude of the sensor with respect to a motionless target, (i.e. *positioning problem*) then s^* will be time independant.

Considering the control problem as an output regulation problem, we can assume that the concerned task is perfectly achieved during $[0, T]$ if $e(q(t), t) = 0$ for all $t \in [0, T]$. For simplifying the derivation of the control, we shall consider that the motions of the camera and the robot are slow enough to allow the use of a full dynamic model (8) to be avoided. With intend to simplify our purpose, we only consider the case where the matrix $H(R, Q)$ is regular but this approach can be extended to any cases without loss of generality. We therefore focus on the robustness with respect to uncertainties on the interactions by using a gradient-based approach as a particular case of the general approach evoked in section 2.3. In this approach, we assume that a linearly decreasing function $T_d = f(e(q(t), t))$ relates the desired control vector to the task function with a limit condition $\lim_{e \rightarrow 0} f = 0$. Under these assumptions, we may choose:

$$T_d = -\mu C \cdot e(q(t), t) \quad (34)$$

where $\mu > 0$ and C is a constant positive matrix. From eqs.(3) and (34), we obtain:

$$\dot{e} = \dot{s} = H \bullet T_d \Rightarrow \dot{e} = -\mu H \cdot C e \quad (35)$$

An exponential convergency will be ensured under the following sufficient condition:

$$HC > 0 \quad (36)$$

A good and simple way to try to ensure this matrix positivity is to enforce $H(R^*, Q^*) \cdot C = I$ (where I represents the identity matrix) for the equilibrium position $s = s^*$. In this case, the control matrix C takes the following form:

$$C = H(R^*, Q^*)^{-1} \quad (37)$$

Let us note that the computation of C depends of the parameters R referring to the image which are known or easily measured and of the parameters Q referring to the 3D scene. Without an a priori knowledge on the 3D scene, these last parameters cannot be inferred from the image and, by this fact, the matrix C cannot be exactly computed. Fortunately, in many cases, the interaction screw H can be factorized in a product of two matrices such that:

$$H(R^*, Q^*) = B(R^*) \cdot D(R^*, Q^*) \quad (38)$$

where $D(R^*, Q^*)$ is a positive diagonal matrix. For instance, when the target is constituted by three 3D points and when the equilibrium position is such that the target is parallel to the image plane, the matrix D has the following form:

$$D(R^*, Q^*) = \left(\begin{array}{c|c} \frac{I_3}{z^*} & 0 \\ \hline 0 & I_3 \end{array} \right) \text{ with } z^* > 0 \quad (39)$$

In this cases, we can choose a control matrix such that $C = B(R^*)^{-1}$ since the convergency condition $\|B \cdot D \cdot B^{-1}\| > 0$ is always verified.

Obviously, the results presented above ensure the positivity only at the equilibrium position where $s = s^*$. Far from this position, the convergency property is not always insured and the positivity of $\|H(R(t), Q(t)) \cdot C(R^*, Q^*)\|$ should be checked for all $t \in]0, T]$. Fortunately, as we will see in the next section, outside the task singularities, the satisfaction of this condition is shown to be preserved in practice.

3 Results:

The approach developped in this paper has been validated both in simulation [4] and in an experimental cell constituted by a CCD camera mounted on the effector of a 6dof manipulator [1].

3.1 Simulation results

We focused our investigations on the problem of positioning the sensor with respect to a static or a moving target. We investigated the use of different types of primitives as elementary signals (points, lines, circles). For each case, we performed a first serie of experiments corresponding to the following configurations:

- static target, noise-free synthetic image, noise-free control vector.
- static target, the image noise had a Normal distribution with zero mean and a standard deviation of two pixels, the control vector noise had a Normal distribution with zero mean and a standard deviation of 1 centimeter in translation and two degrees in rotation.

In all cases, the chosen control was of form given by eqs. (34), (37). Some results are given in the figure 2 (noise-free) and 3 (with noise) in a positioning task of the camera with respect to a target constituted by four points. The coordinates of every point, X_i and Y_i are taken as elementary signals in the image frame. In these figures, the two top windows presents the relative attitude of the camera (symbolized by a pyramid) and the target viewed by an outside observer. The two bottom windows presents the target as seen by the camera. The windows on the left correspond to the initial position and the windows on the center to the final desired one. The two windows on the right show the behaviour of the error and of the control vector during the visual servoing. The top window corresponds to the mean quadratic error on the set $\epsilon(t) = \sum_{p=1}^{p=8} (s_p(t) - s_p^*)^2$ and the bottom window the behaviour of each component of the control vector (velocity screw of the camera) during the positioning task.

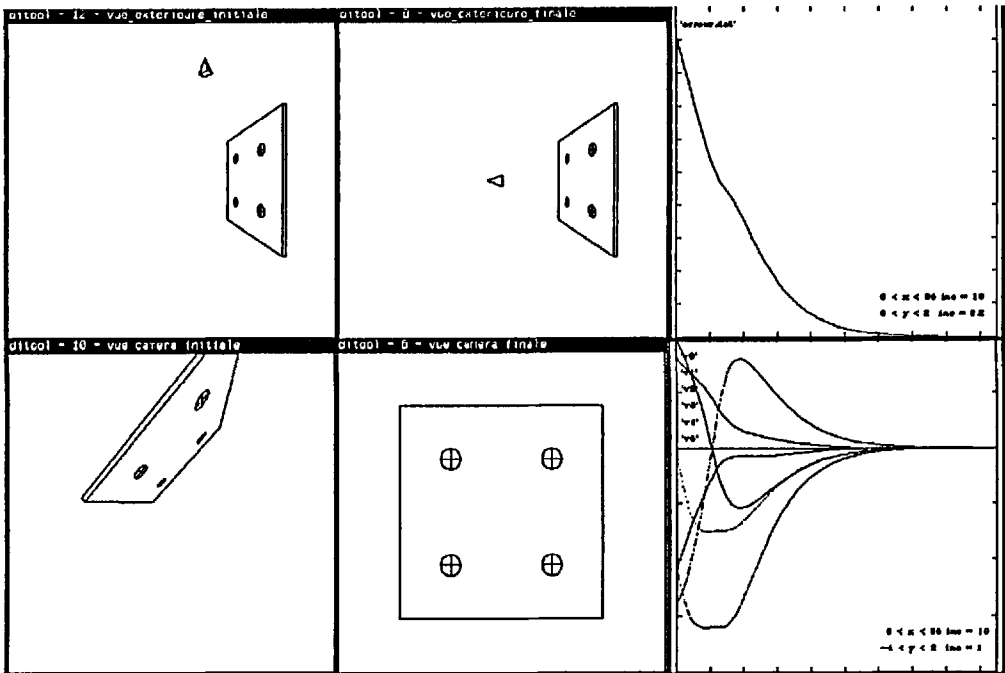


Figure 2: static target, noise-free synthetic image

As shown on the figures, the convergency to the desired configuration of the target in the image is always performed in spite of the distant initial position. Same experiments are successfully performed when the target is moving through the scene with any motion consistent with the sampling of the servocontrol loop. Analog results are obtained with lines and circles primitives.

3.2 Experimental results:

The approach has been also validated in real environments by using a CCD camera mounted on the end effector of a 6dof manipulator (figure 4). The behaviour of the

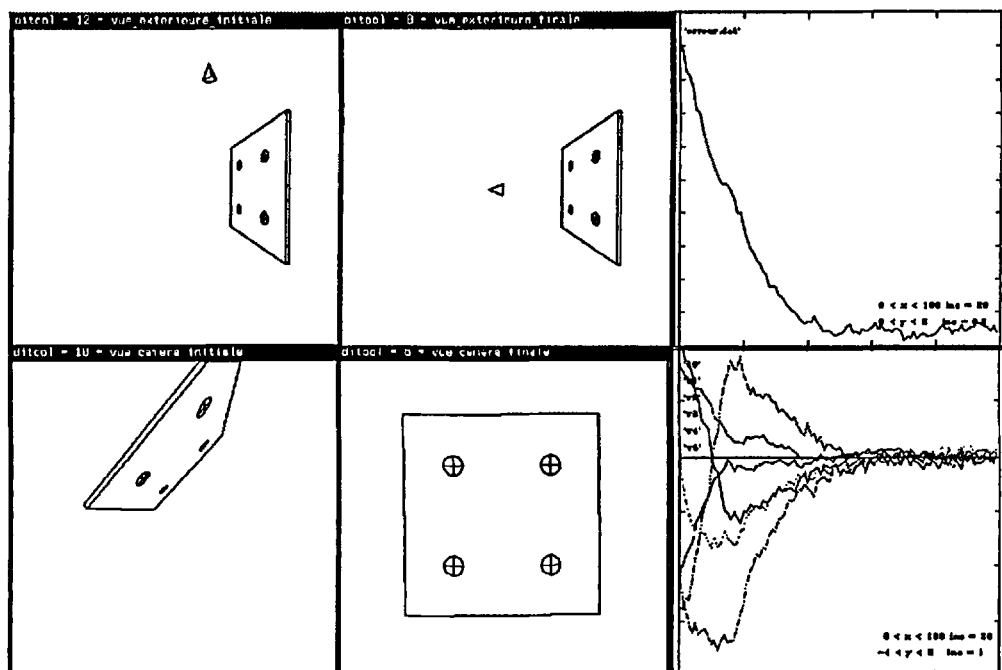


Figure 3: static target, noisy synthetic image

real system is extremely close to the one expected by the simulations. This a posteriori justifies in the studied cases the choice of a control law of the form given by eqs. (34), (37). The choice of $H(Q^*, R^*)$ (i.e. the value of H at the desired equilibrium position) as a control matrix has therefore been experimentally shown to be robust, even for rather large initial position errors.

The four points experiment described in simulation, has been successfully implemented. The figure 5 presents on the first hand, the sequence of images during the positioning task from the initial position to the final one, on the second hand, the behaviour of the mean quadratic error as in the simulation case, and the behaviour of each elementary signal error involved in the control loop.

4 Conclusion

This paper discusses the problem of using vision data directly as an input of the robot control loop. This approach, often referred as visual servoing, is characterized by the use of vision sensor as *local* sensor providing relatively poor instantaneous information but at a rate consistent with the bandwidth of the robot controller, and by the exploitation of the vision data into a robust control scheme based on a task function approach. Concerning the first point, vision data are modelled as a set of *elementary signals*. Each elementary signal is associated at a 2D geometric primitive into the image frame (point, line, circle...) corresponding to the projection of a 3D primitive in the scene frame. We show that if

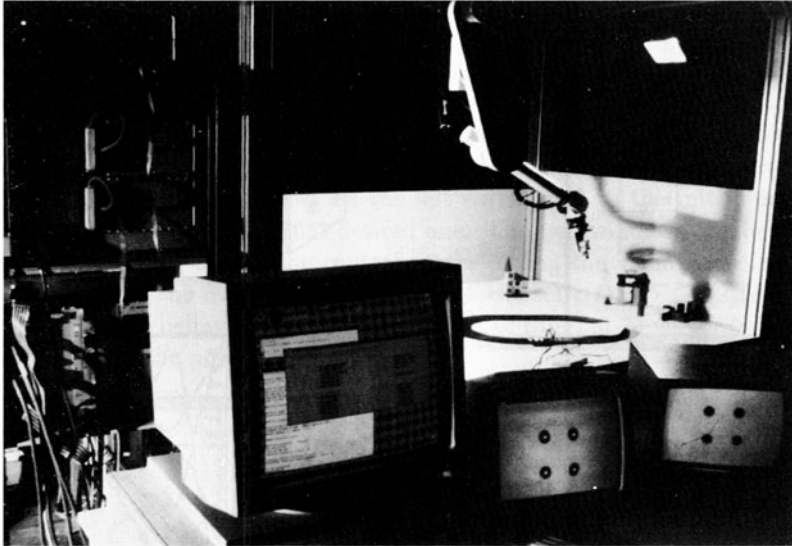


Figure 4: experimental cell

the signal is only dependant of the relative geometry between the sensor and the scene, then the interaction between the sensor and the scene can be described by a screw which relates observed motion in the image frame (optic flow) to relative motion between the camera and the objects in the 3D environment frame.

Concerning the second point, we propose a robust control scheme based on a task function approach. We show that the problem of control can be stated as a problem of regulation directly into the image frame. From the desired image target and the currently image observed by the camera, it is possible to define an error function and to express the problem of regulation as a problem of minimization of this error function. In this case, a class of control based on gradient techniques allows to perform correctly the task with good convergence properties (assuming only the definite positivity of a certain matrix).

This approach has been successfully validated both in simulation and in real experiments. In a next future, we should hope to address the problem of programming complex robotics tasks in terms of a succession of elementary subtasks which can be described by constraints between the frame linked to the effector and the frame linked to the scene (ie colinearity between axis, following of surface at a constant range...).

5 Acknowledgements

This work is supported by the French National Council of Scientific Research (CNRS) under the national project ORASIS-PRC Communication H/M.

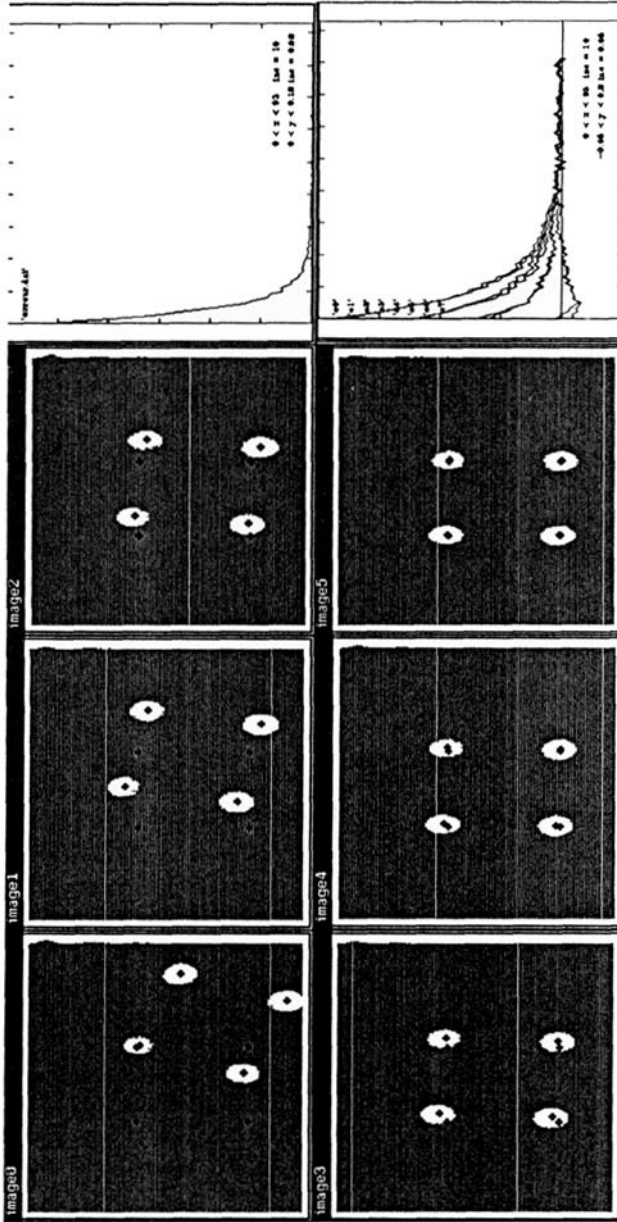


Figure 5: Real experiment

References

- [1] F. Chaumette, P. Rives, *Réalisation et calibration d'un système expérimental de vision composé d'une caméra mobile embarquée sur un robot-manipulateur* Rapport de Recherche INRIA n.994, Mars 1989
- [2] B. Espiau, *Sensory-based Control: Robustness Issues and Modelling Techniques. Application to Proximity Sensing* NATO Workshop on Kinematic and Dynamic Issues in Sensor Based Control, Italy, Oct.1987
- [3] J. T. Feddema, C. S. G. Lee and O. R. Mitchell, *Automatic selection of image features for visual servoing of a robot manipulator* Conf. IEEE Robotics and Automation, Scottsdale, Arizona, USA, May 14-19, 1989
- [4] P. Rives, G. Hegron, *Design of a simulation tool for robots using vision sensors* NATO Workshop on Languages for Sensor Based Control in Robotics, Italy, Sept. 1986
- [5] P. Rives, *Dynamic vision: theoretical capabilities and practical problems.* NATO Workshop on Kinematic and Dynamic Issues in Sensor Based Control, Italy, Oct.1987
- [6] C. Samson, *Une approche pour la synthèse et l'analyse de la commande des robots manipulateurs rigides* Rapport de Recherche INRIA n.669, Mars 1989
- [7] C. Samson, M. Leborgne, B. Espiau, *Robot control: the task function approach* to appear, Ed. Oxford University Press 1989
- [8] L. E. Weiss, *Dynamic Visual Servo Control of Robots. An Adaptive Image based Approach* Technical Report, CMU-RI-TR-84-16; Carnegie Mellon