

Real Time Visual Servoing around a Complex Object

François BERRY[†], Philippe MARTINET[†], and Jean GALLICE[†], *Nonmembers*

SUMMARY In visual servoing, most studies are concerned with robotic application with known objects. In this paper, the problem of controlling a motion by visual servoing around an unknown object is addressed. In this case, the approach is interpreted as an initial step towards a perception goal of an unmodeled object. The main goal is to perform motion with regard to the object in order to discover several viewpoint of the object. An adaptive visual servoing scheme is proposed to perform such task. The originality of our work is based on the choice and extraction of visual features in accordance with motions to be performed. The notion of invariant feature is introduced to control the navigational task around the unknown object. During experimentation, a cartesian robot connected to a real time vision system is used. A CCD camera is mounted on the end effector of the robot. The experimental results present a linkage of desired motion around different kind of objects.

key words: *visual servoing, complex object, real time processing, robot manipulator*

1. Introduction

Most robotic vision works deal with the scenic perception of spatial geometry taken from a mobile camera. These systems come within the framework of active vision. However, the perception of an object may be achieved at many different scales. The highest level of perception is the recognition of the object, its relation to the perceiver and the action to perform on the object [1], [2]. Another approach consists in evaluating the structure of the object during navigation. In this way, most techniques are based on the “structure from motion” approach and the use of optical flow. But in these methods, a choice has to be made between the complexity of the scene [6] and time computing [20]. Our goal is to perceive an unknown object in order to perform a motion around it. Consequently, a coarse knowledge of the object is required. Perception consists in evaluating the pose and the space filled by the object. Based on this point of view, the main idea is to approximate the object within a bounding box. This approximation allows us to simplify the motions around the object.

In a previous work [4], [5], an approach to generating a motion around a known object (cube) was presented. This approach was based on a visual servoing

technique applied to a time varying reference feature. The reference in the sensor frame was computed according to the desired trajectory in robot workspace.

For complex scenes, other works propose an automatic selection of visual features (edge, corner, ...). Papanikolopoulos in [17] used a method based on a SSD optical flow technique. This technique may fail however when the image contains a lot of repeated patterns of the same intensity and is also sensitive to large rotations and small changes in lighting. In [11], the authors propose an approach based on geometric constraints. These are imposed by the feature extraction (type of features, size, number, ...) and the pose estimation process (field of view, focus, ...). In this strategy, the trajectory should be approximately known in order to perform a good selection of image features.

In most cases, an initial learning step is necessary to obtain information characterizing the interaction between the apparatus sensor and the environment (Eigen space method [7], [22], image jacobian [10], [21]). So, the proposed method [3] is to perform automatically motions around an unmodeled object in order to learn this interaction.

In the first part of the paper, the modeling aspect is particularly developed: discussion on the use of different visual features and kinematic modeling for the most important feature are presented. In the second part, the control aspect is described from a theoretical basis. Its application has been retained to complete motions around an unknown object. The visual servoing process uses adaptive principles as the computation of an invariant feature. Finally, results obtained at video rate with our robotic platform and parallel vision system show the validity of this approach.

2. Modeling

2.1 Visual Features

Generally speaking in Mechanics, it is usual to parameter objects in order to apply the kinematics and the dynamics of rigid bodies. For static bodies, chosen parameters allow one to get a description of the objects such as the mass, length, height, position, orientation, ... and with this data, answers can be given to questions like:

“Is it a big object ?”

Manuscript received October 12, 1999.

Manuscript revised March 3, 2000.

[†]The authors are with Laboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique, Université Blaise Pascal de Clermont Ferrand, UMR 6602 du C.N.R.S, F-63177 Aubière Cedex, France.

“Is it large or long ? ”

“What is its disposition?,” etc ...

In this way, the fact that a flower pot is red or blue does not change its mechanical properties. If somebody wants to grasp it, it is more important to know the position and the size than the color. Consequently, the texture and the shading are not taken into account in this study. Indeed, these kinds of measures are very sensitive to the lighting conditions and what imposes hypotheses of simplification (Lambertian surface, constrained light, ...). These hypotheses are not realistic in real robotic application, whereas geometrical descriptions of the object remain well adapted [8], [13].

From the image of an object, it is possible to extract the following information in order to control robot motion:

- **Position,**
- **Orientation,**
- **Size.**

For each of the above, geometrical features can be brought together enabling action around the object. In our case, the perception sensor is a gray scale level CCD camera mounted on the end effector of a manipulator. Thus, all measures are calculated in the image plane. As explained above, neither the photogrametric variations nor the texture parameters need to be considered. Consequently, the image of the object is reduced to a binary image such:

$$\begin{cases} P_{i,j} = 1 & \text{if } P_{i,j} \in \text{object} \\ P_{i,j} = 0 & \text{if } P_{i,j} \notin \text{object} \end{cases}$$

where $P_{i,j}$ is a pixel in a column i and the row j .

In this context, there are three visual features selected.

- Initially, one would think that an efficient way of representing the position of an object in image space would be to locate its centroid. However, in several cases the shape is not necessarily convex (Figs.1 (b) and 1 (c)) or uniform (Fig.1 (d)) and thus the centroid would not be the best feature to quantify the position of the object. For example, the object in Fig.1 (c) is very dissymetrical and a centering task could not be well performed (Fig. 2). For these reasons, the center of the bounding box which frames the object in the image as a centering position has been chosen. This approach has three advantages: (i) the bounding box is quickly extracted from the contour of the object, (ii) the positioning of the bounding box is quite easy to perform in image space, and (iii) there is no preponderance in function of the shape of the object (Fig. 3).
- A second visual feature concerns the distance between the camera and the object. To control this distance, a geometric feature varying in function

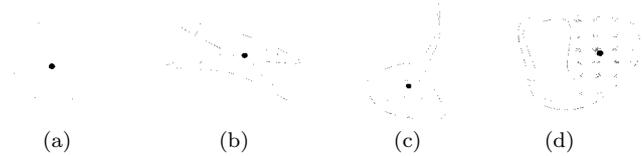


Fig. 1 Centroid of different shape.

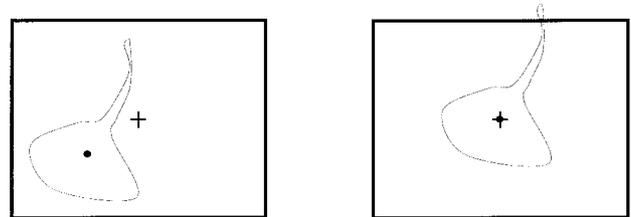


Fig. 2 Centering of a dissymmetric shape.

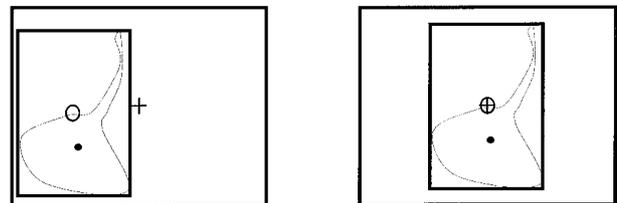


Fig. 3 Centering of a dissymmetric shape with a bounding box.

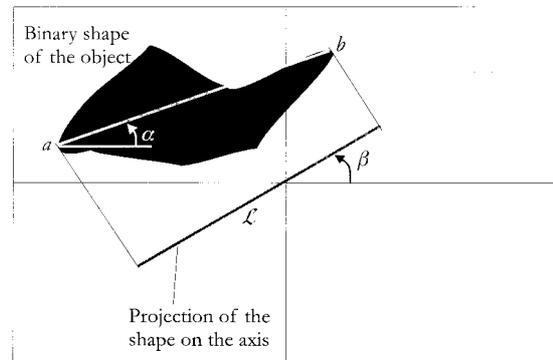


Fig. 4 Projection of a binary shape on an axis Δ .

of depth is necessary: this is the case for the projection \mathcal{L} of the shape on an axis Δ (Fig. 4) in image space. This segment \mathcal{L} represents the projection of a segment \mathcal{S} (function of the object) on Δ^\dagger . From the length of the segment \mathcal{L} , it is possible to control the distance between the camera and the object. The projection axis Δ is centered in the image frame and makes an angle β with the abscissa axis. So, the coordinates of the projected shape on Δ are given by:

$$\begin{aligned} x_{proj} &= (x \cdot \cos \beta + y \cdot \sin \beta) \cdot \cos \beta \\ y_{proj} &= (x \cdot \cos \beta + y \cdot \sin \beta) \cdot \sin \beta \end{aligned}$$

[†]The choice of the axis orientation β will be discussed in paragraph 3.2.

where (x, y) represent the coordinates of binary shape points and (x_{proj}, y_{proj}) are the coordinates of their projection on Δ . In other words, $(x_{proj}, y_{proj}) = Proj(x, y)|_{\Delta}$.

- In order to control orientation, the moment of inertia applied to the shape in the image is well adapted. If I_x, I_y represent respectively the x, y moment of inertia and I_{xy} the product of inertia, the orientation of the shape is deduced with the well-known formula:

$$\delta = -\frac{1}{2} \arctan \left(\frac{2I_{xy}}{I_x - I_y} \right)$$

where δ is the angle between the principal axis and the abscissa axis.

2.2 Kinematic Modeling

For each visual feature \underline{s} described above, it is possible to model their variation $\dot{\underline{s}}$ in function of the camera motion T through the relation:

$$\dot{\underline{s}} = M_{\underline{s}}^T \cdot T$$

where T represents the kinematic screw applied to the sensor and $M_{\underline{s}}^T$ the jacobian matrix (called interaction matrix) relative to the sensor feature \underline{s} . In this sequel, the development of each interaction matrix is then presented.

- With the bounding box around the object (in image space), the first robotic task consists in centering the object within the image space. This means performing a positioning task for a point feature m at coordinates $\underline{x} = (x \ y \ z)^T$. The projection of m in the image space is called M ($\underline{X} = (X \ Y)^T$). The corresponding interaction matrix M_M^T (with a unit focal length) is well known [9]:

$$M_M^T = \begin{pmatrix} -\frac{1}{z} & 0 & \frac{X}{z} & XY & -1 - X^2 & Y \\ 0 & -\frac{1}{z} & \frac{Y}{z} & 1 + Y^2 & -XY & -X \end{pmatrix} \tag{1}$$

- The second visual feature is the projection \mathcal{L} of a segment \mathcal{S} of the object on one axis Δ . The segment \mathcal{S} can be represented with the vector $\underline{P_S}$, and \mathcal{L} with $\underline{P_{\mathcal{L}}}$ such as:

$$\underline{P_S} = \begin{pmatrix} X_S \\ Y_S \\ L_S \\ \alpha \end{pmatrix} \quad \text{et} \quad \underline{P_{\mathcal{L}}} = \begin{pmatrix} X_{\mathcal{L}} \\ Y_{\mathcal{L}} \\ L_{\mathcal{L}} \end{pmatrix}$$

where (X_S, Y_S) (resp. $(X_{\mathcal{L}}, Y_{\mathcal{L}})$) is the middle of \mathcal{S} (resp. \mathcal{L}), and L_S (resp. $L_{\mathcal{L}}$) is the corresponding length. The relations between (\mathcal{S}) and (\mathcal{L}) are easily

obtained:

$$\begin{cases} X_{\mathcal{L}} = (X_S \cdot \cos \beta + Y_S \cdot \sin \beta) \cdot \cos \beta \\ Y_{\mathcal{L}} = (X_S \cdot \cos \beta + Y_S \cdot \sin \beta) \cdot \sin \beta \\ L_{\mathcal{L}} = L_S \cdot \cos(\alpha - \beta) \end{cases} \tag{2}$$

The expression of the interaction matrix M_S^T for the segment \mathcal{S} is given in appendix. It is possible to obtain the interaction matrix $M_{\mathcal{L}}^T$ of the projection \mathcal{L} using the following relation:

$$M_{\mathcal{L}}^T = \frac{\partial \underline{P_{\mathcal{L}}}}{\partial \underline{P_S}} \cdot M_S^T$$

where $\frac{\partial \underline{P_{\mathcal{L}}}}{\partial \underline{P_S}}$ is expressed by:

$$\begin{pmatrix} \cos^2 \beta & \sin \beta \cos \beta & 0 & 0 \\ \sin \beta \cos \beta & \sin^2 \beta & 0 & 0 \\ 0 & 0 & \cos(\beta - \alpha) & L_S \sin(\beta - \alpha) \end{pmatrix}$$

In the matrix M_S^T , only the sub-matrix corresponding to the length of the projection $L_{\mathcal{L}}$ on Δ is considered, and then the related interaction sub-matrix $M_{L_{\mathcal{L}}}^T$ is given by:

$$\begin{aligned} M_{L_{\mathcal{L}}}^T [1, 1] &= \nu_1 \cdot \cos \beta \\ M_{L_{\mathcal{L}}}^T [1, 2] &= \nu_1 \cdot \sin \beta \\ M_{L_{\mathcal{L}}}^T [1, 3] &= \nu_2 \cdot L_{\mathcal{L}} - \nu_1 \cdot X_S \cdot \cos \beta - \nu_1 \cdot Y_S \cdot \sin \beta \\ M_{L_{\mathcal{L}}}^T [1, 4] &= L_{\mathcal{L}} (X_S \cdot \cos \alpha \cdot \sin \alpha + Y_S (1 + \sin^2 \alpha) \\ &\quad + \tan(\beta - \alpha) \cdot (-X_S \cdot \sin^2 \alpha \\ &\quad + Y_S \cdot \cos \alpha \cdot \sin \alpha)) \\ M_{L_{\mathcal{L}}}^T [1, 5] &= -L_{\mathcal{L}} (Y_S \cdot \cos \alpha \cdot \sin \alpha + \tan(\beta - \alpha) \cdot \\ &\quad (-X_S \cdot \cos \alpha \cdot \sin \alpha + Y_S \cdot \cos^2 \alpha) \\ &\quad + X_S (\cos^2 \alpha + 1)) \\ M_{L_{\mathcal{L}}}^T [1, 6] &= -L_{\mathcal{L}} \cdot \tan(\beta - \alpha) \end{aligned} \tag{3}$$

where $\nu_1 = \frac{z_a - z_b}{z_a z_b}$ and $\nu_2 = \frac{z_a + z_b}{2z_a z_b}$ (z_a and z_b represents the depth of the points a and b in Fig. 4).

- The last feature is the orientation of the binary shape. This orientation can be modeled from the principal axis of inertia. It corresponds to the axis of the best fitting ellipse on the shape. So, for the interaction

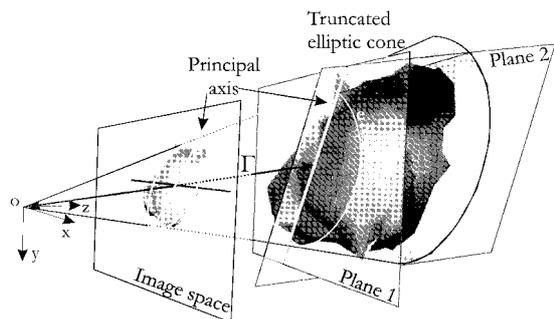


Fig. 5 Orientation of an object.

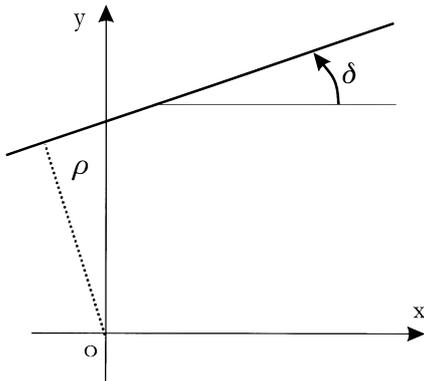


Fig. 6 δ and ρ parameters.

matrix, it is necessary to establish a relation between the visual feature E and the corresponding tridimensional feature e.

In our case, the feature E is considered as the projection of the section e of a truncated elliptic cone fitted around the object (Fig. 5). The distance between this section and the camera is noted Γ and the orientation of the ellipse is noted δ . The principal axis of the ellipse e is defined by the intersection of two orthogonal planes (1 and 2). They can be defined by:

$$\begin{cases} \text{Plane 1: } c_1 \cdot z + d_1 = 0 \\ \text{Plane 2: } a_2 \cdot x + b_2 \cdot y + c_2 \cdot z + d_2 = 0 \end{cases} \quad (4)$$

where plane 1 is always parallel to the image plane. Thus the orientation of e is identical with the orientation of E. The projection of the principal axis can be parameterized in image space by:

$$A \cdot X + B \cdot Y + C = 0$$

where

$$\begin{cases} A = -a_2 d_1 \\ B = -b_2 d_1 \\ C = c_1 d_2 - c_2 d_1 \end{cases}$$

Polar coordinates (δ, ρ) (Fig. 6) have been retained as the parameters for this axis. Thus, the interaction matrix M_δ^T of the parameter δ is given by [12]:

$$\begin{pmatrix} \nu \cdot \sin \delta & -\nu \cdot \cos \delta & \nu \cdot \rho & -\rho \cdot \sin \delta & \rho \cdot \cos \delta & -1 \end{pmatrix}$$

with $\nu = (a_2 b_1 - a_1 b_2) / \sqrt{A^2 + B^2}$. In Eq. (4), the plane 1 is parallel to the image plane so $a_1 = b_1 = 0$ and so $\nu = 0$. The interaction matrix becomes:

$$M_\delta^T = \begin{pmatrix} 0 & 0 & 0 & -\rho \cdot \sin \delta & \rho \cdot \cos \delta & -1 \end{pmatrix} \quad (5)$$

At the equilibrium, when the object is centered, the parameter ρ is close to 0 ($\rho \approx 0$).

2.3 Limits of These Visual Features

In this paragraph, the validity of visual features is discussed. In our approach, the configuration of the object

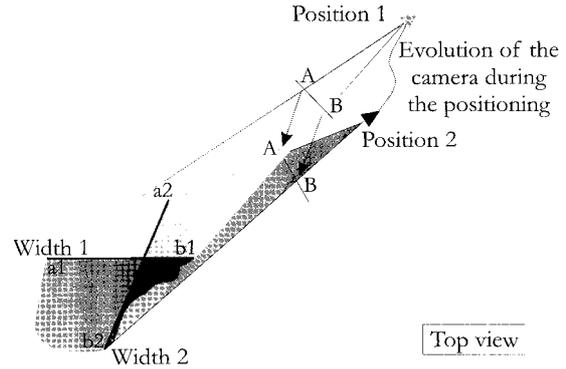


Fig. 7 Evolution of the width during a positioning task.

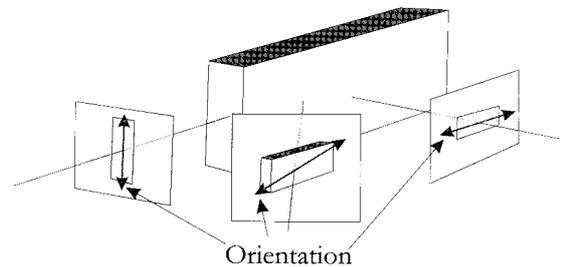


Fig. 8 Variation of an axis of inertia around an elongated box.

is unknown, so the chosen parameterization of the feature is approximated.

For example, during a positioning task the camera can “discover” other viewpoints of the object and segment \mathcal{S} has no physical reality. Considering the width of a particular section of the object (Fig. 7), segment AB (in image) corresponds indifferently to segment $a_1 b_1$ or $a_2 b_2$ in function of the camera position. It is important to know that in spite of the discontinuities between points of view 1 and 2, evolution of AB is smooth and continuous. The main reason is that each extremity (A and B) has a limited position such:

$$x_A = \frac{x_{a1}}{z_{a1}} = \frac{x_{a2}}{z_{a2}} \quad \text{and} \quad x_B = \frac{x_{b1}}{z_{b1}} = \frac{x_{b2}}{z_{b2}}$$

Another limit is important for the regulation of the object orientation. As mentioned above, during a positioning task, the orientation of the axis of inertia can vary from 0° to 90° ! It is the case for an elongated object (Fig. 8). However, during a positioning task the variation of the axis of inertia is small in comparison to a navigational task. For this reason, orientation is only regulated during a positioning task.

3. Control and Visual Servoing Process

In this section, control law and the visual servoing process are developed. First, the fundamental basis concerning the *Task function approach* [19] is summarized, then the development in relation to our application is presented. Lastly, the visual servoing process is devel-

oped.

3.1 The Task Function Approach

The control law used in this study is based on the *Task function* formalism [19]. In this approach, the control is directly specified in terms of regulation in the image. It may be noted that this approach has the advantage of avoiding the intermediate step of the 3D estimation of the target with regard to the end effector [15], [24]. For a given robotics task, a *target image* is built, corresponding to the desired position of the end effector with regard to the environment. It can be shown that all servoing schemes may, in general, be expressed as the regulation to zero of a function $\underline{f}(\underline{r}, t)$ called *the task function*. So, the use of a vision sensor allows us to build up such a task function used in visual servoing. It is expressed by the relation:

$$\underline{f}(\underline{r}, t) = C[\underline{s}(\underline{r}, t) - \underline{s}^*] \quad (6)$$

where

- \underline{s}^* is considered as a reference target image to be reached in the image frame.
- $\underline{s}(\underline{r}, t)$ is the value of visual information currently observed by the camera. This information depends on the situation between the end effector of the robot and the scene (noted \underline{r}).
- C is a constant matrix, with which it is possible to take into account more visual information than the number of degrees of freedom of the robot, with good conditions of stability and robustness.

The variations of $\underline{f}(\underline{r}, t)$ are given by the following differential relation:

$$\frac{d\underline{f}(\underline{r}(t), t)}{dt} = \frac{\partial \underline{f}}{\partial \underline{r}} \cdot \frac{d\underline{r}}{dt} + \frac{\partial \underline{f}}{\partial t} = C \frac{\partial \underline{s}}{\partial \underline{r}} \cdot \frac{d\underline{r}}{dt} + \frac{\partial \underline{f}}{\partial t} \quad (7)$$

where $\frac{d\underline{r}}{dt} = T = (\vec{V}, \vec{\Omega})$ is the kinematic screw. T represents the relative velocity between the camera and its environment and the term $\frac{\partial \underline{s}}{\partial \underline{r}} = M^T$ called interaction matrix or image jacobian, characterizes the interaction between the sensor and its environment. The concept of interaction matrix is fundamental for modeling systems using an exteroceptive sensor. It allows one to take into account most information required to design and analyze sensor based control schemes.

If the image jacobian is not full rank (number of d.o.f > number of independent visual features), it is possible to use an hybrid task. In an hybrid task, the primary task \underline{e}_1 [†] allows one to maintain a visual constraint during the trajectory, while the secondary task \underline{e}_2 can be seen as representing a minimization of a secondary cost h_s with the gradient $\underline{g}_s^T = (\frac{\partial h_s}{\partial \underline{r}})^T$.

A global *task function* \underline{e} takes the form:

$$\underline{e} = W^+ \underline{e}_1 + \gamma \cdot (I_n - W^+ W) \underline{g}_s^T \quad (8)$$

where W^+ and $(I_n - W^+ W)$ are two projection operators which guarantee that the camera motions due to the secondary task are compatible with the regulation of \underline{s} to \underline{s}^* . W is a full rank matrix such as $Ker(W) = Ker(M^T)$. The parameter γ is used to tune the preponderance between the primary and the secondary task.

Considering an exponential decay of $\dot{\underline{e}}(\underline{r}, t)$:

$$\dot{\underline{e}}(\underline{r}, t) = -\lambda \underline{e}(\underline{r}, t) \quad (9)$$

with λ a positive scalar constant. In applying relation 7 to the global task function \underline{e} , the kinematic screw can be expressed with:

$$T = - \left(\frac{\partial \underline{e}}{\partial \underline{r}} \right)^{-1} \left(\lambda \underline{e} + \frac{\partial \underline{e}}{\partial t} \right) \quad (10)$$

To ensure the stability of the system, the following condition

$$\left(\frac{\partial \underline{e}}{\partial \underline{r}} \right) \cdot \left(\widehat{\frac{\partial \underline{e}}{\partial \underline{r}}} \right)^{-1} > 0 \quad (11)$$

must be verified [19]. This is done when the combination matrix C is fixed to $W.M^{T+}$. In addition, the previous condition is always verified when choosing $\left(\frac{\partial \underline{e}}{\partial \underline{r}} \right)^{-1} = \mathbb{I}_6$.

Considering a motionless environment, it gives $\frac{\partial \underline{s}}{\partial \underline{r}} = 0$ and $\frac{\partial \underline{e}_1}{\partial \underline{r}} = 0$. Finally, from the relations 8 and 10, the control law has the following expression:

$$T = -\lambda \underline{e}(\underline{r}, t) - \gamma (I_n - W^+ W) \frac{\partial \underline{g}_s^T}{\partial t} \quad (12)$$

3.2 Moving around an Unknown Object

In this subsection, the general control law (Eq. (12)) is adapted in order to *move around an unknown object*. The control laws for a positioning task and a navigational task are presented here.

- For the positioning task, the four visual features (position, orientation and length of the projection) described in paragraph 2.2 can be used ($\underline{s}(\underline{r}, t) = (X \ Y \ L_{\mathcal{L}} \ \delta)^T$). The equilibrium configuration is reached when the shape is centered in image, and when the axis of inertia is aligned with the image frame. The distance between the camera and the object is specified as an expression: “height of object = $x\%$ size of image” or more generally “ $L_{\mathcal{L}}^* = n$ pixels.” As a result, the equilibrium position is characterized by:

$$\begin{aligned} X = Y = 0 &\Rightarrow \rho \approx 0 \\ z_a = z_a^*, z_b = z_b^* &\Rightarrow \nu_1^* = \frac{z_a^* - z_b^*}{z_a^* \cdot z_b^*}, \nu_2^* = \frac{z_a^* + z_b^*}{2 \cdot z_a^* \cdot z_b^*} \\ z = z^*, \delta = 0, L_{\mathcal{L}} &= L_{\mathcal{L}}^* \end{aligned}$$

[†]The primary task is built from the visual information as written in Eq. (6).

For experimentation, z_a^* and z_b^* are approximated to z^* , thus $\nu_1^* \approx 0$ and $\nu_2^* \approx \frac{1}{z^*}$.

The global interaction matrix $M_{|\underline{s}=\underline{s}^*}^T$ is defined (at equilibrium) by:

$$\begin{array}{c} \begin{array}{cccccc} V_x & V_y & V_z & \Omega_x & \Omega_y & \Omega_z \end{array} \\ \hline \begin{pmatrix} -1/z^* & 0 & 0 & 0 & -1 & 0 \\ 0 & -1/z^* & 0 & 1 & 0 & 0 \\ \nu_1^* \cdot \cos \beta & \nu_1^* \cdot \sin \beta & \nu_2^* \cdot L_{\mathcal{L}}^* & L_{\mathcal{L}}^* \cdot Y_S^* & 0 & L_{\mathcal{L}}^* \cdot \tan(\alpha^*) \\ 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \end{array}$$

Near equilibrium only four d.o.f are necessary to perform this positioning task ($\dim(Ker(M_{|\underline{s}=\underline{s}^*}^T)) = 2$). In our application, the rotations around x , y and z axis and the translation along z axis have been retained. For the following development, the corresponding reduced matrix $M_{R|\underline{s}=\underline{s}^*}^T$ is extracted from $M_{|\underline{s}=\underline{s}^*}^T$:

$$M_{R|\underline{s}=\underline{s}^*}^T = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ \nu_2^* \cdot L_{\mathcal{L}}^* & L_{\mathcal{L}}^* \cdot Y_S^* & 0 & L_{\mathcal{L}}^* \cdot \tan(\alpha^*) \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

In this expression, the terms α^* , Y_S^* are functions of the observed object and are not regulated, so some approximations to simplify $M_{R|\underline{s}=\underline{s}^*}^T$ are proposed. First, from the orientation task, the object can be considered nearly horizontal, so $\alpha \approx 0$. A second approximation concerns the term Y_S^* . The centering task brings the shape to the image center, so $Y_S^* \approx 0$. The interaction matrix is square and full rank, thus $W = \mathbb{I}_4$. From these approximations and with the convergence condition ($C = M_{R|\underline{s}=\underline{s}^*}^{T-1}$) (Eq. (11)), a control law can be defined such as:

$$\begin{pmatrix} T_z \\ R_x \\ R_y \\ R_z \end{pmatrix} = \lambda \cdot \begin{pmatrix} -\frac{1}{\nu_2^* \cdot L_{\mathcal{L}}^*} \cdot (L_{\mathcal{L}} - L_{\mathcal{L}}^*) \\ -Y \\ X \\ \delta \end{pmatrix} \quad (13)$$

where λ is the gain of the exponential decay. Several works [3], [4], [6], [8], [9], [13], [15] have shown that the use of the interaction matrix computed at equilibrium allows the regulation of the visual task. In addition, it avoids the singularities during servoing when computing the inverse of the image jacobian.

• For a navigational task around an unknown object, it is necessary to use an hybrid task composed of:

- a primary task, where the goal is to gaze at the object, to center it in the sensor frame and to hold a constant distance between the camera and the object.
- a secondary task which generates the translation along the X and Y axis.

As explained in Sect. 2.3, it is impossible to regulate the orientation during a navigational task. For this reason, the visual feature is modeled by:

$$\underline{s}(\underline{r}, t) = \begin{pmatrix} X \\ Y \\ L_{\mathcal{L}} \end{pmatrix}$$

With such features, only 3 d.o.f can be controlled (i.e. R_x, R_y, T_z), and 2 d.o.f are needed for the navigational task (i.e. T_x, T_y). So, only the reduced system to these d.o.f is considered, and then the interaction matrix associated to $\underline{s}^*(\underline{r}, t)$ becomes:

$$M_{R|\underline{s}=\underline{s}^*}^T = \begin{pmatrix} -\frac{1}{z} & 0 & 0 & 0 & -1 \\ 0 & -\frac{1}{z} & 0 & 1 & 0 \\ \nu_1^* \cdot \cos \beta & \nu_1^* \cdot \sin \beta & \nu_2^* \cdot L_{\mathcal{L}}^* & 0 & 0 \end{pmatrix}$$

From the kernel of $M_{R|\underline{s}=\underline{s}^*}^T$, the motions allowed by the interaction can be given by:

$$Ker(M_{R|\underline{s}=\underline{s}^*}^T) = \left\{ \begin{pmatrix} 1 & 0 & \frac{-\nu_1^* \cdot \cos \beta}{\nu_2^* \cdot L_{\mathcal{L}}^*} & 0 & -\frac{1}{z} \\ 0 & 1 & \frac{-\nu_1^* \cdot \sin \beta}{\nu_2^* \cdot L_{\mathcal{L}}^*} & \frac{1}{z} & 0 \end{pmatrix} \right\}$$

Considering a motion around the object such as $\underline{T} = \begin{pmatrix} A \cdot \cos \theta \\ A \cdot \sin \theta \end{pmatrix}$, the general form of the allowed camera motion is:

$$T = \begin{pmatrix} T_x = A \cdot \cos \theta \\ T_y = A \cdot \sin \theta \\ T_z = A \cdot \frac{-\nu_1^* \cdot (\cos \beta \cdot \cos \theta + \sin \beta \cdot \sin \theta)}{\nu_2^* \cdot L_{\mathcal{L}}^*} \\ R_x = A \cdot \frac{\sin \theta}{z} \\ R_y = -A \cdot \frac{\sin \theta}{z} \end{pmatrix}$$

This motion is composed of a combination of translation and rotation along the x and y axis. However, the translation along the optical axis (T_z) is not null. In other words, the decoupling of T_z is only done when $\cos \beta \cdot \cos \theta + \sin \beta \cdot \sin \theta = 0$. This condition is obtained for $\theta = \beta + \frac{\pi}{2}$, so the orientation of the axis Δ must be orthogonal to the motions around the object (projected

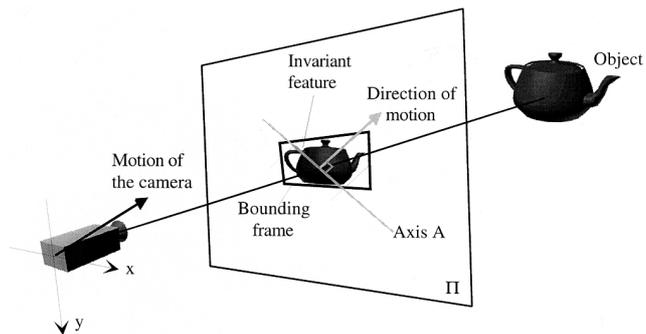


Fig. 9 Orientation of axis Δ in comparison to the camera motions.

in the image space) (Fig. 9). The task function can be written like:

$$\underline{e} = M_{R|\underline{s}=\underline{s}^*}^{T+}(\underline{s} - \underline{s}^*) + \gamma(\mathbb{I}_5 - M_{R|\underline{s}=\underline{s}^*}^{T+} M_{R|\underline{s}=\underline{s}^*}^T) \underline{g}_s^T$$

where $(\mathbb{I}_5 - M_{R|\underline{s}=\underline{s}^*}^{T+} M_{R|\underline{s}=\underline{s}^*}^T)$ is an orthogonal projector. Then, the control law is given by:

$$T = -\lambda \underline{e} - \gamma(\mathbb{I}_5 - M_{R|\underline{s}=\underline{s}^*}^{T+} M_{R|\underline{s}=\underline{s}^*}^T) \frac{\partial g^T}{\partial t} \quad (14)$$

In our case, the secondary cost function h_s is defined by:

$$h_s = \frac{1}{2}(x - x_o - V_x t)^2 + \frac{1}{2}(y - y_o - V_y t)^2$$

where (x, y) represents the position of the camera, (x_0, y_0) is the initial position (in our case $(x_0, y_0) = (0, 0)$) and (V_x, V_y) is the velocity of the camera used for the navigation. In other words, the velocity (V_x, V_y) allows one to describe the motion around the object. For example a vertical motion on top at 0.1 m.s^{-1} is achieved for $V_x = 0$ and $V_y = -0.1 \text{ m.s}^{-1}$. The gradient of this cost function is given by \underline{g}_s^T is:

$$\underline{g}_s^T = \begin{pmatrix} (x - x_o - V_x t) \\ (y - y_o - V_y t) \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

3.3 Visual Servoing Process

This subsection describes the visual servoing process with regard to the positioning or navigational task to perform. Figure 10 illustrates the visual servoing process: actions, vision and control processes. In function of the actions to be performed, both vision and control processes have to be adapted. In the vision process, this is the visual features extraction, and in the control process this is the sensor vector and the corresponding interaction matrix.

For all kinds of tasks, an area of interest is defined to reduce the execution time. From this area of interest, a bounding box which frames the unknown object is built, and the coordinate of the center and the size of this box is computed. In the case of a navigational task, an invariant feature in the direction of the motion has to be built. In our application, the length of a particular segment is chosen (Sect. 2.1). First, knowing the direction of the motion an axis of projection Δ is defined in image space as orthogonal to this direction of motion. Second, the segment is obtained by projection of the shape on this axis and the length is computed (Fig. 11).

Finally, as a navigational task imposes movement around an object, the vision process can be affected

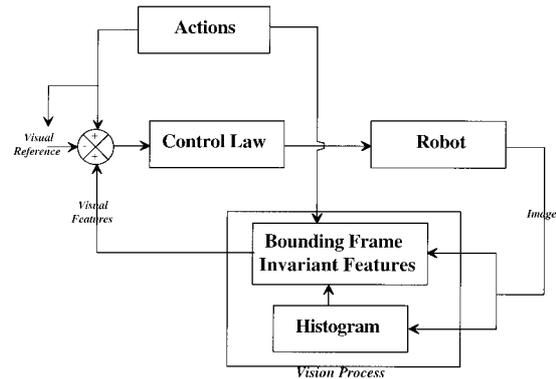


Fig. 10 Global scheme of our application.

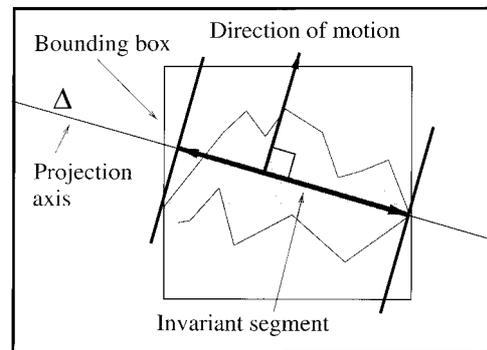


Fig. 11 Invariant feature construction.

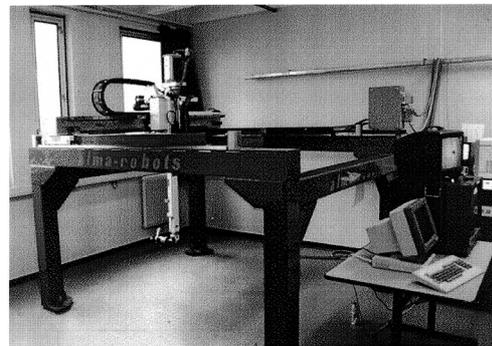


Fig. 12 Robotic platform.

by the different lighting conditions encountered during this movement. So, it is necessary to adapt some characteristics. Particularly, the low level extraction of the bounding box needs to adapt the thresholds according to the lighting conditions. So, at each iteration an histogram is computed and all thresholds are modified using the main results found in [23].

4. Results

4.1 Overview of the Robotic Platform

Our experimental cell is composed of a cartesian robot with 6 d.o.f (Fig. 12). A CCD camera is embedded on the end effector and is connected to the vision parallel

architecture Windis [16], [18].

Windis architecture uses the concept of active windows [14] and includes three basic modules.

WINDIS Window Distributor Subsystem is used for window extraction, the execution of low level processing and the distribution of active windows toward the Window Processing Subsystem.

On WINPROC Window Processing Subsystem, one to sixteen DSP 96002 modules with one distributor module are associated. DSP modules are put together on mother boards and execute medium level processing on windows.

WINMAN Window Manager Subsystem controls the distributor and DSP modules, and executes high level processing of the application tasks. Moreover, it is used for the tracking of the active windows throughout the sequence and for the command of the robot. A 68040 based cpu board implements this module.

The management of the system is ensured under VxWorks Real Time Operating System.

The vision process has been implemented on the Windis architecture. On the low level board, the grey levels and a list of selected pixels corresponding to the highest gradient are extracted. Two DSP modules are used in parallel: the first DSP computes the histogram of grey levels and the adaptive thresholds, and the second DSP extracts the bounding box of the object and the invariant segment. The window manager manages the visual tracking and the adaptive processes along the sequence. All of this implementation is made at video rate (40 ms).

4.2 Experimental Results

This subsection presents experimental results obtained with our robotic platform. Both positioning navigational tasks have been tested.

- Positioning task

For this experiment, the object is a multicolor “toy car.” The positioning task consists in centering the

object with a null orientation at a given distance. The latter is defined by an apparent height of the object in image space (fixed to 20% of image size (102 pixels)). The gain λ of the control law is tuned to 0.5 and the parameter z^* is arbitrarily fixed to 0.7 m. Figure 13 shows the different images obtained during the servoing process. At the end, the object is centered and its orientation decreases to zero[†].

[†]It is important to remark that the natural axis (such width, height, ...) is different to the principal axis.

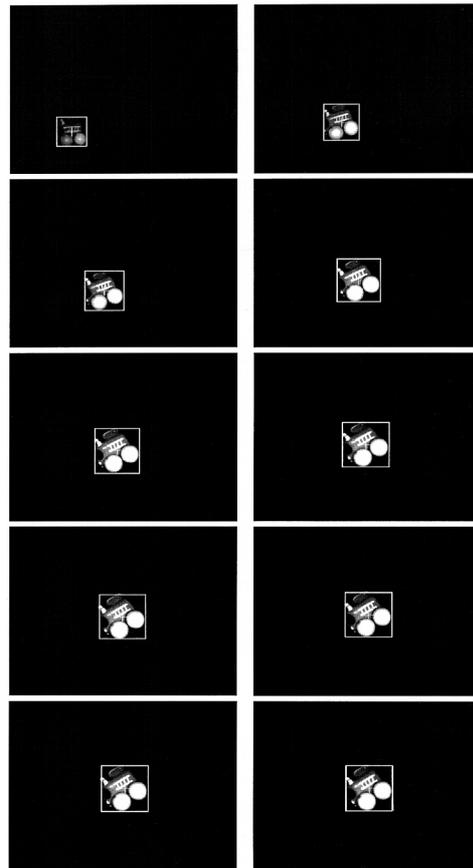


Fig. 13 Evolution of the object during servoing task.

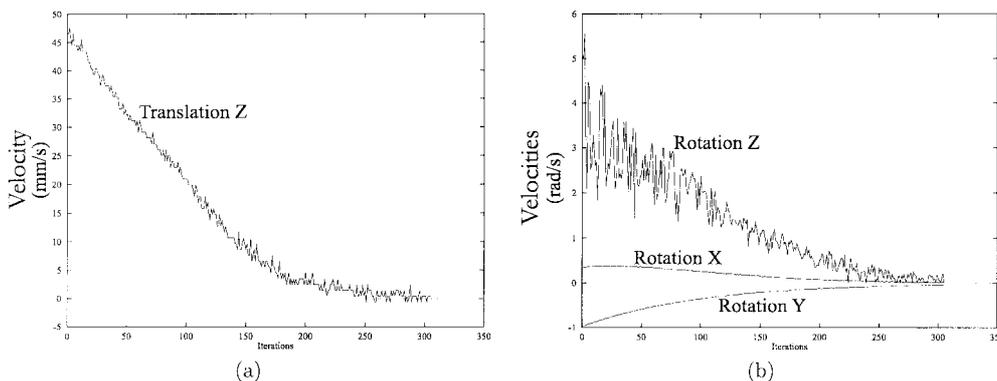


Fig. 14 Translation and rotation velocities.

Figure 14 presents the velocities of the effector. Every velocity (T_z, R_x, R_y, R_z) has an exponential decay. Though, it is interesting to denote that the perturbations on the rotation axis z is due to the noisy measure of orientation.

- Navigational task

For the navigational task, two motions are performed around a little rubber giraffe. Considering the object centered at a given distance (Height of object = 30% of image size (170 pixels)), the camera moves to the left side while keeping the object centered and then rises above the object (Fig.15). For the control law, the parameters are $\lambda = 0.8$, $\gamma = 1.0$ and $z^* = 0.7$ m. These parameters are tuned experimentally in accordance with the task to perform. The velocities applied to the effector are $T_x = -0.08$ m.s⁻¹ and $T_y = -0.08$ m.s⁻¹. The rotation axis (in image) is respectively the height and the width of the bounding frame. The visual reference feature is chosen from the last measure during the previous motion. Such choice allows one to keep the same distance for both motions. Figure 16 represents the evolution of the object during the servoing and Fig. 17 presents the velocity of the kinematic screw. The servoing task is composed of three steps. The first step concerns the positioning task, second and third steps – the navigational tasks. Velocities become noisy during navigational task and particularly noisier during the third step. In the

latter, the width of the object is used instead of the height (used in the second step), and in our implementation, the discretization in x and y direction are not the same: in vertical direction, only one frame is used. The image is “compressed” in this direction and there is a filtering effect along this direction. The motions around the object are performed after the positioning task (Fig. 18 (b)). The choice of the desired distance z^* between the camera and the object is very important and determines the achievement of the secondary task.

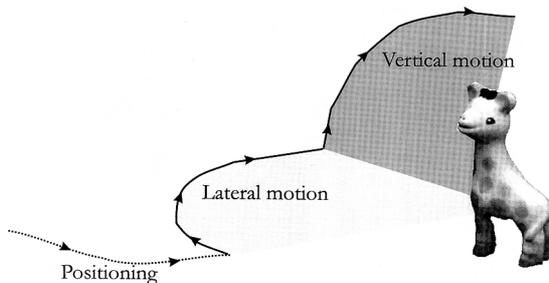


Fig. 15 Trajectory around the giraffe.

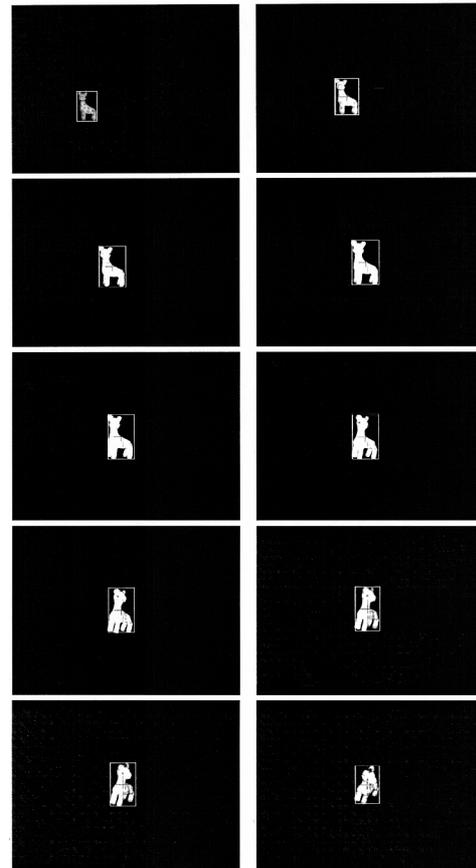


Fig. 16 Evolution of the objet during the navigation.

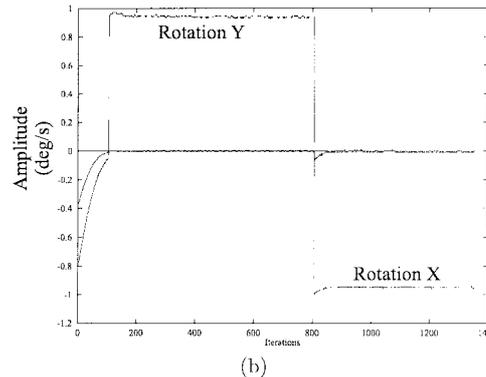
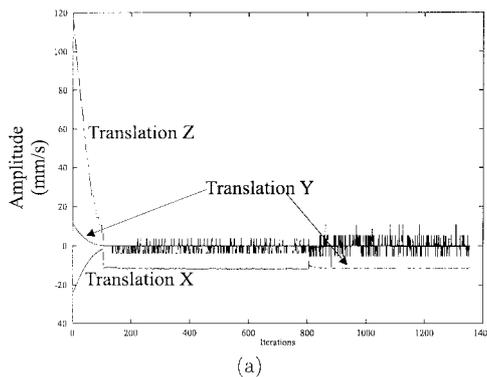


Fig. 17 Translation and rotation velocities.

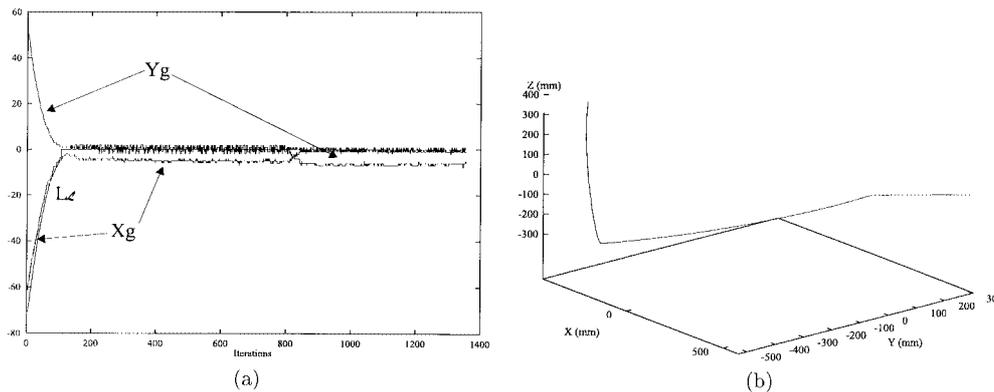


Fig. 18 (a) Error on features. (b) 3D trajectory of the effector.

Though this distance cannot be measured and is set arbitrarily, thus the residual error (Fig. 18(a)) is mainly due to this estimation.

5. Conclusion

Many studies in visual servoing concern known objects. The present study addresses the problem of “how to move” in respect to an unknown object. One application is the first step towards a recognition process where it is necessary to perform known motion around the object. The proposed method is based on the visual servoing techniques and is particularly robust. A study of the different interaction relations for the visual features has shown the allowed motions around an object and the conditions of good achievement. All experiments have been successfully implemented on our robotic platform and have shown the validity of such approach. However, the proposed methodology requires some improvement in respect to two points:

- It would be interesting to implement a depth estimation to evaluate the distance z^* ,
- better vision processing would allow one to reduce the measured noise in the experimental results.

Also, other robotic tasks can be defined and new invariant features can be found in consequence.

References

- [1] J. Aloimonos and A. Bandyopadhyay, “Active vision,” ICCV87. Proc. First Int. Conf. Computer Vision, 1987.
- [2] R. Bajcsy, “Active perception,” Proc. IEEE 76, pp.996–1005, Aug. 1988.
- [3] F. Berry, P. Martinet, and J. Gallice, “Visual servoing around a complex object,” Proc. IAPR Workshop on Machine Vision Applications, MVA '98, pp.254–257, Nov. 1998.
- [4] F. Berry, P. Martinet, and J. Gallice, “Trajectory generation by visual servoing,” Proc. IEEE International Conference on Intelligent Robots and Systems, IROS '97, vol.2, pp.1065–1071, Grenoble, France, Sept. 1997.
- [5] F. Berry, P. Martinet, and J. Gallice, “Visual feedback in camera motion generation: Experimental results,” Proc. IEEE International Conference on Intelligent Robots and Systems, IROS '99, vol.1, pp.513–518, Kyongju, Korea, Oct. 1999.
- [6] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin, “Structure from controlled motion,” IEEE Trans. Pattern Anal. & Mach. Intell., vol.18, no.5, pp.492–504, May 1996.
- [7] K. Deguchi and I. Takahashi, “Image based simultaneous control of robot and target object motions by direct image interpretation method,” Proc. IEEE International Conference on Intelligent Robots and Systems, IROS '99, vol.1, pp.375–380, Kyongju, Korea, Oct. 1999.
- [8] B. Espiau, F. Chaumette, and P. Rives, “A new approach to visual servoing in robotics,” IEEE Trans. Robotics and Automation, ICRA92, vol.8, pp.313–326, Paris, June 1992.
- [9] S. Hutchinson, G.D. Hager, and P.I. Corke, “A tutorial on visual servo control,” IEEE Trans. Robot. and Automation, vol.12, no.5, pp.651–670, Oct. 1996.
- [10] M. Jägersand, O. Fuentes, and R. Nelson, “Experimental evaluation of uncalibrated visual servoing for precision manipulation,” Proc. IEEE International Conference on Robotics and Automation, ICRA '97, vol.3, pp.2874–2880, Albuquerque, USA, 1997.
- [11] F. Janabi-Sharifi and W.J. Wilson, “Automatic selection of image features for visual servoing,” IEEE J. Robotics and Automation, vol.5, no.3, pp.404–417, Oct. 1997.
- [12] D. Khadraoui, C. Debain, R. Rouveure, P. Martinet, P. Bonton, and J. Gallice, “Visual-based control in driving assistance of agricultural vehicles,” International Journal of Robotics Research, vol.17, no.10, pp.1040–1054, Oct. 1998.
- [13] D. Khadraoui, G. Motyl, P. Martinet, J. Gallice, and F. Chaumette, “Visual servoing in robotics scheme using a camera/laser-stripe sensor,” IEEE Trans. Robotics and Automation, vol.12, no.5, pp.743–750, Oct. 1996.
- [14] H. Kubota, K. Fuhui, M. Ishikawa, H. Mizoguchi, and Y. Kuno, “Advanced vision processor with an overall image processing unit and multiple local image processing modules,” Proc. IAPR Workshop on Machine Vision Applications, MVA '90, pp.401–404, Tokyo, Japan, Nov. 1990.
- [15] P. Martinet, J. Gallice, and D. Khadraoui, “Vision based control law using 3d visual features,” Proc. Second World Automation Congress, ISRAM '96, vol.3, pp.497–502, Montpellier, France, May 1996.
- [16] P. Martinet, P. Rives, P. Fickinger, and J.J. Borelly, “Parallel architecture for visual servoing applications,” Workshop on Computer Architecture for Machine Perception, CAMP '91, Dec. 1991.
- [17] N.P. Papanikolopoulos, “Selection of features and evaluation of visual measurements during robotic visual servoing tasks,” J. Intelligent and robotic systems, no.13, pp.279–

Appendix

Detail of the interaction matrix for the segment.

$$\begin{pmatrix} \dot{X}_S \\ \dot{Y}_S \\ \dot{L}_S \\ \dot{\alpha} \end{pmatrix} = \begin{pmatrix} \begin{bmatrix} -\nu_2 \\ \nu_2 X_c - \nu_1 L \cos \alpha / 4 \\ -(1 + X_c^2 + L^2 \cos^2 \alpha / 4) \end{bmatrix} & \begin{bmatrix} 0 \\ X_c Y_c + L^2 \cos \alpha \sin \alpha / 4 \\ Y_c \end{bmatrix} \\ \begin{bmatrix} 0 \\ \nu_2 Y_c - \nu_1 L \sin \alpha / 4 \\ -X_c Y_c - L^2 \cos \alpha \sin \alpha / 4 \end{bmatrix} & \begin{bmatrix} -\nu_2 \\ 1 + Y_c^2 + L^2 \sin^2 \alpha / 4 \\ -X_c \end{bmatrix} \\ \begin{bmatrix} \nu_1 \cos \alpha \\ \nu_2 L - \nu_1 (X_c \cos \alpha + Y_c \sin \alpha) \\ -L(X_c(1 + \cos^2 \alpha) + Y_c \cos \alpha \sin \alpha) \end{bmatrix} & \begin{bmatrix} \nu_1 \sin \alpha \\ L(X_c \cos \alpha \sin \alpha + Y_c(1 + \sin^2 \alpha)) \\ 0 \end{bmatrix} \\ \begin{bmatrix} -\nu_1 \sin \alpha / L \\ \nu_1 (X_c \sin \alpha - Y_c \cos \alpha) / L \\ X_c \cos \alpha \sin \alpha - Y_c \cos^2 \alpha \end{bmatrix} & \begin{bmatrix} \nu_1 \cos \alpha / L \\ -X_c \sin^2 \alpha + Y_c \cos \alpha \sin \alpha \\ -1 \end{bmatrix} \end{pmatrix} \cdot \begin{pmatrix} V_x \\ V_y \\ V_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} \quad (\text{A} \cdot 1)$$

304, 1995.

- [18] P. Rives, J.J. Borelly, J. Gallice, and P. Martinet, "Parallel architecture for visual servoing applications," Workshop on Computer Architecture for Machine Perception, CAMP '93, Dec. 1993.
- [19] C. Samson, M. Le Borgne, and B. Espiau, Robot Control. The task function approach, ISBN 0-19-8538057, Clarendon Press, Oxford, 1991.
- [20] G.P. Stein and A. Shashua, "Direct methods for estimation of structure and motion from three views," Technical Report, Report of M.I.T, no.1594, Nov. 1996.
- [21] H.I. Suh, "Visual servoing of robot manipulators by fuzzy membership function based neural networks," vol.7 of Robotics and Automated Systems, pp.285-315, World Scientific Publishing, 1993.
- [22] I. Takhashi and K. Deguchi, "Image based control of robot and target object motions by eigen space method," Proc. IAPR Workshop on Machine Vision Applications, MVA '98, pp.472-475, Chiba, Japan, Nov. 1998.
- [23] R.Y. Tsai, "Moment perserving thresholding: A new approach," Computer Vision, Graphics and Image Processing, CVGIP, vol.29, pp.377-393, Academic Press INC., 1985.
- [24] W.J. Wilson, C.C. Williams Hulls, and G.S. Bell, "Relative end-effector control using cartesian position based visual servoing," IEEE Trans. Robotics and Automation, vol.12, no.5, pp.684-696, Oct. 1996.



François Berry was born in Roanne, France, in 1970. He received the Ph.D. degree from the Blaise Pascal University of Clermont-Ferrand, France, in 1999. Since 1999, he is assistant professor of Computer Science at the Blaise Pascal University of Clermont-Ferrand, France. During his Ph.D. his research interests include visual servoing, vision based control, active vision, visual tracking, and parallel architecture for visual servoing applications.

His current research focus is artificial retina and smart vision sensor.



Philippe Martinet was born in Clermont-Ferrand, France 1962. He received his Electrical Engineering Diploma from the CUST, University of Clermont-Ferrand, France, in 1985, and the Ph.D. degree in Electrical Engineering from the Blaise Pascal University of Clermont-Ferrand, France, in 1987. He spent three years in the Electrical industry. Since 1990, he has been assistant professor of Computer Science at the CUST, Blaise

Pascal University of Clermont-Ferrand, France. His research interests include visual servoing, vision based control, robust control, Automatic Guided Vehicles, active vision, visual tracking, and parallel architecture for visual servoing applications. He works in the LASMEA Laboratory, Blaise Pascal University, Clermont-Ferrand, France.



Jean Gallice was born in Melay, France, in 1945. He received his Ph.D. degrees in 1972 and his Doctorate of Physical Sciences in 1977 from the Blaise-Pascal University (UBP) of Clermont-Ferrand, France. He is Professor since 1978 in the Electrical Engineering Department of the UBP Centre Universitaire des Sciences et Techniques C/U/S/T. His main interests of research and teaching are in Digital Signal Processing and

Robotic. He is the co-supervisor of the research group GRAVIR (Automatique: Vision et Robotique) Laboratoire des Sciences et Matériaux pour l'Electronique, et d'Automatique: LASMEA UMR 6602 du CNRS. His current research focuses on Visual Servoing and Active Vision.