IEEE International Workshop on Intelligent Motion Control
Boğaziçi University, Istanbul, 20-22 August 1990.
Edited by Okyay KAYNAK. IEEE Cat. No: 90TH0272-5

VISION-BASED-CONTROL FOR ROBOTIC TASKS

F. CHAUMETTE*, P. RIVES**

* IRISA-INRIA Rennes, Campus de Beaulieu, 35042 Rennes-cedex, France
** INRIA Sophia Antipolis, 2004 Route des Lucioles, 06565 Valbonne, France

Abstract : The work described in this paper deals with a closed loop achieving robotic tasks using vision data. Vision data, provided by a camera mounted on the robot, are modelled as a set of elementary signals associated to the 2D geometric primitives in the image (point, line, ellipse,...) corresponding to the projection of the 3D primitives in the scene. We show that the *interaction* between the sensor and the scene can be described by a *screw* which links the behaviour of the signal to the camera motion. For a given task, we build an *image target*, constituted by a set of elementary signals, corresponding to a good realization of the task. If we consider the desired image target and the image currently observed by the camera, the problem of control can be stated as a problem of regulation in the image frame. Velocity control based on *gradient* techniques then allows to correctly perform the task with good convergence properties.

Finally, we present some experimental results obtained both in simulation and in an experimental cell constituted by a CCD camera mounted on the effector of a 6dof manipulator.

Keywords. Velocity control; sensor-based-control; image processing; interaction screw.

INTRODUCTION

Recent advances in vision sensor technology and image processing let us believe that the use of vision data directly into the control loop of a robot is not utopic anymore. Generally, the approach in robot vision is the following: process vision data into the frame linked to the sensor, convert data into the frame linked to the scene by means of inverse calibration matrix, compute, with respect to the robot task, the control vector of the robot into the frame linked to the scene, control the robot 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 problem constraints: sensor geometry (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 in specifying the robot task in terms of control directly into the sensor frame. This approach is often referred to as Visual Servoing (Weiss, 1984; Feddema, 1989) or Sensor-Based-Control (Espiau, 1987); in this case, a closed loop can be really performed from the vision data and allows to compensate for the perturbations using 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 bandwith of the robot controller, and the exploitation of the vision data into a robust control scheme based on a task function approach (Samson, 1987).

PROBLEM STATEMENT

Let us consider a vision sensor moving across a three dimensional environment (3D-scene). We use the classical pinhole approximation to model the perspective geometry of the sensor, and we assume, without loss of generality, the focal length equal to unity (Fig. 1). At each time, a point $m = (x\ y\ z)^T$, linked to an object, projects onto the image plane as a point $M = (X\ Y)^T$ with:

$$X = x/z \ ; \ Y = y/z \tag{1}$$

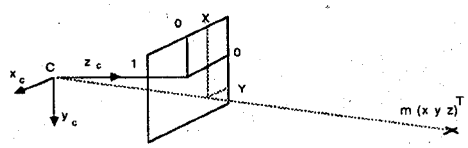

Fig. 1. Perspective model.

We assume that the motion of the sensor is fully controlable and can be characterized by its translational velocity $V_c = (V_{c_x} V_{c_y} V_{c_z})^T$ and by its rotational velocity $\Omega_c = (\Omega_{c_x} \Omega_{c_y} \Omega_{c_z})^T$. These velocities can be expressed by the velocity screw $T_c = (V_c\ \Omega_c)^T$. In practice, this vision sensor can be mounted on the end effector of a manipulator or carried by a mobile robot. Due to the motion of sensor and objects, the point m moves with a relative velocity screw $T = (V\ \Omega)^T$ with respect to the camera frame (C, x_c, y_c, z_c) by means of:

$$\dot{m} = V + \Omega \times Cm \tag{2}$$

By differentiating (1) and using (2) , we can derive the well known equation relating optical flow measurement to 3D structure and motion in the scene:

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = H \,.\, T \ \text{ with} \tag{3}$$

$$H = \begin{pmatrix} 1/z & 0 & -X/z & -XY & 1+X^2 & -Y \\ 0 & 1/z & -Y/z & -(1+Y^2) & XY & X \end{pmatrix}$$

This equation shows the basic structure of the *interaction* between the vision sensor and its environment. This interaction depends on the inverse of the depth z of the point expressed in the camera frame and, generally, without some a priori knowledge about the environment, it is not possible to know the true value of the interaction.

Now, let us consider geometrical primitives more complex than points. In a general way, we assume that a 3D geometrical primitive can be represented as a multidimensional function $h(x,y,z,Q) = 0$ which projects onto the image plane under the form $g(X,Y,R) = 0$ where $Q = \{Q_i\ ;\ i = 1,m\}$ and $R = \{R_i\ ;\ i = 1,n\}$ 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 well suited parametrization of g (minimal and without singularity of representation) and to establish the interaction screw $H_i(R,Q)$ such that:

$$\dot{R}_i = H_i(R,Q).T \tag{4}$$

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

VISUAL SIGNALS

Assuming that g is a C^1 function, let us derive a general solution for computing $H_i(R,Q)$:

$$g(X,Y,R) = 0\ ,\ \forall t \Rightarrow \dot{g}(X,Y,R) = 0\ ,\ \forall t \tag{5}$$

after developments, we obtain:

$$\sum_{i=1}^{n} \frac{\partial g}{\partial R_i}\dot{R}_i = -\frac{\partial g}{\partial X}\dot{X} - \frac{\partial g}{\partial Y}\dot{Y}\ , \tag{6}$$

$$\forall(X,Y) \text{ such that } g(X,Y,R) = 0$$

Eq. (6) allows us to relate the variation of the parameters R_i which characterizes the 2D primitive in the image plane to the optical flow components and thus, to the velocity screw of the camera by means of Eq. (3). An *elementary visual signal* will be defined as a function $s = f(R_i,\cdots,R_k)$ characterizing usual geometrical properties in the image like distance between two points, orientation between two lines, surface, mass centroid and so on.

Furthermore, let us search for a velocity screw T^* such that $\dot{s} = 0$, i.e. the elementary signal is invariant with respect to the motion T^*. T^* is a solution of:

$$H\ .\ T^* = 0 \Leftrightarrow T^* \in Ker(H). \tag{7}$$

Likewise, considering the set of elementary signals $S = (s_1,\ldots,s_p)^T$, we have $\dot{S} = H.T$ with $H = (H_1,..,H_p)^T$. The velocity screw T^*, leaving S unchanged, belongs to $Ker(H)$. The set $(S;H)$ constitutes a *virtual linkage* (Espiau, 1987).

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 $Ker(H)$. The concept of virtual linkage is an easy way for the user to specify the task he wishes to realize through the visual signals. It allows to determine the motions which may be sensor-controlled and the ones which remain free.

Case of points.

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

$$h: \begin{cases} x - x_1 = 0 \\ y - y_1 = 0 \\ z - z_1 = 0 \end{cases} \longrightarrow g: \begin{cases} X - X_1 = 0 \\ Y - Y_1 = 0 \end{cases} \tag{8}$$

From the Eqs. (6) and (3), we have:

$$\begin{array}{rcl} \dot{X}_1 & = & H_1\ .\ T \\ \dot{Y}_1 & = & H_2\ .\ T \end{array} \quad \text{with} \tag{9}$$

$$\begin{array}{rcl} H_1 & = & \begin{bmatrix} 1/z_1 & 0 & -X_1/z_1 & -X_1Y_1 & 1+X_1^2 & -Y_1 \end{bmatrix} \\ H_2 & = & \begin{bmatrix} 0 & 1/z_1 & -Y_1/z_1 & -(1+Y_1^2) & X_1Y_1 & X_1 \end{bmatrix} \end{array}$$

Finally, we can define two elementary visual signal $s_1 = X_1$ and $s_2 = Y_1$. The set $(S = (s_1 s_2)^T\ ;\ H = (H_1 H_2)^T)$ constitutes a virtual linkage of class 4. A basis of $Ker(H)$ can be easily found, for example:

$$\begin{pmatrix} X_1 & 0 & z_1(1+X_1^2+Y_1^2) & 0 \\ Y_1 & 0 & 0 & z_1(1+X_1^2+Y_1^2) \\ 1 & 0 & 0 & 0 \\ 0 & X_1 & -X_1Y_1 & 1+X_1^2 \\ 0 & Y_1 & -(1+Y_1^2) & X_1Y_1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{10}$$

Each velocity screw T^* which belongs to $Ker(H)$ leaves unchanged the projection of the 3D point in the image plane.

Other elementary visual signal based on points. :

In many cases, it can be fruitful to consider more relevant signals with regard to the task. For example, let us consider the distance squared between two points:

$$s = L_{12}^2 = (X_1 - X_2)^2 + (Y_1 - Y_2)^2 \tag{11}$$

We have:

$$\dot{s} = 2(X_1 - X_2)(\dot{X}_1 - \dot{X}_2) + 2(Y_1 - Y_2)(\dot{Y}_1 - \dot{Y}_2) \tag{12}$$

Using Eq. (3), we obtain $\dot{s} = H\ T$ with:

$$H^T = \begin{pmatrix} 2\,(X_1 - X_2)(1/z_1 - 1/z_2) \\ 2\,(Y_1 - Y_2)(1/z_1 - 1/z_2) \\ 2\,[(X_1 - X_2)(X_2/z_2 - X_1/z_1) \\ +(Y_1 - Y_2)(Y_2/z_2 - Y_1/z_1)] \\ 2\,[(X_1 - X_2)(X_2Y_2 - X_1Y_1) \\ +(Y_1 - Y_2)(Y_2^2 - Y_1^2)] \\ 2\,[(X_1 - X_2)(X_1^2 - X_2^2)+ \\ (Y_1 - Y_2)(X_1Y_1 - X_2Y_2)] \\ 0 \end{pmatrix} \tag{13}$$

In a similar manner, it is possible to compute the interaction screw of any visual signal representing geometrical characteristics based on points as mass centroid, surface, orientation of inertial axis and so on.

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 instead of 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:

$$h(x,y,z,Q): \begin{cases} a_1x + b_1y + c_1z + d_1 = 0 \\ a_2x + b_2y + c_2z + d_2 = 0 \end{cases} \tag{14}$$

A 3D line in the scene projects onto the image plane as a 2D line (except on the degenerate case $d_1 = d_2 = 0$) the equation of which is $g(X, Y, R)$:

$$(a_1 d_2 - a_2 d_1)X + (b_1 d_2 - b_2 d_1)Y + (c_1 d_2 - c_2 d_1) = 0 \qquad (15)$$

Some attention has to be taken concerning the parametrization of the 2D line (i.e. the choice of R). As we have underlined previously, it has to be minimal in order to compute $\dot{R}_i$ in an non-ambiguous maner.

Representation (a, b). : Let us consider the classical representation of a 2D line. This parametrization needs two charts ($Y = aX + b$, $X = aY + b$) 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 prefer an other representation of the 2D lines.

Representation (ρ, θ). : We choose $R = \{\rho, \theta\}$ and we have $g(X, Y, R)$:

$$\mathcal{D} : X cos\theta + Y sin\theta - \rho = 0 \ , \ \theta \in] - \pi/2, \pi/2] \qquad (16)$$

deriving this equation from Eq. (6) gives:

$$\dot{\rho} + (X sin\theta - Y cos\theta)\dot{\theta} = cos\theta \dot{X} + sin\theta \dot{Y} \ , \ \forall (X, Y) \in \mathcal{D} \qquad (17)$$

finally, using Eqs. (3), (14), (16), we obtain:

$$\begin{aligned} \dot{\theta} &= (\lambda_1 cos\theta \quad \lambda_1 sin\theta \quad -\lambda_1 \rho) \cdot V & (18) \\ &\quad + (\rho cos\theta \quad \rho sin\theta \quad 1) \cdot \Omega \\ \dot{\rho} &= (\lambda_2 cos\theta \quad \lambda_2 sin\theta \quad -\lambda_2 \rho) \cdot V & (19) \\ &\quad + (-(1 + \rho^2) sin\theta \quad (1 + \rho^2) cos\theta \quad 0) \cdot \Omega \end{aligned}$$

with $\lambda_1 = (b_1 cos\theta - a_1 sin\theta)/d1$ and $\lambda_2 = -(c_1 + \rho a_1 cos\theta + \rho b_1 sin\theta)/d1$.

Eqs. (18) and (19) define the interaction screws associated to ρ and θ. 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 as a sphere which intersects a plane. We have $h(x, y, z, Q)$:

$$\begin{cases} (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0 \\ (z - z_0) - \alpha(x - x_0) - \beta(y - y_0) = 0 \end{cases} \qquad (20)$$

It projects onto the image plane as an ellipse $g(X, Y, R)$:

$$\frac{(X - X_C + e(Y - Y_C))^2}{a^2(1 + e^2)} + \frac{(Y - Y_C - +e(X - X_C))^2}{b^2(1 + e^2)} - 1 = 0 \qquad (21)$$

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:

$$\begin{pmatrix} \dot{X}_C \\ \dot{Y}_C \\ \dot{e} \\ \dot{a} \\ \dot{b} \end{pmatrix} = H \ . \ T \qquad (22)$$

VISUAL SENSING AND TASK FUNCTION

So far, 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. The problem can be stated as follows: is it possible to specify a robotic task in terms of reaching a particular configuration, which is constituted 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 to the front of a door at a given distance. This task may be translated in the image frame by the following: match in the image a set of polygonal lines which can be associated with the model of the door, control the motion of the robot to bring these lines to form a rectangle centered in the image frame, with two horizontal and vertical lines, the surface of which is equal to a given value.

For a given robotic task, we have to choose, using the concept of virtual linkage, a set S of visual signals which allows to perform the task. Then, we define a task vector $e(t)$ such that $e(t) = S(t) - S^*$ where S^* can be considered as a reference image target to be reached in the image frame and $S(t)$ as the visual signals currently observed by the camera. Considering the control problem as an output regulation problem, we can assume that the concerned task is perfectly achieved if $e(t) = 0$. We focus on the robustness with respect to the uncertainties on the interactions by using a gradient-based approach: in this approach, it is possible to define an error function ($= \|e(t)\|$) and to express the regulation problem as a minimization problem. Under these assumptions, we may choose as camera control vector, expressed in the camera frame:

$$T_c = \mu \ C \ e(t) \qquad (23)$$

where $\mu > 0$ and C is a constant matrix. We have:

$$\dot{e} = \dot{S} = H(R(t), Q(t)) \ T \qquad (24)$$

with $T = -T_c$ and, from Eq. (23), we obtain:

$$\dot{e} = -\mu \ H(R(t), Q(t)) \ . \ C \ e(t) \qquad (25)$$

An exponential convergence will be ensured under the following sufficient condition (Samson, 1987):

$$H(R(t), Q(t)) \ . \ C > 0 \ , \ \forall t \qquad (26)$$

in the sense that a $n \times n$ matrix A is positive if $x^T A x > 0$ for any nonzero $x \in \mathbf{R^n}$ (i.e. the eigenvalues of $A + A^T$ are positive).

Using the change of variable $q(t) = Ce(t)$, we have:

$$\begin{aligned} \dot{q} &= C\dot{e} = -\mu \ C \ . \ H(R(t), Q(t)) \ . \ C \ e(t) & (27) \\ &= -\mu \ C \ . \ H(R(t), Q(t)) \ q(t) & (28) \end{aligned}$$

An other sufficient convergence condition may therefore be obtained:

$$C \ . \ H(R(t), Q(t)) \ > 0 \ , \ \forall t \qquad (29)$$

A good and simple way to try to ensure this matrix positivity is to enforce $H(R^*, Q^*).C = I_p$ or $C.H(R^*, Q^*) = I_6$ (where I_n represents the $n \times n$ identity matrix) for the equilibrium position $S = S^*$. If $H(R^*, Q^*)$ is rectangular ($p \neq 6$) or not full-rank ($N > 0$), C may be chosen as the pseudo-inverse of $H(R^*, Q^*)$.

Obviously, the above presented results ensure the positivity only at the vicinity of the equilibrium position where $S = S^*$. Far from this position, the convergence property is not always insured.

Fortunately, the positivity of $H.C$ or $C.H$ is only a sufficient condition and, as we will see in the next section, the convergence and the stability is always ensured in practice.

Let us note that the computation of C depends on the parameters R^* referring to the image and on the parameters Q^* referring to the 3D scene. These parameters are chosen during the definition of the task, and, with some a priori knowledge on the shape and on the dimensions of the considered 3D-objects, the parameters R^* and Q^* can be exactly computed. For example, if the task consists in going to the front of a door, we have to assume that there is a door in the scene, and, to stop at a given distance of that door, we have to know its dimensions to compute the exact image target to be reached.

But, in many cases, the shape assumption is the only one which is necessary: indeed, if the interaction screw H can be factorized in a product of two matrices such that:

$$H(R^*, Q^*) = B(R^*) \cdot D(R^*, Q^*) \tag{30}$$

where $D(R^*, Q^*)$ is a positive diagonal matrix, we can choose a control matrix C such that C is the pseudo inverse of $B(R^*)$ since the convergence condition $C.B(R^*).D(R^*, Q^*)$ positive is always verified. For instance, when the target is constituted by 3D points and when the desired position is such that the target is parallel to the image plane, the matrix $D(R^*, Q^*)$ has the following form:

$$D(R^*, Q^*) = \left(\begin{array}{c|c} I_3/z^* & 0 \\ \hline 0 & I_3 \end{array} \right) \tag{31}$$

z^* being the unknown distance between the target and the image plane. Any arbitrary distance $\hat{z}$ can be used to compute the image target to be reached, but it is impossible, without the assumption of the dimensions of the object, to know the final distance between the target and the image plane.

Finally, let us remark that the approach described above considers that the objects of the scene may be mobile with an unknown motion consistent with the sampling of the servocontroller loop. Indeed, the motion of the objects will be compensated by a motion of the camera in order to obtain the chosen image target.

RESULTS

The approach developped in this paper has been validated both in simulation and in an experimental cell constituted by a CCD camera mounted on the end-effector of a 6dof manipulator (Chaumette, 1989).

Simulation results

In this section, three different tasks are described: the first one is based on using points as elementary visual signals, the second one on using lines, and the last one on using circles.

Case of points. : The first simulated task consists in positioning the camera with respect to a square in order to obtain a centered square in the image such that its sides are parallel to the image axes. The coordinates of each square corner are taken as elementary visual signals: $S = (X_1, ..., X_4, Y_1, ..., Y_4)^T$. The image target to be reached is fixed to $S^* = (-A, A, A, -A, A, A, -A, -A)^T$ where $A = L/2z^*$, L being the lenght of the square side and z^* being the desired distance between the image plane and the square. The 6×8 control matrix C is chosen as the pseudo inverse of the interaction screw associated to S^* and computed using Eq. (9).

Some results are given in Fig. 3: the two top windows present the relative attitude of the camera (symbolized by a pyramid) and the target viewed by an outside observer. The two bottom windows present the target as seen by the camera. The windows on the left correspond to the initial position and the windows in the center to the final one. The two windows on the right show the behaviour of the error $\|e(t)\|$ (top window) and of each component of the control vector T_c during the visual servoing (bottom window).

Let us note that, during this simulation, the image noise had a normal distribution with zero mean and standard deviation of two pixels, and that the control vector noise had a normal distribution with zero mean and standard deviation of 1 cm in translation and two degrees in rotation.

As shown on Fig. 3, the convergence to the desired image target is performed in spite of the distant initial position. The same experiments are successfully performed when the target is moving through the scene with any motion consistent with the sampling of the servocontroller loop.

Case of lines. : Let us now consider a task aimed to position a camera with respect to a 'road', which is symbolized by three parallel straight lines in a plane (lateral and central white bands). The goal position is such that:

- the camera lies at a height y^* at the middle of the right lane;
- the camera axis $\vec{z}$ coincides with its direction and its axis $\vec{y}$ is vertical.

Naturally, the representation (ρ, θ) is used to characterize each 2D-line, and, from these assumptions, it is possible to build the desired image target:

$$h_1(\bar{x}, \bar{p}^*) : \begin{cases} y + y^* = 0 \\ x + l/4 = 0 \end{cases} \Rightarrow \begin{cases} \theta_1^* = \arctan(-l/4y^*) \\ \rho_1^* = 0 \end{cases} \tag{32}$$

$$h_2(\bar{x}, \bar{p}^*) : \begin{cases} y + y^* = 0 \\ x - l/4 = 0 \end{cases} \Rightarrow \begin{cases} \theta_2^* = \arctan(l/4y^*) \\ \rho_2^* = 0 \end{cases} \tag{33}$$

$$h_3(\bar{x}, \bar{p}^*) : \begin{cases} y + y^* = 0 \\ x - 3l/4 = 0 \end{cases} \Rightarrow \begin{cases} \theta_3^* = \arctan(3l/4y^*) \\ \rho_3^* = 0 \end{cases} \tag{34}$$

From Eqs. (18) and (19), it is easy to compute the interaction screw H associated to $S^* = (\rho_1^*, \theta_1^*, \rho_2^*, \theta_2^*, \rho_3^*, \theta_3^*)^T$:

$$H = \begin{pmatrix} -\cos^2\theta_1^*/y^* & -\cos\theta_1^*\sin\theta_1^*/y^* & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & sin\theta_1^* & -\cos\theta_1^* & 0 \\ -\cos^2\theta_2^*/y^* & -\cos\theta_2^*\sin\theta_2^*/y^* & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & sin\theta_2^* & -\cos\theta_2^* & 0 \\ -\cos^2\theta_3^*/y^* & -\cos\theta_3^*\sin\theta_3^*/y^* & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & sin\theta_3^* & -\cos\theta_3^* & 0 \end{pmatrix} \tag{35}$$

The set $(S; H)$ constitutes a virtual linkage of class 1: the allowed motion is the translation V_{c_z}. In Fig. 4, configurated as Fig. 3, this motion has been fixed to 1cm/s. The control law uses the others degrees of freedom of the camera to reach the desired image target.

Case of circles. : The same approach is used for positioning the camera in front of a tube in order to obtain two centered concentric circles in the image. The set $(S = (X_{c_1}, Y_{c_1}, a_1, b_1, X_{c_2}, Y_{c_2}, a_2, b_2)^T; H)$ constitutes a virtual linkage of class 1 at the configuration $S^* = (0, 0, R_1, R_1, 0, 0, R_2, R_2)^T$ where R_1 and R_2 are the desired radii. The allowed motion is the rotation Ω_{c_z}, fixed to 4.5 degrees/s in Fig. 5.

by Eq. (??). The choice of the control matrix has therefore been experimentally shown to be robust, even for rather large initial position errors. Figure 6 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 $\|e\|$ as in the simulation case, and the behaviour of each elementary signal error $(s_i - s_i^*)$ involved in the control loop.

Figure 2: Experimental Testbed

CONCLUSION

This paper has adressed the problem of using vision data directly as an input to the robot control loop. This approach is characterized by the use of vision sensor as local sensor, and by the exploitation of the vision data into a robust control scheme.

Concerning the first point, vision data are modelled as a set of *elementary visual signals.* The interaction between the sensor and the scene can be described by a screw which relates the behaviour of the signals to the relative motion between the camera and the objects.

Concerning the second point, we have proposed a robust control scheme based on a task function approach. 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 regulation problem as a minimization problem. In this case, a class of control based on gradient techniques and a choice of the control matrix based on the interaction screw allow to perform correctly the task with good convergence properties.

In the near future, we should hope to address the problem of programming complex robotics tasks in terms of a succession of elementary subtasks which can be performed using the approach described in this paper.

REFERENCES

Chaumette, F., and P. Rives (1989). Réalisation et calibration d'un système expérimental de vision composé d'une caméra mobile embarquée sur un robot-manipulateur. *INRIA Research Report 994.*

Espiau, B. (1987). Sensory-based Control: Robustness issues and modelling techniques. Application to proximity sensing. *NATO Workshop on Kinematic and Dynamic Issues in Sensor Based Control,* Italy, October 1987.

Feddema, J.T., C.S.G. Lee, and O.R. Mitchell (1989). Automatic selection of image features for visual servoing of a robot manipulator. *Conf. IEEE Robotics and Automation,* Scottsdale, Arizona, USA, May 14-19.

Samson, C. (1987). Une approche pour la synthèse et l'analyse de la commande des robots manipulateurs rigides. *INRIA Research Report 669.*

Weiss, L.E. (1984). Dynamic visual servo control of robots. An adaptive image based approach. *Carnegie Mellon Technical Report,* CMU-RI-TR-84-16.

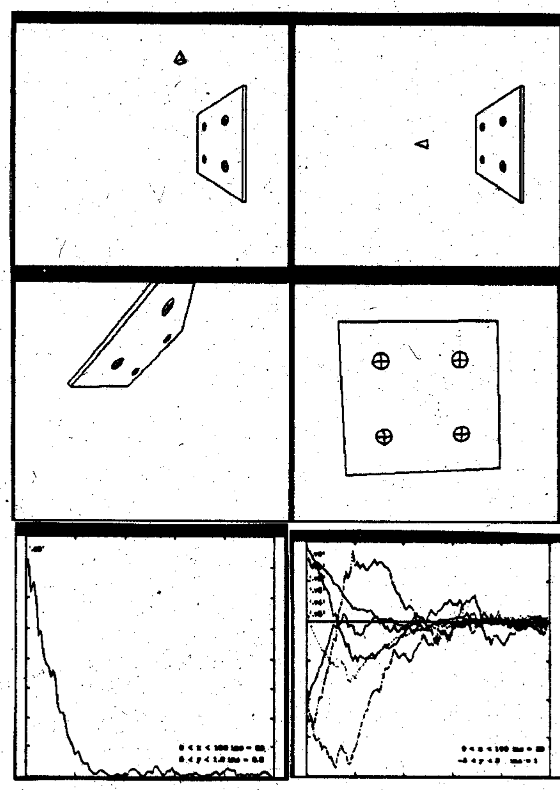

Fig. 3. Positioning in front of a square (with noise).

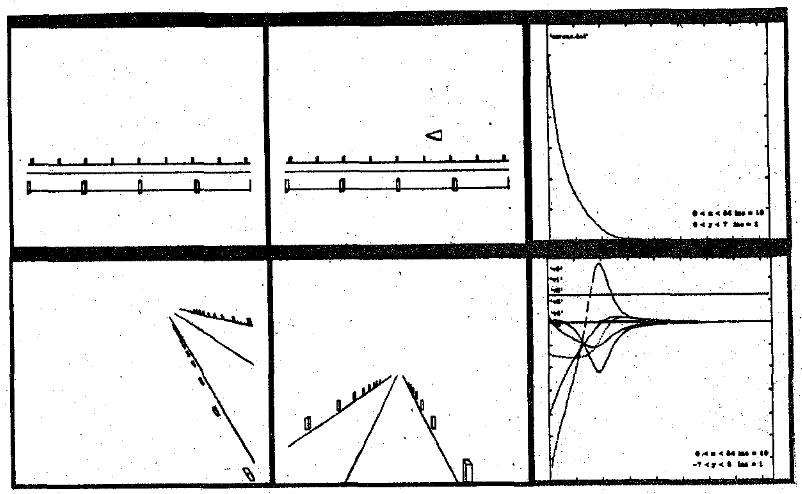

Fig. 4. Positioning and moving over a road (without noise).

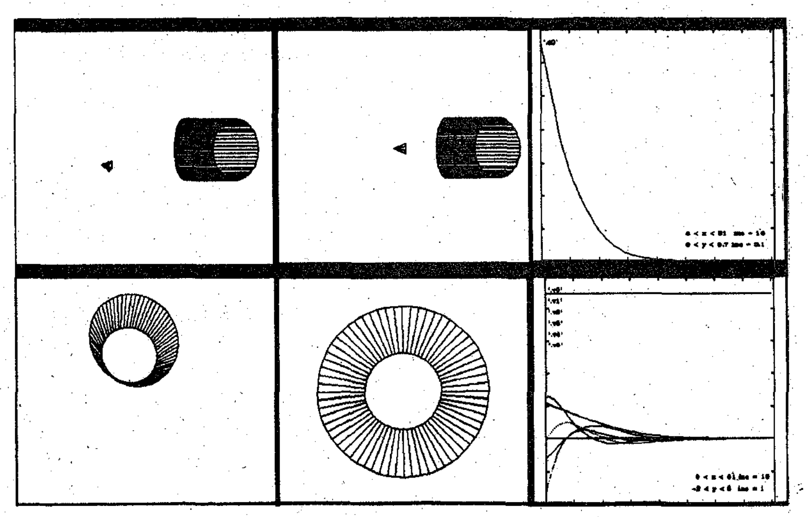

Fig. 5. Positioning in front of a tube (without noise).

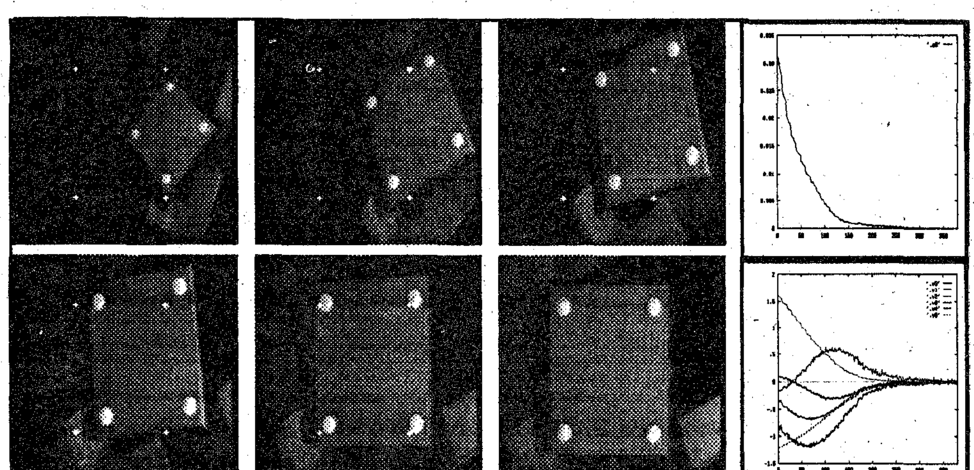

Fig. 6. Real experiment.