VISUAL SERVOING USING IMAGE FEATURES DEFINED UPON GEOMETRICAL PRIMITIVES

François Chaumette

IRISA / INRIA Rennes,
Campus de Beaulieu,
35042 Rennes cedex, France
chaumett@irisa.fr

Abstract

The image features used in visual servoing or tracking are generally the coordinates of image points. In this paper, we present how it is possible to define more complex visual features based on geometrical primitives such as lines, spheres and cylinders. Using such features enables to realize a large variety of robotics tasks depending on the desired virtual linkage between the camera and the objects in the scene. We then describe a control scheme in closed loop with respect to these image features which is based on the task function approach. This scheme combines the regulation of the selected vision-based task with the minimization of a secondary cost function, such as a trajectory tracking using the robot degrees of freedom not constrained by the vision-based task. We finally present real time experimental results obtained with a camera mounted on the end-effector of a six d.o.f. robot.

1. Introduction

This paper focuses on the utilization of the visual servoing approach as an adequate way to perform non-contact sensor-based tasks in robotics. Non-contact sensing is useful in the achievement of many kinds of robotics tasks in various application domains. For example, relative positioning errors due to the inaccuracies in the modeling of robot kinematics can be compensated using an eye-in-hand robotics system. By visual servoing, we mean the use of image features embedded in a closed-loop control scheme and providing with some measurements of the interaction between the robot and its local environment at a sufficiently high sampling rate for ensuring the stability of the robot control loop. Generally, the image features which are used in visual servoing or visual tracking are composed with a set of image point coordinates which have to reach particular values in the image in order to realize the specified task [1], [5], [6], [7], [8], [10]. In this paper, we generalize this approach to more complex visual features defined upon the projection in the image of geometrical primitives (such as lines, spheres, cylinders,...). Using such features enables to realize a large variety of robotics tasks depending on the desired virtual linkage between the camera and the objects in the scene. For example, positioning the camera with respect to a square realizes a rigid linkage, with respect to a sphere a ball-and-socket linkage and with respect to a cylinder a prismatic-revolute linkage [3]. Furthermore, it can be interesting to combine these focusing and fixating vision-based tasks with trajectory trackings around the considered object, allowing for instance to inspect its surface. We thus describe a control scheme based on the task function approach [9] which combines the regulation of the selected vision-based task with the minimization of a secondary cost function, such as a trajectory tracking using the robot degrees of freedom not constrained by the vision-based task. We finally present real time experimental results obtained with a camera mounted on the end-effector of a six d.o.f. robot. The first detailed experiment consists in positioning the camera with respect to a cylinder while performing camera motion around it. The second one consists in positioning the camera with respect to a sphere.

2. The interaction matrix

In this section, we describe what are the data extracted from an image which can be incorporated in a vision-based control scheme. In fact, it can be shown that such an ability relies on the knowledge of the spatio-temporal behavior of a visual data with respect to camera motion (that we call in the following the interaction matrix related to the considered feature).

Let us consider a geometrical primitive $\mathcal{P}_s$ of the scene; its configuration is specified by an equation of the type:

$$h(\underline{x}, \underline{p}) = 0 \ , \quad \forall \underline{x} \in \mathcal{P}_s \tag{1}$$

where h defines the kind of the primitive and the value of $\underline{p}$ stands for its corresponding configuration. Let us denote $\mathcal{P}_i$ as the projection in the image plane of $\mathcal{P}_s$. The configuration of $\mathcal{P}_i$ can be written as follows:

$$g(\underline{X}, \underline{P}) = 0 \ , \quad \forall \underline{X} \in \mathcal{P}_i \qquad (2)$$

where g defines the kind of the image primitive and the value of $\underline{P}$ its configuration. Let us denote $T_c = (V, \omega)$ as the camera velocity screw, where V and ω represent its translational and rotational components. The time variation of $\underline{P}$, which links the motion in the image to the camera motion T_c, can be explicitly derived [4] and we get:

$$\dot{\underline{P}} = L^T_{\underline{P}} \, T_c \qquad (3)$$

where $L^T_{\underline{P}}$, called the interaction matrix related to $\underline{P}$, fully characterizes the interaction between the camera and the considered primitive.

In [10] and [6], an experimental learning approach is proposed to compute the interaction matrix related to points. But it is also possible to derive it in an explicit way [5]. More generally, in [4], a systematic method for computing the interaction matrix of any set of visual features defined upon geometrical primitives (lines, spheres, cylinders,...) is proposed.

For example, in the classical case of a point with coordinates (x, y, z) and $\underline{X} = (X, Y) = (x/z, y/z)$ in the image plane, we have:

$$L^T_{\underline{X}} = \begin{pmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{pmatrix} \qquad (4)$$

Let us now consider the case of a cylinder whose equation is given by:

$$\begin{aligned} h(\underline{x}, \underline{p}) \ &= \ (x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2 \quad (5) \\ &- \ (ax+by+cz)^2 - r^2 = 0 \end{aligned}$$

$$\text{with} \left\{ \begin{array}{l} a^2 + b^2 + c^2 = 1 \\ ax_0 + by_0 + cz_0 = 0 \end{array} \right.$$

where r is the radius of the cylinder, a, b and c are the coordinates of its direction vector and x_0, y_0 and z_0 are the coordinates of the nearest point belonging to the cylinder axis from the projection center.

The projection of a cylinder on the image plane is (for non-degenerated cases) a set of two straight lines with equation:

$$\left\{ \begin{array}{lll} D_1 & : & X\cos\theta_1 + Y\sin\theta_1 - \rho_1 = 0 \\ D_2 & : & X\cos\theta_2 + Y\sin\theta_2 - \rho_2 = 0 \end{array} \right. \qquad (6)$$

with:

$$c_1 = \cos\theta_1 = \frac{rx_0/A - \alpha}{\sqrt{(rx_0/A-\alpha)^2 + (ry_0/A-\beta)^2}}$$

$$c_2 = \cos\theta_2 = \frac{rx_0/A + \alpha}{\sqrt{(rx_0/A+\alpha)^2 + (ry_0/A+\beta)^2}}$$

$$s_1 = \sin\theta_1 = \frac{ry_0/A - \beta}{\sqrt{(rx_0/A-\alpha)^2 + (ry_0/A-\beta)^2}}$$

$$s_2 = \sin\theta_2 = \frac{ry_0/A + \beta}{\sqrt{(rx_0/A+\alpha)^2 + (ry_0/A+\beta)^2}}$$

$$\rho_1 = \frac{rz_0/A - \gamma}{\sqrt{(rx_0/A-\alpha)^2 + (ry_0/A-\beta)^2}}$$

$$\rho_2 = \frac{rz_0/A + \gamma}{\sqrt{(rx_0/A+\alpha)^2 + (ry_0/A+\beta)^2}}$$

$$\text{where} \left\{ \begin{array}{l} A = \sqrt{x_0^2 + y_0^2 + z_0^2 - r^2} \\ \alpha = cy_0 - bz_0 \\ \beta = az_0 - cx_0 \\ \gamma = bx_0 - ay_0 \end{array} \right.$$

Finally, by choosing $\underline{P} = (\rho_1, \theta_1, \rho_2, \theta_2)$, the interaction matrix related to $\underline{P}$ is given by [3]:

$$L^T_{\underline{P}} = \begin{pmatrix} \lambda_{\rho_1}c_1 & \lambda_{\rho_1}s_1 & -\lambda_{\rho_1}\rho_1 & \alpha_1 s_1 & -\alpha_1 c_1 & 0 \\ \lambda_{\theta_1}c_1 & \lambda_{\theta_1}s_1 & -\lambda_{\theta_1}\rho_1 & -\rho_1 c_1 & -\rho_1 s_1 & -1 \\ \lambda_{\rho_2}c_2 & \lambda_{\rho_2}s_2 & -\lambda_{\rho_2}\rho_2 & \alpha_2 s_2 & -\alpha_2 c_2 & 0 \\ \lambda_{\theta_2}c_2 & \lambda_{\theta_2}s_2 & -\lambda_{\theta_2}\rho_2 & -\rho_2 c_2 & -\rho_2 s_2 & -1 \end{pmatrix} \qquad (7)$$

$$\text{with} \left\{ \begin{array}{lcl} \alpha_1 & = & (1+\rho_1^2) \\ \alpha_2 & = & (1+\rho_2^2) \\ \lambda_{\rho_1} & = & -(x_0\rho_1 c_1 + y_0\rho_1 s_1 + z_0)/A^2 \\ \lambda_{\theta_1} & = & (y_0 c_1 - x_0 s_1)/A^2 \\ \lambda_{\rho_2} & = & -(x_0\rho_2 c_2 + y_0\rho_2 s_2 + z_0)/A^2 \\ \lambda_{\theta_2} & = & (y_0 c_2 - x_0 s_2)/A^2 \end{array} \right.$$

We may thus choose as visual features, denoted $\underline{s}$, the parameters $\underline{P}$ which describe the configuration of one or several primitives observed in the image (such as the coordinates of a point, the orientation and distance to origin of a line, the inertial moments of an ellipse, etc) or, more generally, any differentiable expression obtained from $\underline{P}$ (such as the distance between a point and a line, the orientation between two lines, etc).

The design of a vision-based task now consists in selecting the visual features $\underline{s}$, which realize the specified task from the a priori knowledge on the application, and their desired value $\underline{s}_d$ to be reached in the image. It seems useful to provide the user with systematic methods allowing to specify vision-based tasks with a natural formalism. The one we chose rely on the well-known theory of mechanisms. More precisely, the properties of the interaction matrix can be analyzed in terms of constraints in the configuration space, and represented by a formalism very similar to the one which is currently used for kinematics of contacts between solids.

It may occur that, for a peculiar value $\underline{s}^*$, certain camera motions T_c^* do not result in any change of the value $\underline{s}^*$. For example, when the camera translational motion is in the direction of the line joining a point in the scene to the optical center of the camera, then the coordinates (X, Y) of the image point remain unchanged (see Eq. (4)). More generally, at a camera position such that $\underline{s} = \underline{s}^*$, the set of camera motion (T_c^*) is characterized through:

$$\dot{\underline{s}}_{|\underline{s}=\underline{s}^*} = 0 \Rightarrow (T_c^*) = \text{ Ker } L^T_{\underline{s}|\underline{s}=\underline{s}^*} \qquad (8)$$

Thus, imposing $\underline{s} = \underline{s}_d$ is equivalent to introducing constraints on the camera motion (such that $T_c \in$ Ker $L^T_{\underline{s}|\underline{s}=\underline{s}_d}$). By analogy with the classical formalism of contacts between solids, such constraints define a virtual linkage between the camera and the considered visual features. In [3], a classification of the different vision-based tasks, based on the different virtual linkages that can be defined using a camera, is proposed. For the most classical ones (rigid, revolute, prismatic, etc.), several combinations of visual features $\underline{s}$ are presented by using the properties of their interaction matrices. In the section of this paper devoted to experimental results, we describe the realization of two classical virtual linkages, namely the prismatic/revolute one and the ball-and-socket one.

3. Control

Embedding visual servoing in the task function framework allows us to take advantage of general results helpful for analysis and synthesis of efficient closed loop control schemes taking explicitly into account redundancy in the measurements. We here only recall the obtained results, all the developments being fully described in [9] and, in the particular case of vision-based control, in [4].

We define a vision-based task, $\underline{e}_1$:

$$\underline{e}_1 = W L^{T+}_{\underline{s}|\underline{s}=\underline{s}_d} (\underline{s} - \underline{s}_d) \qquad (9)$$

where

- $\underline{s}_d$ is the desired value of the selected visual features;
- $\underline{s}$ is their current value, measured from the image at each iteration of the control loop;
- $L^{T+}_{\underline{s}|\underline{s}=\underline{s}_d}$ is the pseudo-inverse of the interaction matrix related to $\underline{s}_d$ and represents the inverse jacobian of the vision-based task;
- W is a full-rank matrix such that:

$$\text{Ker } W = \text{Ker } L^T_{\underline{s}|\underline{s}=\underline{s}_d} \qquad (10)$$

A secondary task, $\underline{e}_2$, can be combined with $\underline{e}_1$ and is expressed as the minimization of a cost function h_s, with gradient $\underline{g}_s = (\frac{\partial h_s}{\partial \underline{r}})^T$. The task function $\underline{e}$ achieves the goal of the minimization of h_s under the constraint $\underline{e}_1 = 0$ when it takes the form:

$$\underline{e} = \ W^+ \underline{e}_1 + \left(\mathbb{I} - W^+ W\right) \underline{g}_s^T \qquad (11)$$

where W^+ and $\mathbb{I} - W^+ W$ are two projection operators which guarantee that the camera motion due to the secondary task is compatible with the regulation of $\underline{s}$ to $\underline{s}_d$.

A general control scheme aimed to regulate the task function $\underline{e}$ is described in [9]. We here only present the simplified control scheme that we have used to perform the experimentations described in the next section. Similar control approaches can be found in [7] and [8].

For making $\underline{e}$ exponentially decrease and then behave like a first order decoupled system, we obtain [4]:

$$T_c = -\lambda \underline{e} - \widehat{\frac{\partial \underline{e}}{\partial t}} \qquad (12)$$

where:

- T_c is the desired camera velocity sent to the robot controller;
- λ is the proportional coefficient involved in the exponential convergence of $\underline{e}$;
- $\widehat{\frac{\partial \underline{e}}{\partial t}}$ can be written under the form:

$$\widehat{\frac{\partial \underline{e}}{\partial t}} = W^+ \widehat{\frac{\partial \underline{e}_1}{\partial t}} + (\mathbb{I} - W^+ W) \frac{\partial \underline{g}_s^T}{\partial t} \qquad (13)$$

The choice of the secondary cost function generally allows to know $\frac{\partial \underline{g}_s^T}{\partial t}$. On the other hand, vector $\widehat{\frac{\partial \underline{e}_1}{\partial t}}$ represents an estimation of the contribution of a possible autonomous target motion. If the target moves, this estimation has to be introduced in the control law in order to suppress tracking errors. It can be obtained using classical filtering techniques such as Kalman filter [2] or $\alpha - \beta - \gamma$ filter [1].

4. Results

In this section, we present experimental results obtained with a SONY camera mounted on the end-effector of a six d.o.f. AFMA robot (see Figure 1).

4.1 Positioning with Respect to a Cylinder

We want to position the camera with respect to a motionless cylinder in order that the cylinder appears centered and vertical in the image. At a right position, the cylinder equation is (see Eq. (6)):

$$h(\underline{x}, \underline{p}) = x^2 + (z - z_d)^2 - r^2 = 0 \qquad (14)$$

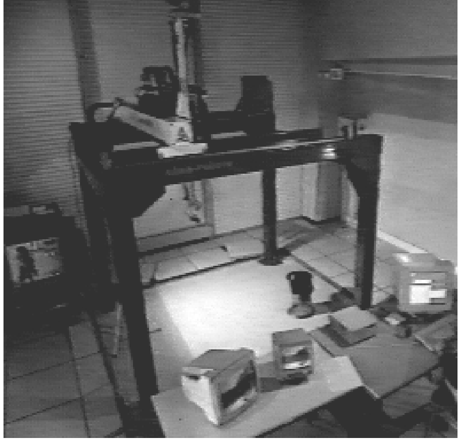


Figure 1: Experimental cell

where r is the radius of the cylinder (4 cm in the presented case) and z_d is the desired distance between the camera and the cylinder (1 m). The image of the cylinder is obtained from Eq. (6):

$$\left\{ \begin{array}{ll} D_1: & X - \rho_d = 0 \\ D_2: & X + \rho_d = 0 \end{array} \right. \quad \text{with } \rho_d = -r/\sqrt{{z_d}^2 - r^2} \tag{15}$$

By choosing for $\underline{s}$ the parameters ρ_1, θ_1, ρ_2 and θ_2 describing the current position of the two straight lines in the image, we obtain $\underline{s}_d = (\rho_d, 0, \rho_d, \pi)$. The interaction matrix related to $\underline{s}_d$ can easily be derived from Eq. (7) and is given by:

$$L^T_{|\underline{s}=\underline{s}_d} = \begin{pmatrix} \lambda_\rho & 0 & -\lambda_\rho \rho_d & 0 & -\alpha_d & 0 \\ 0 & 0 & 0 & -\rho_d & 0 & -1 \\ -\lambda_\rho & 0 & -\lambda_\rho \rho_d & 0 & \alpha_d & 0 \\ 0 & 0 & 0 & \rho_d & 0 & -1 \end{pmatrix} \tag{16}$$

with $\lambda_\rho = -z_d/({z_d}^2 - r^2)$ and $\alpha_d = 1 + {\rho_d}^2$. The set of camera motions such that the image of the cylinder remains constant, can be easily determined:

$$(T_c^*) = \text{Ker } L^T_{|\underline{s}=\underline{s}_d} = \begin{pmatrix} 0 & \alpha_d \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \lambda_\rho \\ 0 & 0 \end{pmatrix} \tag{17}$$

which means that two camera d.o.f. are not constrained by the task. As intuitively obtained, the first one consists in translating along the direction of the cylinder, and the second one consists in turning around it. The realized virtual linkage between the camera and the cylinder is called prismatic/revolute.

Since $L^T_{|\underline{s}=\underline{s}_d}$ is of full rank 4, we may choose in the definition of $\underline{e}_1$ (see Eq. (9)) $W = L^T_{|\underline{s}=\underline{s}_d}$. We therefore have:

$$\underline{e} = W^+(\underline{s} - \underline{s}_d) + (\mathbb{I}_6 - W^+W)\, \underline{g}_s^T \tag{18}$$

where:

$$W^+ = \begin{pmatrix} \lambda_\rho/2\beta & 0 & -\lambda_\rho/2\beta & 0 \\ 0 & 0 & 0 & 0 \\ -1/2\lambda_\rho\rho_d & 0 & -1/2\lambda_\rho\rho_d & 0 \\ 0 & -1/2\rho_d & 0 & 1/2\rho_d \\ -\alpha_d/2\beta & 0 & \alpha_d/2\beta & 0 \\ 0 & -1/2 & 0 & -1/2 \end{pmatrix}$$

$$(\mathbb{I}_6 - W^+W) = \begin{pmatrix} \alpha_d^2/\beta & 0 & 0 & 0 & \lambda_\rho\alpha_d/\beta & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_\rho\alpha_d/\beta & 0 & 0 & 0 & \lambda_\rho^2/\beta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

with $\beta = \lambda_\rho^2 + \alpha_d^2$.

In order to turn around the cylinder, we have specified a secondary task consisting in moving the camera with a constant velocity $V_x = 5$ cm/s along its $\vec{x}$ axis. The secondary cost function is then:

$$h_s = \frac{1}{2}\beta_x(x(t) - x_0 - V_x t)^2 \tag{19}$$

where $x(t)$ and x_0 respectively represent the current and initial camera position on its x-axis, and where β_x is a positive scalar weight. We have:

$$\underline{g}_s^T = \begin{pmatrix} \beta_x(x(t) - x_0 - V_x t) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \frac{\partial \underline{g}_s^T}{\partial t} = \begin{pmatrix} -\beta_x V_x \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{20}$$

and the desired camera velocity T_c is finally given by:

$$T_c = -\lambda\, \underline{e} + \begin{pmatrix} \beta_x V_x \alpha_d^2/l \\ 0 \\ 0 \\ 0 \\ \beta_x V_x \alpha_d \lambda_\rho / l \\ 0 \end{pmatrix} \tag{21}$$

Now, by choosing $\beta_x = l/\alpha_d^2$, the translational camera velocity along $\vec{x}$ axis has the desired value V_x when $\underline{e} = 0$.

Experimental results are depicted on Figure 2: Figure 2.a represents the initial image acquired by the camera and the selected cylinder; Figure 2.b represents the image acquired by the camera after the realization of the vision-based task. The image processing step consists in tracking the two straight lines along the image sequence. Thanks to a dedicated image processing board EDIXIA, the extraction, maintenance and tracking of the contour segments are achieved in less than 40 ms, so that the rate of the control law is equal to the video rate (25 Hz).

The plots shown on Figure 2.e, Figure 2.f and Figure 2.g respectively represent the evolution, at each iteration of the control law, of the four component of $\underline{s}-\underline{s}_d$ (in cm and dg), of $\|\underline{s}-\underline{s}_d\|$ and of the six components of the camera velocity T_c (in cm/s and dg/s). Let us point out the exponential decay of these evolutions during the convergence phase (it. 0 to 200). The secondary task has been introduced in the control law after this phase ($V_x = 5$ cm/s from it. 200 to 380, $V_x = -5$ cm/s from it. 420 to 630). Let us note that the effect of the camera translational motion is perfectly compensated by a rotational motion (see Eq. (21)) such that the cylinder always appears at its specified position in the image. Figure 2.c and Figure 2.d represent two images acquired by the camera during the trajectory tracking, which allows the camera to observe the cylinder under different constrained view points.

4.2 Positioning with Respect to a Sphere

The task here consists in positioning the camera at a desired distance z_d (50 cm) from a sphere of radius r (1.8 cm), such that it appears as a centered circle in the image (see Figure 3 where the initial and final images acquired by the camera are depicted). The image processing step, again performed at the video rate, now consists in extracting and tracking the center of gravity (X_c, Y_c) and the normalized inertial moments $(\mu_{20}, \mu_{11}, \mu_{02})$ of the projected ellipse along the image sequence. In that case, we have $\underline{s} = (X_c, Y_c, \mu)$ with $\mu = (\mu_{20}+\mu_{02})/2$, $\underline{s}_d = (0, 0, \mu_d)$ with $\mu_d = r^2/(z_d^2 - r^2)$ and [3]:

$$L^T_{|\underline{s}=\underline{s}_d} = \begin{pmatrix} -1/z_d & 0 & 0 & 0 & -1-\mu_d & 0 \\ 0 & -1/z_d & 0 & 1+\mu_d & 0 & 0 \\ 0 & 0 & 2\mu_d/z_d & 0 & 0 & 0 \end{pmatrix} \tag{22}$$

The set of camera motion (T_c^*) is given by:

$$(T_c^*) = \begin{pmatrix} 0 & z_d(1+\mu_d) & 0 \\ 0 & 0 & z_d(1+\mu_d) \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{23}$$

which means that all camera rotations are free, both of them being compensated by translational motion in order to turn around the sphere. Such constraints between the camera and the sphere define a classical virtual linkage, called ball-and-socket linkage.

In the design of the vision-based task, we can again choose $W = L^T_{|\underline{s}=\underline{s}_d}$. We thus have $\underline{e_1} = \underline{s} - \underline{s}_d$, and, since no secondary task was specified in this experiment, the desired camera velocity takes the form:

$$T_c = -\lambda W^+ \underline{e}_1 \tag{24}$$

The plots shown in Figure 3 represents the time evolution of $\underline{s}-\underline{s}_d$ (in pixels), $\|\underline{s}-\underline{s}_d\|$ and T_c during the convergence of the vision-based task.

5. Conclusion

In this paper, we have presented how it is possible to use in visual servoing more complex visual features than the usual coordinates of points. These new visual features are based on geometrical primitives such as lines, spheres and cylinders. Using such features enables to realize a large variety of robotics tasks. The real time experimental results we have presented show the efficiency and robustness of our control law to perform vision-based task conbined with trajectory tracking.

References

[1] P.K. Allen, A. Timcenko, B. Yoshimi, P. Michelman, "Automated Tracking and Grasping of a Moving Object with a Robotic Hand-Eye System", *IEEE Transactions on Robotics and Automation*, Vol. 9, n.2, pp. 152-165, April 1993.

[2] F. Chaumette, A. Santos, "Tracking a moving object by visual servoing", *Proc. of the 12th IFAC World Congr.*, Vol. 9, pp. 409-414, 1993.

[3] F. Chaumette, P. Rives, B. Espiau, "Classification and realization of the different vision-based tasks", in Visual Servoing, K. Hashimoto editor, World Scientific Series in Robotics and Automated Systems, Vol 7, pp. 199-228, Singapore, 1993.

[4] B. Espiau, F. Chaumette, P. Rives "A New Approach to Visual Servoing in Robotics", *IEEE Transactions on Robotics and Automation*, Vol. 8, n. 6, pp. 313-326, June 1992.

[5] J.T. Feddema, C.S.G. Lee, O.R. Mitchell, "Weighted Selection of Image Features for Resolved Rate Visual Feedback Control", *IEEE Transactions on Robotics and Automation*, Vol. 7, n. 1, pp. 31-47, Feb. 1991.

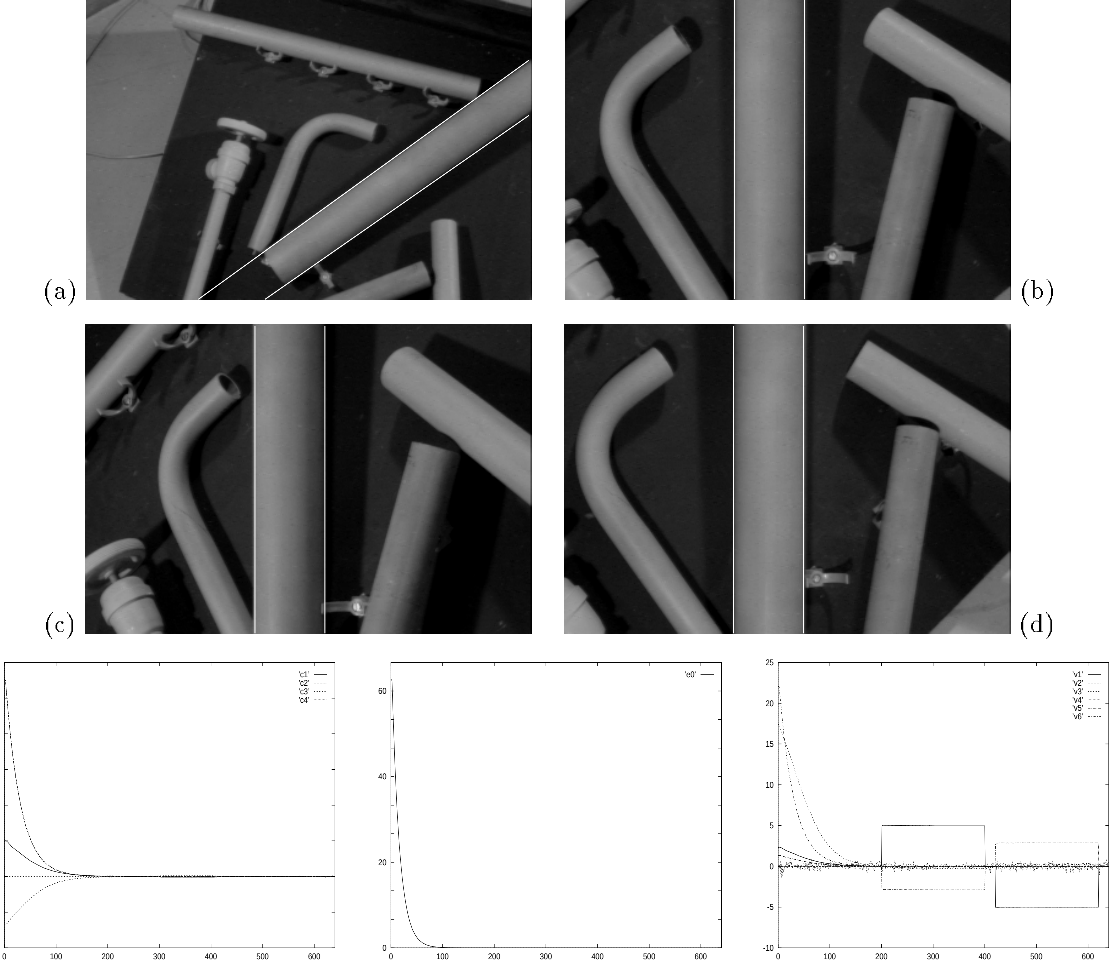
[6] F. Harashima, H. Hashimoto, T. Kubota, "Sensor-based Robot Control Systems", *IEEE Int. Workshop on Intelligent Motion Control*, Vol. 1, pp. 1-10, Istanbul, Turkey, August 1990.

[7] K. Hashimoto, "Visual Servoing", World Scientific Series in Robotics and Automated Systems, Vol 7, Singapore, 1993.

[8] N. Papanikolopoulos, P.K. Khosla, T. Kanade, "Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision", *IEEE Trans. on Robotics and Automation*, Vol. 9, n. 1, pp. 14-35, February 1993.

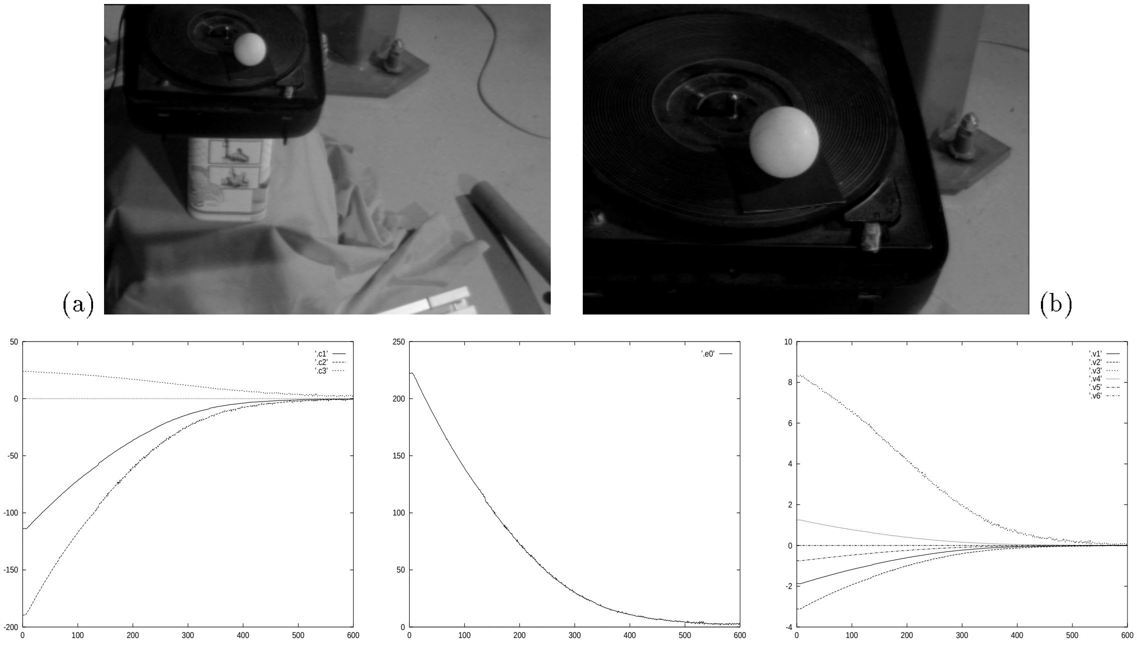
[9] C. Samson, B. Espiau, M. Le Borgne, *Robot Control: the Task Function Approach.* Clarendon Press, Oxford, England, 1991.

[10] L. E. Weiss, A.C. Sanderson, C.P. Neuman, "Dynamic Sensor-Based Control of Robots with Visual Feedback", IEEE Journal of Robotics and Automation", Vol. 3, n. 5, pp. 404-417, October 1987.



(e) : $\underline{s} - \underline{s}_d$ (f) : $||\underline{s} - \underline{s}_d||$ (g) : T_c

Figure 2: Positioning with respect to a cylinder.



(c) $\underline{s} - \underline{s}_d$ (d) $||\underline{s} - \underline{s}_d||$ (e) T_c

Figure 3: Positioning with respect to a sphere.