Visual servoing based on ellipse features

P. Rives, H. Michel

INRIA Sophia Antipolis

2004, Route des Lucioles - BP 93, 06902 Sophia-Antipolis Cedex (France)

ABSTRACT

The work presented here focuses on the problem of positioning a mechanical structure with respect to a deep underwater bore-hole by using a visual servoing approach. First, we will briefly recall a paradigm we have been using as a theoretical framework for implementing visual servoing tasks. Then, we will apply this general formalism to our particular case where the visual features are two ellipses perceived in the image which correspond to the projection of two circles bounding the bore-hole in the scene.

The last part is devoted to implementation aspects. We will show some results which have been obtained both in simulation and on our testbed consisting of a 6 degrees–of–freedom arm with a camera mounted on its end effector. Among implementation aspects, the control loop sampling rate is the most crucial issue. In a vision–based control approach, we have to deal with strong real–time constraints in terms of image processing in order to ensure performances and stability of the closed loop control scheme. To solve this problem, we have designed a new parallel machine vision architecture fully adapted to vision–based control approaches and able to achieve real–time video rate performances for the servoing loop.

1 INTRODUCTION

In the near future, underwater robotics will certainly be one of the most exciting application fields for autonomous vehicles. Nowadays, underwater operations are performed directly by skin-divers or teleoperated from an underwater manned vehicle. In all cases, they are very expensive. Among the different types of underwater missions which will be useful to automatize, docking missions play a major role. For these reasons, IFREMER (the French agency in charge of oceanographic research) has started a program aimed at designing an underwater autonomous vehicle, NADIA, which will be able to perform automatic docking missions. In this program, we are in charge of validating sensor based control approaches for this purpose. In opposite to classical robotics approaches, the visual servoing approach [8],[3], does not require an explicit 3D reconstruction of the scene. The basic idea is to assume that a task can be fully specified in terms of features in an image. The task is achieved when a desired image of the scene is reached. In terms of control, this can be formulated as a problem of regulation to zero of a certain output function directly defined in the image frame. When implementing a visual servoing application several problems need to be solved . In a first step, we have to specify the task in terms of visual signals to be controlled in the image. The second step requires the choice of the image processing methods for extracting these signals. The last step consists of designing a closed loop control scheme law based on these visual signals which allows the image perceived by the camera to converge to the desired one. In previous papers, we established a general framework based on theoretical formalisms, such as mechanical linkages theory, screw spaces properties and a task function approach to Robotics, which allows the design of vision–based tasks. The central issue of such a framework is the modeling of the interactions between the sensor attached to the robot and its local environment. In the present paper, we only recall the main characteristics of this formalism, more details can be found in [5], [1], [6] and [2].

2 VISUAL SERVOING FRAMEWORK

2.1 Modeling the interaction between the sensor and the environment

We consider a camera (C). This camera is carried by a robot which is a mechanical system with several degrees of freedom. The attitude (position and orientation) of (C) is an element $\bar{r}$ of $\mathbb{R}^3 \times SO_3$ which is a six dimensional differential manifold. (C) interacts with an environment. Let us assume that the image given by (C) is fully characterized by the relative position of (C) with respect to (T) (we don't consider image changes dues to lighting variations). Moreover, let us consider that the information in the image may be modeled as a set of *elementary signals* characterizing the geometric features which result from the projection onto the image of the 3D objects belonging to the scene. Then, from these elementary signals a vector $s(\bar{r}, t)$ is constructed. The dependence on t represents possible motions of the objects in the scene. As an important result (see [2] for more details), if we take s to be a differentiable mapping from $\mathbb{R}^3 \times SO_3$ to $\mathbb{R}$ (one dimensional signal) then the derivative of s can be expressed by a screw product :

$$\dot{\underline{s}} = H \bullet T_{CT}, \tag{1}$$

where H is called the *interaction screw*, T_{CT} is the velocity screw of the camera and $\bullet$ represents the screw product.

If we now consider multi–dimensionnal vectors of signals we define the *interaction matrix* L^T whose rows are interaction screws :

$$\dot{\underline{s}} = L^T . T_{CT}, \tag{2}$$

This interaction matrix can be derived for many exteroceptive sensors. In the case of vision sensors, elementary signals can be chosen among a set of geometric primitives like points, lines, circles.... An analytical expression for the interaction matrix when the image features are general algebraic curves can also be obtained (for more details, see [1]).

2.2 Relating interaction matrix and mechanism theory

Let us now consider motions of the camera which leave several visual signals s unaffected, this motions (T^*_{CT}) sre such that :

$$\dot{\underline{s}} = L^T . T^*_{CT} = 0 \tag{3}$$

So these motions T^*_{CT} need to belong to the kernel of the interaction matrix. Imposing contraints on the visual signals, consequently impose constraints on the compatible motions of the camera. By analogy with mechanisms theory, we say that these visual signals constitute a *virtual linkage* between the sensor and the objects of the environment. This concept of virtual linkage provides a simple way to design vision–based tasks by applying the following steps :

- Describe the robotic task in terms of a virtual linkage between the camera and the environment (i.e bringing a mobile robot in front of a door can be understood as aligning the optical axis of the camera with a particular axis which is perpendicular to the door).
- Select a set of elementary visual signals $\underline{s}$ able to satisfy this virtual linkage (i.e. the corners or the edges of the door).
- Compute the desired value of the vector of elementary visual signals $\underline{s^*}$ corresponding to the desired final attitude between the robot and its environment (i.e. two vertical and one horizontal lines in the image corresponding to the edges of the door).
- Perform a robust control scheme ensuring convergence from the current value of the vector of elementary visual signals in the image to the desired one.

With this approach, it is possible to specify a low level robotic task in terms of a *target image* to be reached when the task is achieved. In the next section, we will show how such an approach can be embedded in a robust closed loop control scheme using the *task function approach* [6].

2.3 vision–based control

A now classical approach in robotics is to consider the process of achieving a task such as tracking or positioning as a problem of regulation of a certain function: *the task function* (see [6] for full details about the task function approach in robot control). The application of the task function approach to vision–based control is straightforward: the task is now described as desired values for a certain set of elementary visual signals. In mathematical terms a visual task function vector $\underline{e}$ is :

$$\underline{e}\ (t) = C\ (\underline{s}\ (\bar{r}, t) - \underline{s}^*(t)) \tag{4}$$

- $s\ (\bar{r}, t)$ is the value of the visual features currently observed by the camera, which only depends on the relative attitude between the camera and the scene.
- s^* is the desired value for s
- C is a combination matrix.

For simplicity, we deal with a very simple example of *velocity control scheme* using visual feedback (for more advanced control schemes, see [6]) :

$$T_c = -\lambda e \tag{5}$$

with $\lambda > 0$.

If we assume that $\underline{e}$ depends only on $(\bar{r}, t)$, we get :

$$\dot{e} = \frac{\partial e}{\partial s} . \frac{\partial s}{\partial \bar{r}} . T_c + \frac{\partial e}{\partial t} \tag{6}$$

$$\dot{e} = C.L^T.T_c + \frac{\partial e}{\partial t} \tag{7}$$

where $\frac{\partial e}{\partial t}$ parameterizes a possible motion of the target. In the case of a motionless target $\frac{\partial e}{\partial t} = 0$, and we obtain :

$$\dot{e} = C.L^T.T_c = -\lambda\ C.L^T.e \tag{8}$$

An exponential convergence will thus be ensured under the sufficient condition:

$$CL^T > 0 \tag{9}$$

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

A good and simple way to satisfy this convergence condition in the neighborhood of the desired position is to choose for the matrix C the pseudo–inverse of the interaction matrix associated to s^*:

$$C = {L^T}^+_{|s=s^*} \tag{10}$$

3 USING ELLIPSES AS FEATURES

Since we will use the projections of two circles (seen as ellipses in the camera image) for our underwater application, we now recall some basics aspects of the projection of circles and the representation of ellipses.

3.1 Circles projections and ellipses representation

We will use the classical pinhole model to describe the transformations of 3D features to 2D features observed in the camera image. Furthermore, we will assume that the camera has been properly calibrated so that we can describe the 3D to 2D mapping by the following equation :

$$x = \frac{X}{Z} \; , \; y = \frac{Y}{Z} \tag{11}$$

Where $(X\,Y\,Z)$ are the coordinates of a 3D point in a coordinates frame attached to the camera with origin located at the optical center. This 3D point is projected onto the image frame in a 2D point of coordinates $(x\,y)$.

As a well known result, a 3D circle projects onto the image frame as a 2D ellipse. Let us express this 2D ellipse as a function of the relative attitude between the camera and the 3D circle. In what follows the matrix $\mathcal{M}$ is such that the equation of the projected ellipse can be given as :

$$(x\;y\;1)\;\mathcal{M}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

Its expression is :

$$\mathcal{M} = (\vec{\mathcal{N}}.\vec{X_C})^2\, I - (\vec{\mathcal{N}}.\vec{X_C})\,(\vec{X_C} \otimes \vec{\mathcal{N}} + \vec{\mathcal{N}} \otimes \vec{X_C}) + P\,(\vec{\mathcal{N}} \otimes \vec{\mathcal{N}}) \tag{12}$$

With the notations :

P is the *power* of the optical center of the camera with respect to the sphere which has the the same radius as the given 3D circle and contains this circle.

I is the 3 by 3 identity matrix.

$\vec{X_C}$ is the vector from the optical center to the center of the 3D circle expressed in a coordinates frame attached to the camera where equation 11 holds.

$\vec{\mathcal{N}}$ is a vector which is normal to the plane containing the 3D reference circle, also expressed in the same coordinates frame.

$\vec{A} \otimes \vec{B}$ is the *tensor product* of vectors $\vec{A}$ and $\vec{B}$.

$\mathcal{M}$ represents the *projective cone* under which a 3D circle is observed from a point. It is no more than the equation of the ellipse observed in the camera image in homogeneous coordinates.

At this point we have obtained an equation which relates the 2D visual features observed in the image to the relative attitude of the camera and the 3D features. We now deal with a mapping from $\mathbb{R}^3 \times SO_3$ (attitudes of the camera) to the set of algebraic curves in P^2 (projective plane) which have degree 2 or less. For the sake of simplicity we will restrict our study to cases where the projections are non degenerate, that is to say that the mentionedc urves in P^2 correspond to non degenerate ellipses in $\mathbb{R}^2$, no such things as hyperbolas or lines in the camera image are allowed.

The desired mathematical result which is needed to construct our visual servoing application is the expression of the *interaction matrix*. This matrix will describe how the ellipse observed in the camera image (which can be represented by $\mathcal{M}$) changes as the result of a change in the attitude of the camera which consequently modifies P,$\vec{X_C}$, and $\vec{\mathcal{N}}$. For doing that, we need to relate the projective form to another one more adapted to the computation from the image processing point of view. This representation of projected ellipses should have several properties :

1. Obviously we want an ellipse to be represented as a vector in a finite dimension vector space.
2. The representation must be *complete*, that is to say any ellipse which can be observed can be represented within this representation class.
3. *ambiguous* representation should be avoided. We do not want the same ellipse to be represented by different possible parameters sets.

For many reasons (see [1] for discussion on other representations), we have chosen a representation by moments.

3.1.1 Representation by moments

Following classical notations we will define the moment of order i, j of an ellipse ε by the following surface integral, over the *interior* of the ellipse :

$$m_\varepsilon^{i,j} = \int\int x^i y^j dx dy$$

Often a modified version of these moments is used, it is called representation by *reduced moments.*

Definition 1 *We define the reduced moments of an ellipse to be the real numbers* $(x_c, y_c, \mu^{1,1}, \mu^{2,0}, \mu^{0,2})$ *such that :*

$$\left\{ \begin{array}{ccc} x_c & = & \frac{m^{1,0}}{m^{0,0}} \\ y_c & = & \frac{m^{1,0}}{m^{0,0}} \\ \mu^{1,1} & = & \frac{m^{1,1} - m^{0,0} x_c y_c}{4m^{0,0}} \\ \mu^{2,0} & = & \frac{m^{2,0} - m^{0,0} {x_c}^2}{4m^{0,0}} \\ \mu^{0,2} & = & \frac{m^{0,2} - m^{0,0} {y_c}^2}{4m^{0,0}} \end{array} \right.$$

This representation has nice properties of regularity for our application :

Theorem 1 *Reduced moments of an ellipse have the following properties :*

- *the two first moments are the coordinates of the* center *ot the ellipse.*
- *the 3 last moments are related to* Ψ *the positive definite bilinear form associated with the ellipse, in a very simple way, as is shown in the following equation :*

$$\begin{pmatrix} \mu^{2,0} & \mu^{1,1} \\ \mu^{1,1} & \mu^{0,2} \end{pmatrix} = M^{-1} \tag{13}$$

Ψ *is such that in a coordinates frame whose origin is the* center *of the ellipse, the equation of the ellipse is* $\Psi(\bar{x}) = 1$, M *is the matrix associated to* Ψ.

So, reduced moments is an ellipse representation of the split form : representation of the center, representation of the associated bilinear form, they are perfectly regular on the set of all possible non degenerate ellipses, since the function *inverse of a matrix* is continuous at any order of differentiation on the set of definite positive symmetric matrices. Other choices of a function of M (the matrix associated to Ψ) would have been as interesting from a strictly mathematical point of view, we could have taken the exponentiation of a matrix for example.... But reduced moments have an extra, and very practical, interest since they can be measured almost immediately from a digital image of the ellipse. As a consequence, we will use this representation for our application.

3.2 Computing the Interaction Matrix

We will not get into the details of the rather fastidious computations involved. We can get the desired result by expressing the time derivatives of the quantities $P, \vec{X_C}, \vec{\mathcal{N}}$, and consequently $\mathcal{M}$ which appear in equation 12 as a function of the kinematic screw and then expressing the time derivatives of the reduced moments as a function of the time derivatives of the matrix $\mathcal{M}$.

The interaction matrix is always of rank 5 for any non degenerate ellipse observed in the image except when the axis of the camera and the axis of the circle are identical. By *axis of the circle* we mean the axis which pass through the center of the 3D circle and is orthogonal to the plane containing this circle. In this case the ellipse in the image is in fact a circle and the interaction matrix (of rank 3) has the following form :

$$L^T = \begin{pmatrix} -1/Z_0 & 0 & 0 & 0 & -1-r^2 & 0 \\ 0 & -1/Z_0 & 0 & 1+r^2 & 0 & 0 \\ 0 & 0 & 2r^2/Z_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2r^2/Z_0 & 0 & 0 & 0 \end{pmatrix} \tag{14}$$

We call Z_0 the ("vertical") distance from the optical center to the center of the circle, r is the radius of the circle observed in the image (we have then $\mu^{2,0} = \mu^{0,2} = r^2$).

4 APPLICATION TO A BORE-HOLE REENTRY TASK

The problem which has been addressed here concerns the automatic positioning of a mechanical structure with respect to a deep underwater bore-hole by using a visual servoing approach. In practice, the bore-hole can be viewed as a kind of funnel bounded by two circles. For both practical and technical reasons, we have split the whole docking mission in three steps :

- initial phase : we bring the underwater vehicle to a position where the bore-hole is visible. This initial approach will be done with sonar devices.
- alignment of the optical axis of the camera with the axis of the bore-hole : this phase is achieved by a visual servoing task. The precise height with respect to the plane containing the upper circle of the bore-hole does not need to be precisely regulated.
- descent phase : once the previous requirement of having both axis aligned is achieved, we start a second task (compatible with the previous one) which causes the underwater vehicle to land on the bore-hole following a given height versus time trajectory. Among the 6 parameters which specify the attitude of the camera, only 5 parameters can be controlled, an extra degree of freedom (the rotation whose axis is the axis of the bore-hole) is then left free by the virtual linkage.

Due to the low quality of the submarine images, the only features which can be robustly extracted corresponds to the projections of the upper circle and the lower circle bounding the bore-hole. From a naive point of view, knowing that the dimension of the task space to be controlled by vision is 5, we could deduce that only one ellipse corresponding to the projection of the upper circle of the bore-hole is necessary and sufficient to perfectly achieve the task. However, as mentioned above, when a 3D circle projects onto the image as a 2D circle, we have a loss of rank for the interaction matrix corresponding to a singularity in the visual Jacobian. Unfortunately in our particular task, this singularity corresponds precisely to the desired final configuration when docking is achieved. More precisely, if we just linearize the behavior of the system around the desired final configuration, we get a system that is no more controllable. For this reason, it is necessary to use an interaction matrix computed from the two 3D circles bounding the bore-hole. We now potentially deal with 10 visual features signals, 5 reduced moments for each ellipse. Those 10 signals are not independent, there exist relations between them, an area of further research would be to decide if there is an optimal way to combine these signals and construct the *best* combination matrix C as described in the previous section on the task–function framework.

We conducted both simulations and real experiments to finally choose a particular combination of the visual signals, which exhibits acceptable convergence properties for the type of simple control loops that we wished to use. These control loops are characterized by the following :

- The imposed kinematic screw is proportional to the error signal.
- As the computation of the precise value of the interaction matrix is both long and sensitive to measurements errors, we wish to use an approximation of the real interaction matrix. Most of the time, the one which corresponds to the desired final configuration will do the job.

The vector of visual signals used, is the following :

$$s = \begin{pmatrix} x_{c1} \\ y_{c1} \\ x_{c2} \\ y_{c2} \\ \zeta \end{pmatrix} \quad \text{with :} \quad \zeta = \lambda \frac{1}{\sqrt{\mu_1^{2,0} + \mu_1^{0,2}}} \tag{15}$$

This form of a visual signals vector includes the coordinates of the centers of both ellipses, plus one parameter ζ which depends only on the projection of the upper circle. This particular choice is motivated by the following facts :

- The projection of the upper circle is easier to detect in the image (because it is bigger). So, parameters which depend only on reduced moments of this ellipse are expected to be less sensitive to measurement noises.
- It is highly desired to have a decoupling on the signal errors, since in this case, an exponential convergence of each of the visual signals can be expected. When we reach the desired configurations where both camera axis and cone axis are aligned, we get a partially decoupled system, the positions of ellipses centers depend only on components of translation and rotation velocities which are orthogonal to the axis of the cone, when ζ is an affine function of the "vertical" distance above the cone.

The λ parameter can be used to introduce a relative importance between the two following compatible tasks :

- alignment of camera and cone axis, which is achieved when the first 4 visual signals are zero.
- Once the previous task is achieved, we can start a descent task, based on the regulation of the ζ signal.

The interaction matrix computed when the two axis are aligned is :

$$L^T = \begin{pmatrix} 1/Z & 0 & 0 & 0 & -1-(\frac{R_1}{Z})^2 & 0 \\ 0 & 1/Z & 0 & 1+(\frac{R_1}{Z})^2 & 0 & 0 \\ 1/(Z-H) & 0 & 0 & 0 & -1-(\frac{R_2}{Z-H})^2 & 0 \\ 0 & 1/(Z-H) & 0 & 1+(\frac{R_2}{Z-H})^2 & 0 & 0 \\ 0 & 0 & -\frac{\lambda}{R_1} & 0 & 0 & 0 \end{pmatrix} \tag{16}$$

Z is the distance between the optical center and the plane containing the upper circle, positive towards the bottom of the cone. H the distance between the two planes containing each of the 3D circles.

This matrix has rank 5, and ζ is decoupled from the other signals.

4.1 Simulations

The goal of these simulations was to decipher whether it was feasible to achieve the alignment of camera and cone axis task with a gross approximation of the actual interaction matrix, assuming a constant interaction matrix corresponding to the desired final configuration.

We assume an ideal kinematic behavior of the vehicle equipped with the camera, that is the imposed kinematic screw is instantly achieved (no inertia). 3D lengths are given in meters, values correspond to the actual sizes of the scaled cone used on our testbed (radius of upper cone circle is 0.1 meters). Visual signals are scaled for a camera focal length of 1 meter. Times are in seconds. In the figures shown, the initial position of the camera is : $(X = 0.2, Y = 0.2, Z = 0.4)$, the angle between camera axis and cone axis directions is approximately 36 degrees.

The position of the optical center exhibits a nice decreasing behavior (see figure 1). The behavior of the angle between the directions of the two axis is similar. More interesting is the behavior of the visual signals corresponding to the centers of the ellipse. Figure 2 shows the coordinates of the center of the ellipse which is the projection of the upper circle. Clearly we are far from the ideal exponentially decreasing behavior. This is because this nice behavior can only be expected on the *norm* of the task error function not on all elementary signals. Figure 3 shows the expected behavior of the norm of the global error, the behavior of the ζ signal is similar.

5 IMPLEMENTATION ISSUES

We have validated our visual servoing approach based on ellipse features on our testbed consisting of a 6 degrees of freedom arm with a camera mounted on its end effector. Experiments have been made by using a scale model of the bore-hole. We start from an initial viewpoint where the two ellipses corresponding to the projection of the two circles bounding the conic section of the bore hole are present in the image. The robotics task consists of aligning the optical axis of the camera with the vertical axis of the bore hole. In terms of visual servoing, it means converging towards a target image consisting of two concentric circles centered in the image frame. As in simulation, we use a constant combination matrix $C = {L^T}^+_{|s=s^*}$ corresponding to the desired target image in the control scheme,. Experimental results confirm those obtained in simulation and a video tape will be shown at the conference.

Concerning the implementation aspects, we would like to focus on the critical problem of sampling control rate. As a matter of fact, in a visual servoing approach, the sampling control rate is directly given by the image processing rate. In order to ensure the high performance of the control, this sampling rate has to be sufficiently high. For the most current applications in visual servoing, a rate of 10 images per second can be considered as the lower limit. The lack of well adapted image processing boards able to process large sequences of images in real–time, considerably bridles the development of this kind of method. For these reasons, we have designed, in collaboration with the Electronics Laboratory at the University Blaise Pascal in Clermont–Ferrand, a new architecture (hardware and software) specially dedicated to visual servoing applications (for more details on this architecture, see [4]). This architecture is based on the concept of *active windows*. An active window is attached to a particular region of interest in the image and is in charge of extracting a desired feature in this region of interest. Several active windows can be defined in the image at the same time, and different processing can be attached to each window. Fortunately, in visual servoing applications, we mainly deal with image processing algorithms which only request low or intermediate level processing. We take account of this fact at the hardware level by using a mixed architecture :

- **Pipeline Architecture** is associated to the low level allowing to efficiently perform processing such as convolution which are repeated through the window. In practice, that is done by means of VLSI convolvers which go up to 5X5 mask size. This step can be performed in real–time with a delay directly proportional to the size of the mask. The input to the algorithm is the active window and the values of the mask parameters corresponding to the desired processing. The output will consist of a list of pixels which potentially belong to the geometric features present in the window.

- **MIMD Architecture** is in charge of the intermediate level. At this level, we find merging-like algorithms which take as input the list of pixels of interest provided by the low level stage and merge them to compute a more structured representation of the geometric features present in the window (for example, slope and distance to the origin for a line or coordinates of center and radius for a circle). In our hardware, this part is done by using multi DSP's boards.

This architecture is characterized by its modularity and its real–time capabilities (figure 5). Depending on the complexity of the application, the system is built around one or more of basic modules in such a way that we are able to achieve video rate performance. Three basic modules (each of them corresponding to a VME board) have been designed :

- **WINDIS Window Distributor Subsystem:** This module takes as input an image flow fully compatible with the Datacube Maxbus requirements and is in charge of window extraction, the execution of low level processing and the distribution of active windows to the Window Processing Subsystem. Window distribution consists of dispatching lists of selected pixels and grey levels corresponding to the active window through a 20Mhz window bus.

- **WINPROC Window Processing Subsystem:** This module executes intermediate level processing on windows. It is built around DSP 96002's processors and provides a geometric description of the primitives that are being searched for in each window (these are the elementary visual signals used as feedback). Up to four DSP modules can be associated with each VME mother board (providing around 150 MFlops of computer power) and one WINDIS board can drive up to four WINPROC boards.

- **WINMAN Window Manager Subsystem :** The window manager controls distributor and DSP modules, and executes high level processing of application tasks. Moreover, it is in charge of tracking the active windows along the sequence. A 68040 based cpu board implements this module.

At the system level, all the boards are integrated under the real–time operating system VxWorks and the C programming language is used at the application level.

Using such a machine in our experiment allows us to perform the fitting and the tracking of the two ellipses at video rate along the sequence. To reach these real–time performances, one WINDIS board and one WINPROC with only one DSP processor are necessary. Two active windows have been used, one for each ellipse. The pixels corresponding to the ellipses are extracted ,in the WINDIS board, by an edge operator using a 3X3 mask and flushed to the WINPROC board where the parameters of the ellipses are computed by using a moment representation. The vector of these parameters directly constitute the visual feedback used in the visual servoing loop. Such performance at the image processing level allows us to ensure a 25Hz sampling rate for the complete visual servoing loop.

6 CONCLUSIONS

Recent advances in vision sensor technology and vision processing now enable the use of vision data for control in real–time robots. This approach, often referred to as visual servoing or vision–based control, is particularly well adapted when the task can be expressed in a natural way as a control of the relative attitude (position and orientation) of the camera with regard to a particular part of the scene. Many application fields involve this kind of tasks : automatic vehicle guidance, space docking, target tracking. The execution of these tasks is fully controlled in the image frame and, by virtue of this fact, does not depend on the actual reconstruction of absolute positions of the camera and objects of the scene. In this paper, we have presented an example of such techniques applied to underwater robotics. Thanks to a powerful theoretical formalism, we can specify visual servoing tasks based on ellipse features in a simple way. Implementation issues point out the need for real–time image processing to ensure performance and stability of the vision–based control scheme. The lack of well adapted image processing boards able to process, in real–time, large sequence of images considerably bridles the development of this kind of approach. For these reasons, we have developed a dedicated vision hardware system allowing us to perform vision–based tasks at video rate. With this system, it is now possible to ensure a 25Hz sampling rate for the complete visual servoing loop even if the visual features are as complex as ellipses.

7 REFERENCES

[1] F. Chaumette: *La relation vision-commande: théorie et applications à des tâches robotiques*, Ph-D Thesis,University of Rennes I, France, July 1990.

[2] Espiau B., Chaumette F., Rives P. : *A New Approach to Visual Servoing in Robotics*, IEEE Transaction on Robotics and Automation, Vol. 8, No. 3, June 1992

[3] M. Kabuka, E. McVey, P. Shironoshita : *An Adaptive Approach to Video Tracking*, IEEE Journal of Robotics and Automation, 4(2):228-236, April 1988.

[4] Martinet P.,Rives P.,Fickinger P.,Borrelly J.J : *Parallel Architecture for Visual Servoing Applications*, Workshop on Computer Architecture for Machine Perception, Paris, Dec. 1991

[5] P. Rives, F. Chaumette, B. Espiau: *Visual Servoing Based on a Task Function Approach*, First International Symposium on Experimental Robotics, Montréal, Canada, June 1989.

[6] C. Samson, B. Espiau, M. Le Borgne: *Robot Control: the Task Function Approach*, Oxford University Press, 1991.

[7] L. E. Weiss *Dynamic Visual Servo Control of Robots. An Adaptive Image based Approach*, Technical Report, CMU-RI-TR-84-16; Carnegie Mellon

[8] L. E. Weiss, A. C. Sanderson: *Dynamic Sensor-Based Control of Robots with Visual Feedback*, IEEE Journal of Robotics and Automation, Vol. RA-3, n. 5, Oct. 1987.

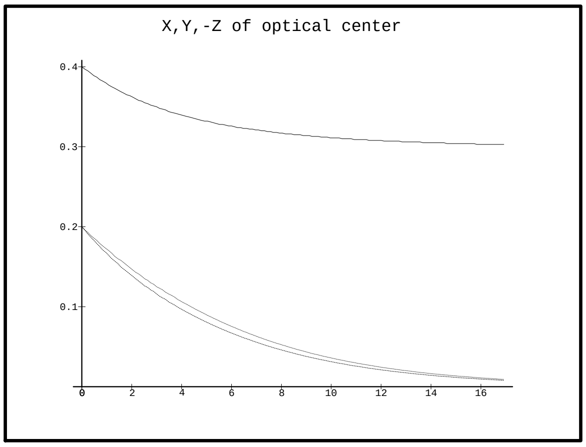

Figure 1: Coordinates of optical center versus time

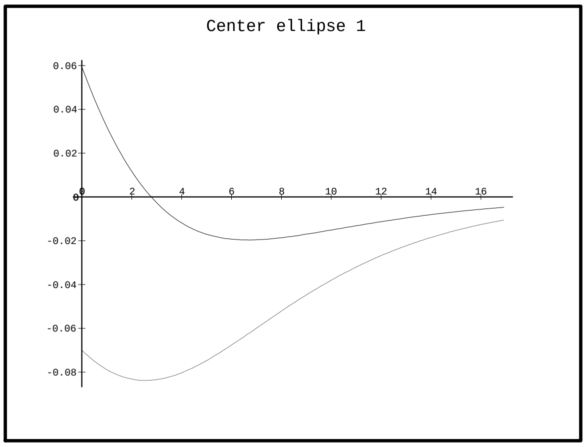

Figure 2: Coordinates of center of the big ellipse versus time

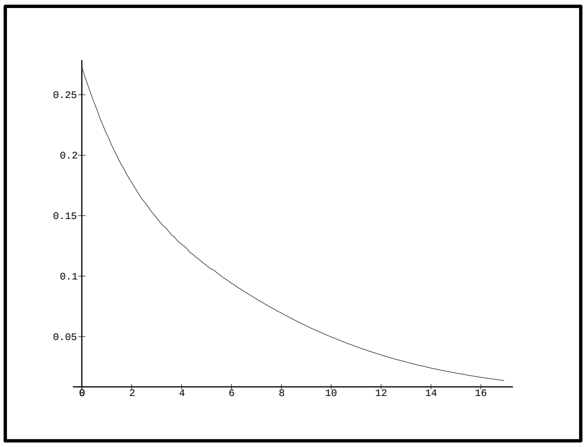

Figure 3: Norm of global error versus time

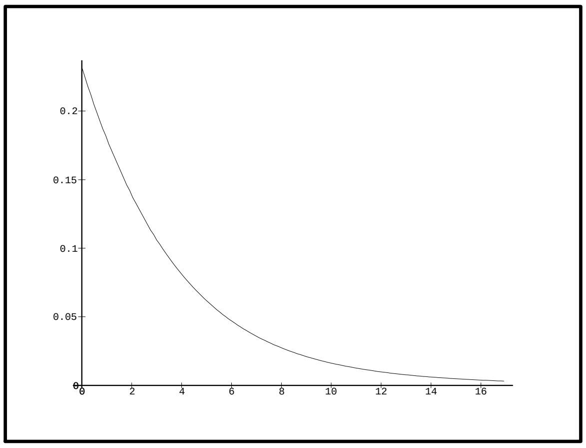

Figure 4: ζ signal versus time

Figure 5: Overview of the image processing hardware