Short Papers

Visual Servoing in Robotics Scheme Using a Camera/Laser-Stripe Sensor

D. Khadraoui, G. Motyl, P. Martinet, J. Gallice, and F. Chaumette

Abstract— **The work presented in this paper belongs to the realm of robotics and computer vision. The problem we seek to solve is the accomplishment of robotics tasks using visual features provided by a special sensor, mounted on a robot end effector. This sensor consists of two laser stripes fixed rigidly to a camera, projecting planar light on the scene. First, we briefly describe the classical visual servoing approach. We then generalize this approach to the case of our special sensor by considering its interaction with respect to a sphere. This interaction permits us to establish a kinematics relation between the sensor and the scene. Finally, both in simulation and in our experimental cell, the results are presented. They concern the positioning task with respect to a sphere, and show the robustness and the stability of the control scheme.**

Index Terms—**Vision-Based Control, Task Function, Laser-Stripe, Interaction Matrix, Sphere.**

I. Introduction

Nowadays, the vision sensor is increasingly an essential element in the resolution of complex problems of environment perception. Its miniaturization and recent image processing developments have made possible, first, the mounting of the visual sensor on the end effector of a robot, and secondly, the integration of visual information in a robot control loop. These developments have made feasible the accomplishment of many more robotics tasks such as target tracking and obstacle avoidance.

Some of the earliest work on the use of sensory feedback was done by Bolles and Paul in [4] and was experimented in a programmable assembly system. Sanderson and Weiss conducted work on the use of visual data in robot control [15]. They presented two separate approaches. The first, commonly called "*position based*", is founded upon the adjustment of the end effector pose parameters [2], [3], [14]. Thus, in this approach, an interpretation step of the end effector pose is necessary. This step usually includes some inaccuracies, depending on the visual sensor geometry, environment and robot models. Moreover, the search for the end effector pose is time consuming and may affect the system's overall behavior. The second approach, which removes the drawbacks of the previous one, directly controls the end effector of a robot using visual data. This control scheme corresponds to the one we developed and is called "*visual servoing*".

In this approach, the control is directly specified in terms of regulation in the image. It is noteworthy that this approach has the advantage of avoiding the intermediate step of the estimation of the 3D pose $\boldsymbol{r}$ of the workpiece with respect to the end effector.

It has been shown in [6] that all visual servoing tasks can be expressed as the regulation to zero of a function $e(\boldsymbol{r}, t)$, called *vision-based task function*, and defined by:

$$e(\boldsymbol{r}, t) = C\,[\boldsymbol{s}(\boldsymbol{r}, t) - \boldsymbol{s}^*] \tag{1}$$

where

- $\boldsymbol{s}(\boldsymbol{r}, t)$ is the value of the visual features currently observed by the camera. These features are associated with the 2D geometric primitives in the image that correspond to the projection of the 3D primitives in the scene. They depend on the pose $\boldsymbol{r}$ between the sensor and the scene;
- $\boldsymbol{s}^*$ is the desired value of $\boldsymbol{s}$ to be reached in the image;
- C is a matrix which has to be selected as an approximation to the inverse Jacobian matrix related to $\boldsymbol{s}$.

For a given vision-based task, modeling consists in choosing the relevant visual features to achieve the task, and then constructing the matrix C. It requires the establishment of the interaction matrix related to the chosen visual features, which is defined by:

$$\dot{\boldsymbol{s}} = L_{\boldsymbol{s}}^T \xi \tag{2}$$

where $\dot{\boldsymbol{s}}$ is the time variation of $\boldsymbol{s}$, and ξ is the object velocity with respect to the sensor (with $\xi = (\boldsymbol{T}, \boldsymbol{\Omega}) = (T_x, T_y, T_z, \Omega_x, \Omega_y, \Omega_z)^T$).

The control problem can then be formalized in terms of *sensor-based-control* [11] applied to visual servoing. A basic control law consists in trying to insure that the task function $e(\boldsymbol{r}, t)$ behaves approximately like a first-order decoupled system. In that case, we should have $\dot{e} = -\lambda e$ where $\lambda (>0)$ controls the speed of the exponential decrease.

Since $e(\boldsymbol{r}, t)$ depends on the motion of both object and sensor, we have:

$$\dot{e} = \frac{\partial e}{\partial \boldsymbol{r}} \xi_c + \frac{\partial e}{\partial t} \tag{3}$$

where the sensor velocity ξ_c is considered as the input of the robot controller. Therefore, ξ_c can be chosen as:

$$\xi_c = \left(\widehat{\frac{\partial e}{\partial \boldsymbol{r}}}\right)^{-1} \left(-\lambda e - \widehat{\frac{\partial e}{\partial t}}\right) \tag{4}$$

where $\widehat{\partial e / \partial \boldsymbol{r}}$ and $\widehat{\partial e / \partial t}$ have to be determined. It is shown in [11] that a sufficient condition for an exponential convergence of e is given by:

$$\frac{\partial e}{\partial \boldsymbol{r}} \left(\widehat{\frac{\partial e}{\partial \boldsymbol{r}}}\right)^{-1} > 0. \tag{5}$$

This relation allows us to choose C as $C = L^{T+}_{|\boldsymbol{s}=\boldsymbol{s}^*}$, where $L^{T+}_{|\boldsymbol{s}=\boldsymbol{s}^*}$ is the pseudo inverse of the interaction matrix computed for $\boldsymbol{s} = \boldsymbol{s}^*$. Indeed, in that case, we have $\partial e / \partial \boldsymbol{r} = L^{T+}_{|\boldsymbol{s}=\boldsymbol{s}^*} L^T_{\boldsymbol{s}}$ which allows us to consider $\widehat{\partial e / \partial \boldsymbol{r}} = \boldsymbol{I}_n$ in the control law (4). The positivity condition (5), now reduced to $L^{T+}_{|\boldsymbol{s}=\boldsymbol{s}^*} L^T_{\boldsymbol{s}} > 0$, is thus satisfied in the neighborhood of $\boldsymbol{s}^*$. Furthermore, $\partial e / \partial t$ represents the contribution of a possible autonomous target motion and is generally unknown.

Manuscript received January 17, 1995; revised June 6, 1996. This work was presented in part at ICAR'93, Tokyo, Japan. This paper was recommended for publication by Associate Editor S. Hutchinson and Editor S. E. Salcudean upon evaluation of reviewers' comments.

D. Khadraoui, P. Martinet, and J. Gallice are with LASMEA, Université Blaise Pascal, 63177 Aubière cedex, France.

G. Motyl is with the Laboratoire TSI, IUT de St-Etienne, Université Jean Monnet, 42023 St-Etienne cedex, France.

F. Chaumette is with IRISA-INRIA Rennes, Campus de Beaulieu, 35042 Rennes cedex, France.

Publisher Item Identifier S 1042-296X(96)07248-5.

Consequently, if the object is motionless ($\partial e/\partial t = 0$), the control law (4) can finally be written:

$$\xi_c = -\lambda e = -\lambda L^{T+}_{|\boldsymbol{s}=\boldsymbol{s}^*}[\boldsymbol{s}(\boldsymbol{r}, t) - \boldsymbol{s}^*] \tag{6}$$

This simple control law only requires the tuning of gain λ, which depends on the rate of the control law and the robot dynamics.

II. Coupling a Camera and Laser Stripe

In [5] and [6], Chaumette and Espiau *et al.* modeled a set of low level geometrical primitives such as points, lines, circles, cylinders and spheres using a single camera. Our work also consists in the modeling of visual data, but by using a sensor termed "*active*" in the sense that it is composed of a camera and two laser stripes. The use of laser stripes allows us to reduce illumination problems. A camera alone detects more information about the image than necessary, and therefore computing time of the image processing is generally high because of the complexity of the scene. Laser stripes remove this drawback because only the information given by the projection of the laser stripes on to the scene is detected by the visual sensor. Laser stripes in robotics have been widely used in real-time tracking of moving objects [13], and also in many applications involving the recognition and interpretation of a workpiece surface [1]. The particular aim of these applications was to search for the three dimensional information of the visualized objects.

In our application, we use two laser stripes rigidly attached to the camera, which are fixed to a robot manipulator. Each stripe projects a light plane on to the scene, which is static.

In this case, the visual features observed are very straightforward to detect and depend only on the geometry of the object. The features are limited to points of discontinuity or straight lines in polyhedral scenes [8], [9], [12]. Therefore, image processing is thus significantly reduced, procuring a saving of time which enhances the dynamics of the system. The only constraint imposed by the laser stripe is to know the approximate position of the laser plane with respect to the camera frame. This can be obtained using classical calibration techniques [8]. Knowing the laser plane parameters and the geometry of the objects, we can model visual data observed in the image. After modeling the related interaction matrices, we can build the control scheme given by (6) which will enable visual servoing.

We now present a general method for the modeling of these interaction matrices. Let us consider an elementary visual signal $\boldsymbol{s}$ provided by this sensor. This camera-laser feature is defined as a function $\boldsymbol{s} = f(\boldsymbol{p}(\boldsymbol{r}))$ which depends on the configuration of the considered primitives, represented by the parameters $\boldsymbol{p}$. These parameters $\boldsymbol{p}$ depend on the pose $\boldsymbol{r}$ between the sensor and the primitive. So, the time variation of $\boldsymbol{s}$ can be obtained as:

$$\dot{\boldsymbol{s}} = \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{r}} \dot{\boldsymbol{r}} \tag{7}$$

where $\dot{\boldsymbol{s}}$ represents the time variation of $\boldsymbol{s}$ in the image, and $\dot{\boldsymbol{r}}$ is nothing but ξ, the object velocity with respect to the sensor ($\xi = \dot{\boldsymbol{r}} = d\boldsymbol{r}/dt$). We then have the interaction matrix $L^T_{\boldsymbol{s}}$ expressed by

$$L^T_{\boldsymbol{s}} = \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{r}} = \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{r}}. \tag{8}$$

The computation of $\partial \boldsymbol{s}/\partial \boldsymbol{p}$ is generally trivial, and we will see in the next section how to compute $\partial \boldsymbol{p}/\partial \boldsymbol{r}$.

If we choose a new representation of the considered primitive, parametered by the function $\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{p})$, which depends on the initial parameters $\boldsymbol{p}$, we can express $\dot{\boldsymbol{q}}$ as follows:

$$\dot{\boldsymbol{q}} = \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{p}} \dot{\boldsymbol{p}} \tag{9}$$

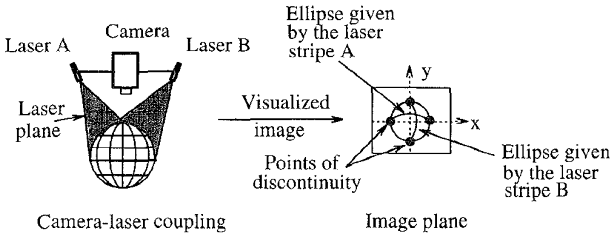


Fig. 1. Camera-laser coupling with a sphere.

and then the corresponding interaction matrix is given by

$$L^T_{\boldsymbol{s}} = \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{q}} \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{r}}. \tag{10}$$

In the past [12], this method has been used to compute interaction matrices related to polyhedral scenes. In that case, visual features only consist of points of discontinuity and straight lines. But this method is more general since it can be applied to any geometrical primitive. In the next section, we present the case of a spherical scene using several representations in the image plane [10].

III. Modelling Visual Features Obtained from a Sphere

In order to model visual features obtained from a spherical scene, it is necessary to select those which can be used in the control scheme. Then, we have to compute the related interaction matrix.The sphere (see Fig. 1) is represented by its center $\boldsymbol{m}_0 = (x_0 y_0 z_0)^T$ and its radius r, such as

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0. \tag{11}$$

Each laser stripe rigidly attached to the camera is characterized by a plane equation:

$$ax + by + cz + d = 0. \tag{12}$$

As shown in [7], the visual servoing approach is not sensitive to approximate models and calibration errors. Therefore we can consider, without loss of generality, a pinhole camera model with unit focal length, so that a point $\boldsymbol{x}(x, y, z)^T$ in 3D space projects into $\boldsymbol{X} = (XY1)^T$ on the image frame with

$$\boldsymbol{X} = \frac{1}{z}\boldsymbol{x}. \tag{13}$$

By using (13) into (12) and (11), we can express the ellipse equation giving the projection in the image of the intersection between the sphere and the laser plane (see Fig. 1). Obviously, the camera only detects the portion of the ellipse corresponding to the near side of the sphere. The equation of this ellipse is given by [5]

$$\begin{aligned} E_1(\boldsymbol{X}, \boldsymbol{A}) = {} & X^2 + A_1 Y^2 + 2A_2 XY + 2A_3 X \\ & + 2A_4 Y + A_5 = 0 \end{aligned} \tag{14}$$

where

$$\begin{aligned} A_1 &= [b^2(z_0^2 + y_0^2 + x_0^2 - r^2) + 2bdy_0 + d^2]/A_0 \\ A_2 &= [ab(z_0^2 + y_0^2 + x_0^2 - r^2) + ady_0 + bdx_0]/A_0 \\ A_3 &= [ac(z_0^2 + y_0^2 + x_0^2 - r^2) + adz_0 + cdx_0]/A_0 \\ A_4 &= [bc(z_0^2 + y_0^2 + x_0^2 - r^2) + bdz_0 + cdy_0]/A_0 \\ A_5 &= [c^2(z_0^2 + y_0^2 + x_0^2 - r^2) + 2cdz_0 + d^2]/A_0 \\ A_0 &= a^2(z_0^2 + y_0^2 + x_0^2 - r^2) + 2adx_0 + d^2 \neq 0. \end{aligned} \tag{15}$$

We now describe the computation of the interaction matrices related to three representations of the sphere projection in the image. The first uses the A_i parameters above. The second uses the classical moments of inertia of the ellipse, and the third searches for the points of discontinuity depicted in Fig. 1.

A. A_i Parameters

In this case, we have $\boldsymbol{s} = \boldsymbol{A} = (A_1, \cdots, A_5)$ and $\boldsymbol{p} = (a, b, c, d, x_0, y_0, z_0, r)$. In order to establish the interaction matrix related to these parameters, we have to compute the time variation of $\boldsymbol{A}$ which can be expressed as follows:

$$\dot{\boldsymbol{A}} = \frac{\partial \boldsymbol{A}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{r}} \xi. \tag{16}$$

In our conditions, we have $\dot{a} = \dot{b} = \dot{c} = \dot{d} = \dot{r} = 0$, since camera and lasers are rigidly coupled, so that the laser plane and the image plane are immovably locked together. $\boldsymbol{p}$ can thus be restricted to $\boldsymbol{p} = \boldsymbol{m}_0 = (x_0, y_0, z_0)$. We have $\dot{\boldsymbol{m}}_0 = -\boldsymbol{T} - \boldsymbol{\Omega} \times \boldsymbol{m}_0$, which allows us easily to compute $\partial \boldsymbol{p}/\partial \boldsymbol{r}$. We obtain

$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{r}} = \begin{pmatrix} -1 & 0 & 0 & 0 & -z_0 & y_0 \\ 0 & -1 & 0 & z_0 & 0 & -x_0 \\ 0 & 0 & -1 & -y_0 & x_0 & 0 \end{pmatrix}. \tag{17}$$

Moreover, we have from (15), (18), shown at the bottom of the page, with $\alpha = 2a^2 x_0 + 2ad$, $\beta = 2a^2 y_0$ and $\gamma = 2a^2 z_0$. L_A^T is thus easily obtained by the matrix product of $\partial \boldsymbol{A}/\partial \boldsymbol{p}$ and $\partial \boldsymbol{p}/\partial \boldsymbol{r}$

B. Moments of Inertia

The ellipse parameters can also be expressed from moments of inertia $m_{ij} = \Sigma_{X \in E} \Sigma_{Y \in E} X^i Y^j$ (with $i + j < 3$), which can easily be extracted from a digitized image. Then, as new representation of the ellipse, we choose the parameters $\boldsymbol{\mu} = (X_c, Y_c, \mu_{20}, \mu_{11}, \mu_{02})$ with [5]

$$\begin{aligned} X_c &= m_{10}/m_{00} = (A_1 A_3 - A_2 A_4)/\Delta \\ Y_c &= m_{01}/m_{00} = (A_4 - A_2 A_3)/\Delta \\ \mu_{20} &= 4(m_{20} - m_{00} X_c^2)/m_{00} = -A_1 K/\Delta \\ \mu_{11} &= 4(m_{11} - m_{00} X_c Y_c)/m_{00} = A_2 K/\Delta \\ \mu_{02} &= 4(m_{02} - m_{00} Y_c^2)/m_{00} = -K/\Delta \end{aligned} \tag{19}$$

with $\Delta = A_2^2 - A_1$ and $K = X_c^2 + 2A_2 X_c Y_c + A_1 Y_c^2 - A_5$. We can now construct the interaction matrix related to $\boldsymbol{s} = \boldsymbol{\mu}$. From (10), we have

$$L_{\boldsymbol{\mu}}^T = \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{A}} \frac{\partial \boldsymbol{A}}{\partial \boldsymbol{r}} \tag{20}$$

where $\partial \boldsymbol{A}/\partial \boldsymbol{r}$ is nothing but the interaction matrix $L_{\boldsymbol{A}}^T$ and $\partial \boldsymbol{\mu}/\partial \boldsymbol{A}$ can easily be determined from (19). We deduce the interaction matrix $L_{\boldsymbol{\mu}}^T$, expressed with the representation $\boldsymbol{\mu}$ using the following relation [8]:

$$\begin{aligned} A_1 &= \mu_{02}/\mu_{20} \\ A_2 &= \mu_{11}/\mu_{20} \\ A_3 &= -(X_c \mu_{20} + Y_c \mu_{11})/\mu_{20} \\ A_4 &= -(Y_c \mu_{02} + Y_c \mu_{11})/\mu_{20} \\ A_5 &= [\mu_{11}^2 - \mu_{20}\mu_{02} + X_c^2 \mu_{20} + 2\mu_{11} X_c Y_c + \mu_{02} Y_c^2]/\mu_{20}. \end{aligned} \tag{21}$$

C. Points of Discontinuity

In order to elaborate the interaction matrix related to the points of discontinuity, it is necessary to determine the expression of two ellipses (see Fig. 1). The first ellipse, given by (14), is the projection on to the image of the intersection between the sphere and the laser plane. The second is given by the sphere projection on to the image (a circle if the sphere is centered in the image) and can be expressed by:

$$\begin{aligned} E_2(\boldsymbol{X}, \boldsymbol{B}) = X^2 + B_1 Y^2 + 2B_2 XY + 2B_3 X \\ + 2B_4 Y + B_5 = 0 \end{aligned} \tag{22}$$

where [5]

$$\begin{aligned} B_1 &= [r^2 - x_0^2 - z_0^2]/B_0 \\ B_2 &= [x_0 y_0]/B_0 \\ B_3 &= [x_0 z_0]/B_0 \\ B_4 &= [y_0 z_0]/B_0 \\ B_5 &= [r^2 - x_0^2 - y_0^2]/B_0 \\ B_0 &= r^2 - y_0^2 - z_0^2 \neq 0. \end{aligned} \tag{23}$$

The intersection of these two ellipses defines two points which are precisely the points of discontinuity under consideration. Then, we can determine the interaction matrix related to each point of discontinuity $\boldsymbol{X}_e = (X_e, Y_e)$. We compute the time variation of the expression $E_1(\boldsymbol{X}, \boldsymbol{A})$ and $E_2(\boldsymbol{X}, \boldsymbol{B})$. We have:

$$\begin{aligned} \dot{E}_1 &= \frac{\partial E_1}{\partial \boldsymbol{X}_e} \dot{\boldsymbol{X}}_e + \frac{\partial E_1}{\partial \boldsymbol{A}} \dot{\boldsymbol{A}} = 0 \\ \dot{E}_2 &= \frac{\partial E_2}{\partial \boldsymbol{X}_e} \dot{\boldsymbol{X}}_e + \frac{\partial E_2}{\partial \boldsymbol{B}} \dot{\boldsymbol{B}} = 0. \end{aligned} \tag{24}$$

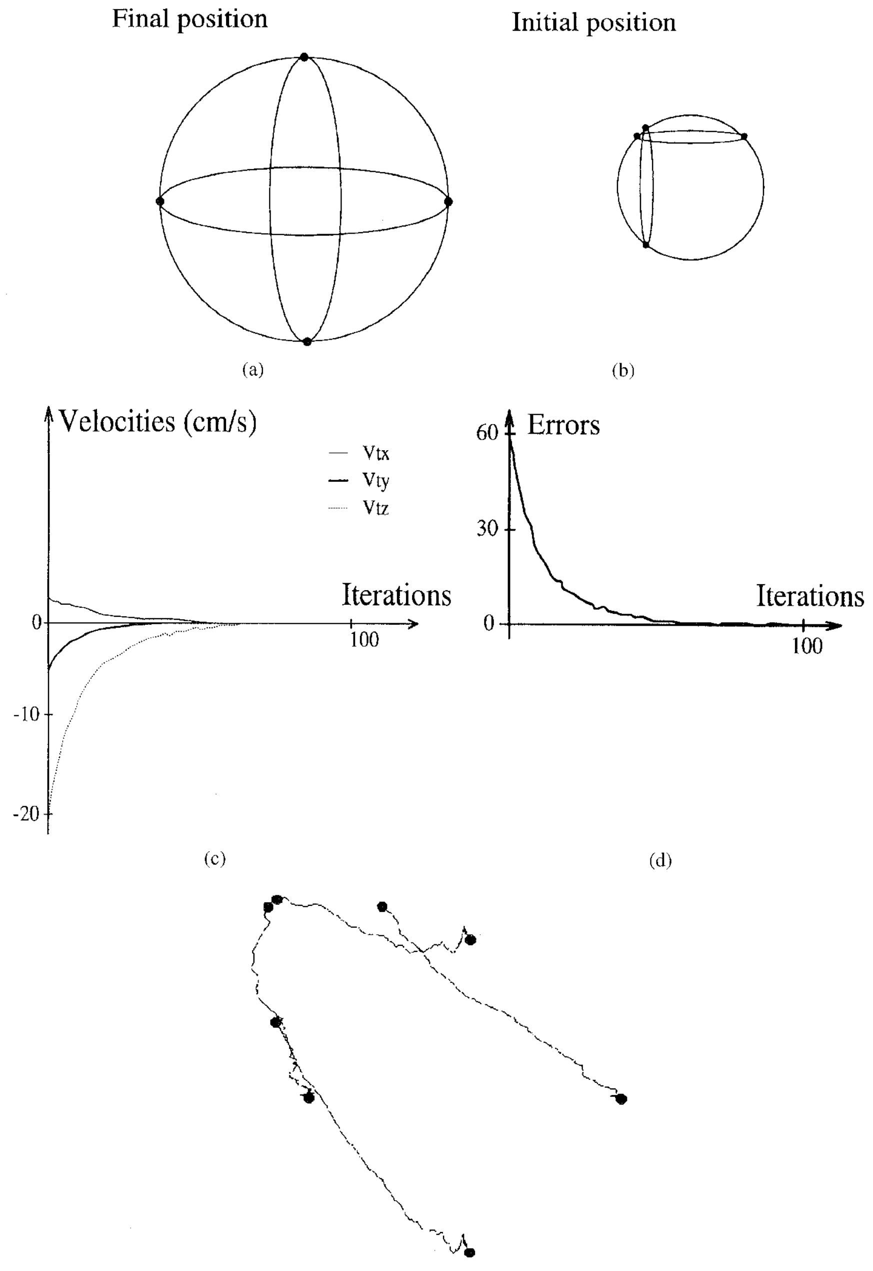
We thus obtain a linear system with $\dot{X}_e, \dot{Y}_e$ as unknowns. The resolution of this system gives the interaction matrix $L_{\boldsymbol{X}_e}^T$ of the point of discontinuity, knowing:

- the coordinates X_e, Y_e of the point of discontinuity extracted after each image acquisition;
- $\boldsymbol{A}$ given by (15), $\dot{\boldsymbol{A}}$ given by (16), $\boldsymbol{B}$ given by (23) and $\dot{\boldsymbol{B}}$ obtained in a similar manner to $\dot{\boldsymbol{A}}$. These expressions depend on ξ and 3D scene parameters.

IV. Results

As a testbed, we used two laser stripes coupled to the camera sensor, mounted on the end effector of a 5 degrees of freedom robot manipulator which does not provide the rotation Ω_x. The chosen task consists in positioning the camera with respect to a sphere in such away that the projection in the image gives a centered circle ($x_0 = y_0 = 0, z_0 = z^*$). The experiments were performed with a sphere of radius 3 cm, the desired distance z^* between camera and object being fixed at 30 cm. We used the different parameterizations presented in the previous section, i.e., the A_i parameters, the moments of inertia and the points of discontinuity. In all our tests, we used a constant gain λ, fixed experimentally at 0.1. With higher values, the control law can become unstable, especially if the robot starts very far from the desired position. On the other hand, with lower values, stability is always ensured with a slower speed of convergence.

$$\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{p}} = \frac{1}{A_0} \begin{pmatrix} 2b^2 x_0 - \alpha A_1 & 2b^2 y_0 + 2bd - \beta A_1 & 2b^2 z_0 - \gamma A_1 \\ 2abx_0 + bd - \alpha A_2 & 2aby_0 + ad - \beta A_2 & 2abz_0 - \gamma A_2 \\ 2acx_0 + cd - \alpha A_3 & 2acy_0 - \beta A_3 & 2acz_0 + ad - \gamma A_3 \\ 2bcx_0 - \alpha A_4 & 2bcy_0 + cd - \beta A_4 & 2bcz_0 + bd - \gamma A_4 \\ 2c^2 x_0 - \alpha A_5 & 2c^2 y_0 - \beta A_5 & 2c^2 z_0 + 2cd - \gamma A_5 \end{pmatrix} \tag{18}$$



(a) (b)

(c) (d)

(e)

Fig. 2. Use of the A_i parameters in the control law.

A. Results Using the A_i Parameters

We first perform the positioning task using the A_i parameters. Both laser planes have been calibrated in order to produce two orthogonal ellipses centered in the image at the desired position. These particular configurations simplify the relation (11) which is given by

$$\text{Laser 1:}\quad \begin{cases} a_{l1}=0, \quad b_{l1}\neq 0, \quad c_{l1}=1 \\ d_{l1}=-(z^{2*}-r^{2*})/z^* \end{cases}$$
$$\text{Laser 2:}\quad \begin{cases} a_{l2}\neq 0, \quad b_{l2}=0, \quad c_{l2}=1 \\ d_{l2}=-(z^{2*}-r^{2*})/z^*. \end{cases} \tag{25}$$

With two laser stripes, we can choose $\boldsymbol{s} = (A_{11}A_{21}A_{31}A_{41}A_{51}A_{12}A_{22}A_{32}A_{42}A_{52})$ where A_{ij} is the parameter A_i corresponding to the ellipse j. At a desired position (see (15)), $\boldsymbol{s}^*$ is given by $\boldsymbol{s}^* = (A_{11}^*00A_{41}^*A_{51}^*A_{12}^*0A_{32}^*0A_{52}^*)$ with:

$$\begin{aligned}
A_{11}^* &= [b_{l1}^2(z^*-r^2)+d_{l1}^2]/d_{l1}^2 \\
A_{41}^* &= [b_{l1}c_{l1}(z^*-r^2)+b_{l1}d_{l1}z^*]/d_{l1}^2 \\
A_{51}^* &= [c_{l1}^2(z^*-r^2)+2c_{l1}d_{l1}z^*+d_{l1}^2]/d_{l1}^2 \\
A_{12}^* &= d_{l2}^2/[a_{l2}^2(z^*-r^2)+d_{l2}^2] \\
A_{32}^* &= [a_{l2}c_{l2}(z^*-r^2)+a_{l2}d_{l2}z^*]/[a_{l2}^2(z^*-r^2)+d_{l2}^2] \\
A_{52}^* &= [c_{l2}^2(z^*-r^2)+2c_{l2}d_{l2}z^*+d_{l2}^2]/[a_{l2}^2(z^*-r^2)+d_{l2}^2].
\end{aligned}$$

The interaction matrix related to $\boldsymbol{s} = \boldsymbol{s}^*$ is expressed as (26), shown on the next page. In this case, the rank of $L^T_{|\boldsymbol{s}=\boldsymbol{s}^*}$ is 3, with kernel given by:

$$Ker(L^T_{|\boldsymbol{s}=\boldsymbol{s}^*}) = \begin{pmatrix} -z^* & 0 & 0 \\ 0 & z^* & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{27}$$

This means that 3 camera degrees of freedom, correctly chosen using the form of the interaction matrix, are sufficient to achieve the positioning task. We have selected the camera translational velocities T_x, T_y and T_z. In this case, the corresponding interaction matrix consists of the first three columns of the general form given in (26). We note that, because of the particular configuration of $L^T_{|\boldsymbol{s}=\boldsymbol{s}^*}$, Ω_y could be used instead of V_x and Ω_x instead of V_y.

We present simulation results in Fig. 2, with noise corresponding to errors in the robot and sensor geometric models, as well to errors due to the image processing (2% white noise on the image coordinates and on the camera location). The different parts show (see Fig. 2):

- $(2a)$: the target image (configuration of the ellipse in the desired image);
- $(2b)$: the initial image observed by the camera before visual servoing;
- $(2c)$: the behavior of each component of the control vector during visual servoing (translational velocity T_x, T_y, T_z);
- $(2d)$: the evolution of the overall error $||\boldsymbol{s} - \boldsymbol{s}^*||$;
- $(2e)$: the behavior of each point of discontinuity during the task.

$$L^T_{|\boldsymbol{s}=\boldsymbol{s}^*} = -\begin{pmatrix}
0 & \frac{2b_1d_1}{A^*_{01}} & \frac{2b_1^2z^*}{A^*_{01}} & \frac{-2b_1d_1z^*}{A^*_{01}} & 0 & 0 \\
\frac{b_1d_1}{A^*_{01}} & 0 & 0 & 0 & \frac{b_1d_1z^*}{A^*_{01}} & 0 \\
\frac{c_1d_1}{A^*_{01}} & 0 & 0 & 0 & \frac{c_1d_1z^*}{A^*_{01}} & 0 \\
0 & \frac{c_1d_1}{A^*_{01}} & \frac{b_1(2c_1z^*+d_1)}{A^*_{01}} & \frac{-c_1d_1z^*}{A^*_{01}} & 0 & 0 \\
0 & 0 & \frac{2c_1(c_1z^*+d_1)}{A^*_{01}} & 0 & 0 & 0 \\
\frac{-2A^*_{12}a_2d_2}{A^*_{02}} & 0 & \frac{-2A^*_{12}a_2^2z^*}{A^*_{02}} & 0 & \frac{-2A^*_{12}a_2d_2z^*}{A^*_{02}} & 0 \\
0 & \frac{a_2d_2}{A^*_{02}} & 0 & \frac{-a_2d_2z^*}{A^*_{02}} & 0 & 0 \\
\frac{-d_2(2A^*_{32}a_2-c_2)}{A^*_{02}} & 0 & \frac{-a_2(-2c_2z^*-d_2+2A^*_{32}a_2z^*)}{A^*_{02}} & 0 & \frac{-d_2z^*(2A^*_{32}a_2-c_2)}{A^*_{02}} & 0 \\
0 & \frac{c_2d_2}{A^*_{02}} & 0 & \frac{-c_2d_2z^*}{A^*_{02}} & 0 & 0 \\
\frac{-2A^*_{52}a_2d_2}{A^*_{02}} & 0 & \frac{2c_2^2z^*+2c_2d_2-2A^*_{52}a_2^2z^*}{A^*_{02}} & 0 & \frac{-2A^*_{52}a_2d_2z^*}{A^*_{02}} & 0
\end{pmatrix} \tag{26}$$

$$L^T_{|\boldsymbol{s}=\boldsymbol{s}^*} = -\begin{pmatrix}
\frac{1}{\alpha} & 0 & 0 & 0 & \frac{z^*}{\alpha} & 0 \\
0 & \frac{\mu^*_{20_1}}{\alpha\mu^*_{02_1}} & -\beta\frac{(2z^*-\alpha)}{\alpha^2\mu^*_{02_1}} & -\frac{\mu^*_{20_1}z^*}{\alpha\mu^*_{02_1}} & 0 & 0 \\
0 & 2\frac{\mu^*_{20_1}\beta}{\alpha\mu^*_{02_1}} & -2\frac{K_1}{\alpha^2\mu^*_{02_1}} & -2\frac{\mu^*_{20_1}\beta z^*}{\alpha\mu^*_{02_1}} & 0 & 0 \\
-\frac{\beta}{\alpha} & 0 & 0 & 0 & -\frac{\beta z^*}{\alpha} & 0 \\
0 & 0 & -2\frac{(z^*-\alpha)}{\alpha^2} & 0 & 0 & 0 \\
\frac{\alpha}{A_0} & 0 & -\gamma\frac{(2z^*-\alpha)}{A_0\mu^*_{02_2}} & 0 & \frac{\alpha z^*}{A_0} & 0 \\
0 & \frac{\mu^*_{20_2}\alpha}{A_0\mu^*_{02_2}} & 0 & -\frac{\mu^*_{20_2}\alpha z^*}{A_0\mu^*_{20_2}} & 0 & 0 \\
0 & 0 & -2\frac{\mu^*_{20_2}(z^*-\alpha)}{A_0\mu^*_{20_2}} & 0 & 0 & 0 \\
0 & -\frac{\alpha\mu^*_{20_2}\gamma}{A_0\mu^*_{02_2}} & 0 & \frac{\alpha\mu^*_{20_2}\gamma z^*}{A_0\mu^*_{02_2}} & 0 & 0 \\
2\frac{\alpha\gamma}{A_0} & 0 & -2\frac{K_2}{A_0\mu^*_{02_2}} & 0 & 2\frac{\alpha\gamma z^*}{A_0} & 0
\end{pmatrix}$$

$$\text{with} \quad \begin{cases} \alpha = (z^{2*} - r^2)/z^* \\ \beta = a_{l1}\mu^*_{20_1} \\ \gamma = b_{l2}\mu^*_{20_2} \end{cases} \quad \text{and} \quad \begin{cases} A_0 = a_{l2}^2(z^{2*} - r^2) + \alpha^2 \\ K_1 = \mu^*_{20_1}(z^* - \alpha) + \beta^2 z^* \\ K_2 = \mu^*_{02_2}(z^* - \alpha) + \gamma^2 z^* \end{cases} \tag{29}$$

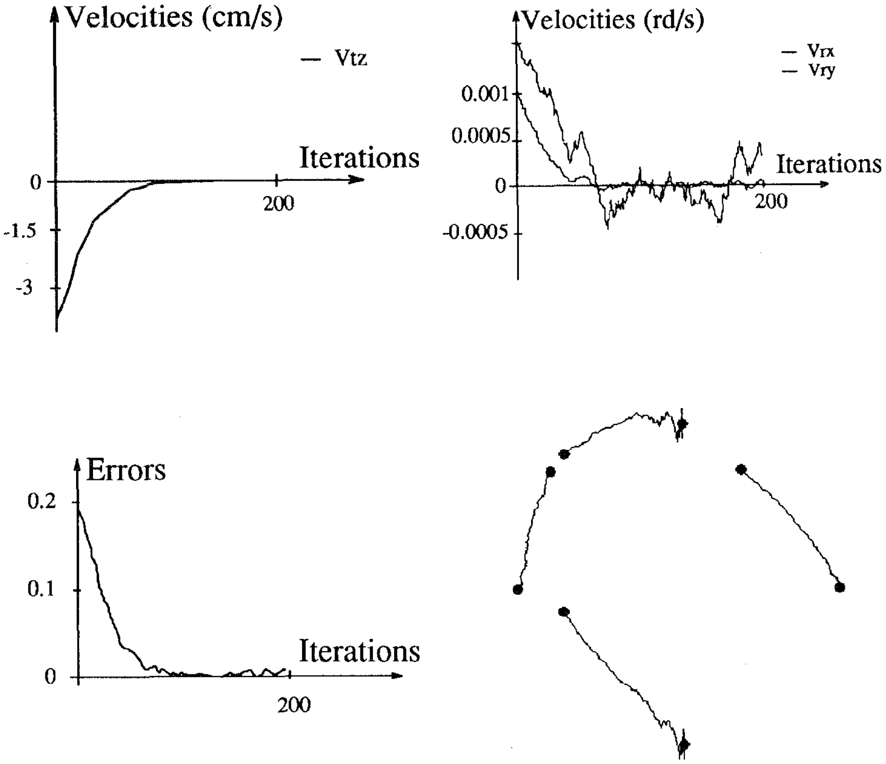


Fig. 3. Use of the moments of inertia in the control law.

We observe in Fig. 2 that the convergence to the desired image is correctly performed. The stability and robustness of the control scheme have been proved under a variety of simulation conditions. Note that the noise introduced into the measurements and robot locations brings little perturbation to the system.

B. Results Using the Moments of Inertia

We can perform the same positioning task from parameters given by the moments of inertia. We now choose $\boldsymbol{s} = (X_{c_1}, Y_{c_1}, \mu_{20_1}, \mu_{11_1}, \mu_{02_1}, X_{c_2}, Y_{c_2}, \mu_{20_2}, \mu_{11_2}, \mu_{02_2})$. Using (19), $\boldsymbol{s}^*$ is expressed by

$$\begin{aligned} X^*_{c_1} &= X^*_{c_2} = Y^*_{c_1} = Y^*_{c_2} = \mu^*_{11_1} = \mu^*_{11_2} = 0 \\ \mu^*_{20_1} &= r^{2*}/(b^2_{l1} z^{2*} + z^{2*} - r^{2*}) \\ \mu^*_{02_2} &= r^{2*}/(a^2_{l2} z^{2*} + z^{2*} - r^{2*}) \\ \mu^*_{02_1} &= \mu^*_{20_2} = r^{2*}/(z^{2*} - r^{2*}). \end{aligned} \tag{28}$$

The expression of the interaction matrix related to $\boldsymbol{s} = \boldsymbol{s}^*$ is easily obtained; see (29), shown at the bottom of the previous page.

This matrix has the same kernel as that using the A_i parameters. In this case, we used camera velocity components T_z, Ω_x and Ω_y to accomplish the task. The expression of the corresponding task function and control law can be obtained in a similar manner as stated previously. Fig. 3 presents the simulation results obtained in the presence of noise.

C. Results Using the Points of Discontinuity

Finally, the positioning task with respect to a sphere was carried out in our experimental cell using the points of discontinuity of the ellipse. Using two laser stripes, we have $\boldsymbol{s} = (X_1, Y_1, \cdots, X_4, Y_4)$, where the first two points belong to laser plane 1 ($a_{l1} = 0$) and the last two points belong to laser plane 2 ($b_{l2} = 0$). At the desired position, the sphere is centered in the image, thus $\boldsymbol{s}^* =$

$$L^T_{|\boldsymbol{s}=\boldsymbol{s}^*} = \frac{-1}{K} \begin{pmatrix} z^* & 0 & -X^*z^* & 0 & {z^*}^2 & 0 \\ \frac{-X^*K}{2b_{l1}} & 0 & \frac{{z^*}^2 + r^2}{2b_{l1}} & 0 & -\frac{X^*Kz^*}{2b_{l1}} & 0 \\ z^* & 0 & X^*z^* & 0 & {z^*}^2 & 0 \\ \frac{X^*K}{2b_{l1}} & 0 & \frac{{z^*}^2 + r^2}{2b_{l1}} & 0 & \frac{X^*Kz^*}{2b_{l1}} & 0 \\ 0 & -\frac{Y^*K}{2a_{l2}} & \frac{{z^*}^2 + r^2}{2a_{l2}} & \frac{Y^*Kz^*}{2a_{l2}} & 0 & 0 \\ 0 & z^* & -Y^*z^* & -{z^*}^2 & 0 & 0cr0 & \frac{Y^*K}{2a_{l2}} & \frac{{z^*}^2 + r^2}{2a_{l2}} & -\frac{Y^*Kz^*}{2a_{l2}} & 0 & 0 \\ 0 & z^* & Y^*z^* & -{z^*}^2 & 0 & 0 \end{pmatrix}$$

$$\text{with} \quad K = {z^*}^2 - r^2 \tag{30}$$

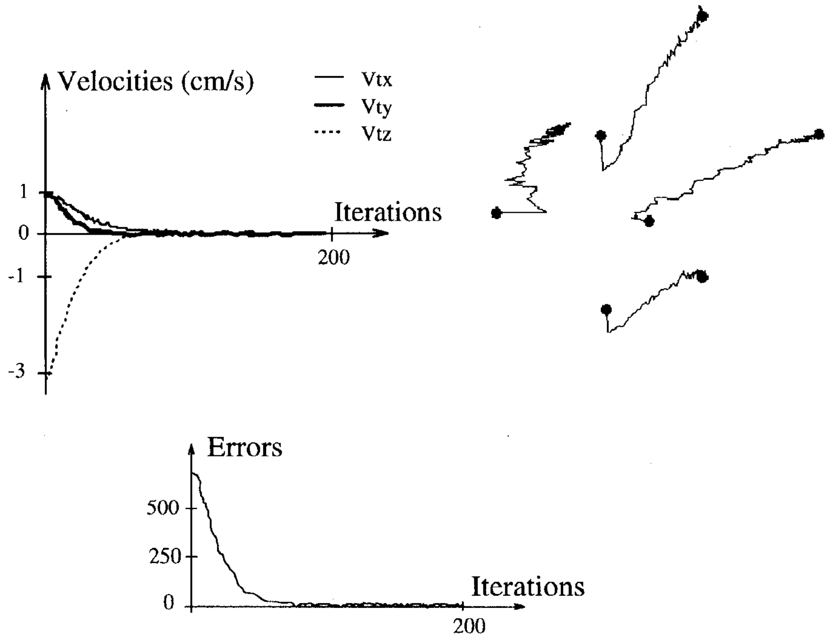


Fig. 4. Use of points of discontinuity in the control law.

$(X^*, 0, -X^*, 0, 0, Y^*, 0, -Y^*)$ with $X^* = Y^* = r/\sqrt{z^{*2} - r^2}$. The rank of the interaction matrix related to $s = s^*$ is again 3. The matrix is obtained as (30), shown at the bottom of the previous page.

The processing applied to the image consists of a simple direct thresholding which keeps only the information corresponding to the over-light area due to the laser stripe projection on to the scene. It is then easy to extract the points of discontinuity of the ellipse with a sampling rate equal to the video rate (25 Hz). Results obtained in our experimental cell using the three translational degrees of freedom are shown in Fig. 4. They show the stability and the exponential convergence of the control law. We may remark that these experimental results and those obtained in simulation with other parameters display approximately the same behavior.

V. Conclusion

In some applications, using only a camera may turn out to be restrictive due to the difficulty of extracting "useful" information from the image. That is why a special sensor, created by the coupling of a camera and laser stripe, was chosen. Indeed, with such a sensor, image processing is significantly reduced, and moreover the primitives in the image are relatively straightforward. In this paper, we have presented a general method for modeling visual features using this useful sensor, and applied this method to a spherical scene case. Then we integrated this work under the *task function approach* which enables positioning robotics tasks to be performed with good results from the point of view of robustness and stability. Finally, in simulation and in our experimental cell, results were presented for the positioning task with respect to a sphere.

In all the experiments, the interaction matrix and its pseudo-inverse C are computed only once since they are chosen constant and correspond to their value computed at $s = s^*$ desired image. Computation time is thus the same as far as interaction matrices are concerned. In the same way, numerical stability is equivalent (and good) for the three representations that we have studied. The main differences between the three approaches are the manner of extracting the visual data by image processing and the level of noise related to each of them. In our case, no significant difference has really been pointed out. Let us note that the method we have presented to compute the interaction matrix is general. Similar derivations could be obtained for other non polyhedral objects such as circles, cylinders, etc.

We have demonstrated the various advantages of a camera-laser coupling. Nevertheless, this sensor has some constraints. This coupling involves some restrictions in the laser stripe projection on to the scene. It is necessary to choose the most favorable attitude of the laser stripe in order to achieve a robotics task under conditions of optimum stability. Moreover, a calibration step is essential in order to compute each parameter of a laser plane. This calibration step is necessary in order to compute the desired position to be reached in the image (this could also be done by a learning approach) but, as shown in [7], calibration parameters are not sensitive for the stability, robustness and convergence of visual servoing.

References

[1] J. E. Agapakis, "Approaches for recognition and interpretation of workpiece surface features using structured lighting," *Int. J. Robot. Res.*, vol. 9, no. 5, pp. 3–16, Oct. 1990.

[2] G. J. Agin, "Real-time control of a robot with a mobile camera," in *9th Int. Symp. on Industrial Robots*, Washington D.C., Mar. 1979, pp. 233–246.

[3] J. S. Albus, A. J. Barbera, and M. L. Fitzgerald, "Hierarchical control for sensory interactive robots," in *Proc. of 11th Int. Symp. on Industrial Robotics*, Tokyo, Japan, Oct. 1981, pp. 497–505.

[4] R. Bollcs and R. Paul, "The use of Sensory feedback in a Programmable Assembly System," Stanford Artificial Intelligence Laboratory, MEMO AIM-220, STAN-Cs-396, Oct. 1970.

[5] F. Chaumette, "La relation vision-commande: théorie et application à des tâches robotiques," Ph.D. Dissertation, IRISA/INRIA, Rennes, France, July 1990.

[6] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Automat.*, vol. 8, n. 3, pp. 313–326, June 1992.

[7] B. Espiau, "Effect of camera calibration errors on visual servoing in robotics," in *Proc. 3rd Int. Symp on Experimental Robotics*, Kyoto, Japan, Oct. 1993.

[8] G. Motyl "Couplage d'une caméra et d'un faisceau laser en commande référencée vision," Ph.D. Dissertation, Clermont-Ferrand, France, Sept. 1992.

[9] G. Motyl, F. Chaumette, and J. Gallice, "A camera and laser stripe in sensor based control," *Second Int. Symp. on Measurement and Control in Robotics*, AIST Tsukuba Research Center, Japan, Nov. 1992, pp. 685–692.

[10] G. Motyl, P. Martinet, J. Gallice, "Visual servoing with respect to a target sphere using a camera/laser-stripe sensor," in *1993 Int. Conf. on Advanced Robotics, ICAR'93*, Tokyo, Japan, Nov. 1993, pp. 591–596.

[11] C. Samson, M. Le Borgne, and B. Espiau, *Robot Control: The Task Function Approach.* Cambridge, U.K.: Oxford Univ. Press, 1991.

[12] J. P. Urban, G. Motyl, and J. Gallice, "Real-time visual servoing using controlled illumination," *Int. J. Robot. Res.*, vol 13, no. 1, pp. 93–100, Feb. 1994.

[13] S. Venkatesan and C. Archibald, "Real-time tracking in five degrees of freedom using two wrist-mounted laser range finders," in *IEEE Int. Conf. on Robotics and Automation*, pp. 2004–2010, 1990.

[14] M. R. Ward, L. Rossol, S. W. Holland, and R. Dewar, "CONSIGHT: A practical vision based robot guidance system," in *Proc 9th Int. Symp. on Industrial Robotics*, Washington, D.C., Mar. 1979, pp. 195–211.

[15] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE J. Robot. Automat.*, vol. RA-3, no. 5, pp. 404–417, Oct. 1987.

Subspace Methods for Robot Vision

Shree K. Nayar, Sameer A. Nene, and Hiroshi Murase

***Abstract*— In contrast to the traditional approach, visual recognition is formulated as one of matching appearance rather than shape. For any given robot vision task, all possible appearance variations define its visual workspace. A set of images is obtained by coarsely sampling the workspace. The image set is compressed to obtain a low-dimensional subspace, called the eigenspace, in which the visual workspace is represented as a continuous appearance manifold. Given an unknown input image, the recognition system first projects the image to eigenspace. The parameters of the vision task are recognized based on the exact location of the projection on the appearance manifold. An efficient algorithm for finding the closest manifold point is described. The proposed appearance representation has several applications in robot vision. As examples, a precise visual positioning system, a real-time visual tracking system, and a real-time temporal inspection system are described.**

***Index Terms*—Visual workspace, parametric eigenspace representation, learning appearance manifolds, image recognition, nearest neighbor, visual positioning, real-time tracking, temporal inspection.**

I. Introduction

For a robot to be able to interact in a precise and intelligent manner with its environment, it must rely on sensory feedback. Vision serves as a powerful component of such a feedback system. It provides a richness of information that can enable a manipulator to handle uncertainties inherent to a task, react to a varying environment, and gracefully recover from failures. In order for the robot to interact with objects in its workspace, it requires a-priori models of the objects. Traditionally, robot vision systems have heavily relied on shape (CAD) models [4].

Will shape representation suffice? After all, most vision applications deal with brightness images that are functions not only of shape but also other intrinsic scene properties such as reflectance and perpetually varying factors such as illumination. This observation has motivated us to take an extreme approach to visual representation. What we seek is not a representation of geometry but rather *appearance* [20], encoded in which are brightness variations caused by three-dimensional shape, surface reflectance properties, illumination conditions, and the parameters of the robot task. Given the number of factors at work, it is immediate that an appearance representation that captures all possible variations is simply impractical. Fortunately, there exist a wide collection of robot vision applications where pertinent variables are few and hence compact appearance representation in a low-dimensional *subspace* is indeed practical.

A problem of substantial relevance to robotics is *visual servoing;* the ability of a robot to either automatically position itself at a desired location with respect to an object, or accurately follow an object as it moves through an unknown trajectory. We use the visual servoing problem to describe our appearance based approach. To place our approach in perspective, we review existing methods for servoing. All of these methods can be broadly classified into two categories; (a) feature/model based and (b) learning based. The first category uses image features to estimate the robot's displacement with respect to the object. The objective is to find the rotation and translation that must be applied to the end-effector to bring the features back to their desired positions in the image. Image features used vary from geometric primitives such as edges, lines, vertices, and circles [33], [5], [10], [7] to optical flow estimates [26], [13], [3] and object location estimates obtained using stereo [2]. The control schemes used to drive the robot to its desired position vary from simple prediction algorithms employed to achieve computational efficiency to more sophisticated adaptive self-tuning controllers that account for the dynamics of the manipulator. Many of the above methods require prior calibration of the vision sensor's intrinsic parameters (e.g., focal length) as well as its extrinsic parameters (e.g., rotation and translation with respect to the manipulator).

The second category of servoing methods includes a learning component. In the learning stage, the mapping between image features and robot coordinates is generated prior (off-line) to positioning or tracking. This mapping is then used to determine, in real-time, errors in robot position/velocity from image feature coordinates. This is generally accomplished without any explicit knowledge of the object's geometry or the robot's kinematic parameters. In addition, calibration of the vision sensor is not required as long as the sensor-robot configuration remains unaltered between learning and servoing. These methods differ from each other primarily in the type of learning algorithm used. The learning strategies vary from neural-like networks [11], [14], [17], [32] to table lookup mechanisms such as the cerebellar model articulation controller (CMAC) [1], [16].

Our appearance based approach to robot vision offers a solution to servoing that differs from previous work in two significant ways; (a) the method uses raw brightness images directly without the computation of image features, and (b) the learning algorithm is based on principal component analysis [25], [6] rather than a large

Manuscript received March 9, 1995; revised January 15, 1996. This work was supported in part by the NSF National Young Investigator Award, ARPA Contract DACA 76-92-C-0007, and the David and Lucille Packard Fellowship. This paper was presented in part at the IEEE Conference on Robotics and Automation, San Diego, May 1994. This research was conducted at the Center for Research in Intelligent Systems, Department of Computer Science, Columbia University. This paper was recommended for publication by Associate Editor S. Hutchinson and Editor S. E. Salcudean upon evaluation of reviewers' comments.

S. K. Nayar and S. A. Nene are with the Department of Computer Science, Columbia University, New York, NY 10027 USA.

H. Murase is with the NTT Basic Research Laboratory, Kanagawa 243-01, Japan.

Publisher Item Identifier S 1042-296X(96)07238-2.

1042–296X/96$05.00 © 1996 IEEE