Proceedings of the 1991 IEEE
International Conference on Robotics and Automation
Sacramento, California - April 1991

Positioning of a Robot with respect to an Object, Tracking it and Estimating its Velocity by Visual Servoing

François Chaumette*, Patrick Rives, Bernard Espiau*****

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

Abstract

This paper describes an attractive application of the so-called "visual servoing" approach to robot positioning with respect to an object and to target tracking.

After having briefly recalled how the task function approach may be applied to tasks which include the use of visual features, we give a simplified control expression which explicitly takes into account the case of moving objects. Estimating the target velocity while performing the tracking control finally leads to a kind of adaptative control scheme. We then consider the specific case of a "square" target and derive all the components of the control scheme. Finally, we present some experimental results performed at the video rate in an experimental system composed of a camera mounted on the end effector of a six d.o.f. robot.

1 Introduction

Recent advances in vision sensor technology and vision processing now allow the effective use of vision data in the control loop of a robot. The main benefits are the followings. Concerning robotics applications, this enables to handle uncertainties and/or variations of the environment (for example, to compensate for small positioning errors, to grasp objects moving on a conveyor belt,...). Concerning vision aspects, it is then possible to control the camera motion in order to improve recognition, localization or inspection of the camera environment.

We can distinguish two different approaches in vision based control [11]: the first one is based on a 3D position servoing, the second one on a visual feature servoing:

• **The Look and Move approach**: the task consists in positioning of the camera to a desired position $\bar{r}^*$, which signifies desired location and attitude between the camera and a frame linked to the environment (see Figure 1). At each iteration of the control loop, an estimate $\hat{\bar{r}}$ of the actual position has to be achieved from the vision data. This scheme works in open loop with respect to vision data and can not take into account sudden or large variations of the environment, inaccuracies and uncertainties occuring during the processing. Furthermore, such an approach needs to perfectly identify all the 3D models (sensor geometry, environment and robot models).

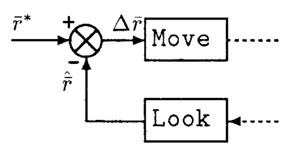


Figure 1: Look and Move

• **The visual servoing approach**: in this approach, the task is directly specified in terms of regulation in the image (see Figure 2). This requires the design of a set s of visual features which are sufficient and relevant for the completion of the task. Thus, a closed loop can be really performed from vision data, which allows to compensate for the perturbations using a robust control scheme. The work described in this paper deals with such an approach.

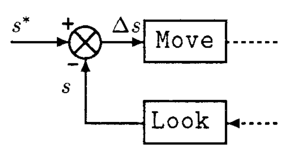


Figure 2: Visual servoing

2 General framework

2.1 The task function approach

For a given task, let us recall that we have to choose in visual servoing a set s of visual features suited for achieving the task. Then, we can define a task function vector $e\ (\bar{r}(t))$ such that:

$$e\ (\bar{r}(t)) = C\ (s\ (\bar{r}(t)) - s^*) \tag{1}$$

where

- s^* can be considered as a reference image target to be reached in the image frame,
- $s\ (\bar{r}(t))$ is the value of the visual features currently observed by the camera, which only depends on the position between the camera and the scene,
- C is a constant matrix which allows, for robustness issues, to take into account more visual features than necessary, and which will be fixed afterwards.

• **Remark:** The task function approach is a very general one which allows to perform classical robotics tasks as well as sensor based tasks [8]. A general approach of the control problem is presented in [9].

When stating the control problem as an output regulation problem, it appears that the concerned task is perfectly achieved if $e\ (\bar{r}(t)) = 0$. We may emphasize robustness issues with respect to model uncertainties by expressing this regulation problem as the problem of minimizing $\|e\ (\bar{r}(t))\|$. Although all the theory was developped in a dynamical framework [9], [3], with true joint torque control, we here assume for simplicity that the velocity T_c of the camera may be considered as a "control" vector. Furthermore, let us consider in a first step that the objects of the scene are motionless. We thus may choose the following control law:

$$T_c = -\lambda\ e\ (\bar{r}(t)) \tag{2}$$

with $\lambda > 0$. Indeed, we have:

$$\dot{e} = \frac{\partial e}{\partial \bar{r}}\ T_c = -\lambda\ CL^T\ e \tag{3}$$

where $L^T = \frac{\partial s}{\partial \bar{r}}$ is called the interaction matrix associated to s and is similar to a jacobian ($\dot{s} = L^T T_c$). An exponential convergence will thus be ensured under the sufficient condition:

$$CL^T > 0 \tag{4}$$

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

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

$$C = {L^T}^+_{|s=s^*} \tag{5}$$

• **Remark:** As we can see on that last equation, it is necessary to know a model of the interaction matrix of visual features for using them in visual servoing. In [10] and [6], an experimental learning approach is proposed. It is also often possible to explicitly compute this matrix: the results for points and segments are given in [4] and [5], for lines in [7]. In [2] and [3], a general method for computing the interaction matrix of any visual features defined upon geometrical primitives is proposed and explicit results are given for the parameters describing the projection in the image of circles, spheres and cylinders.

2.2 Case of a mobile object

Let us now consider that the primitives constituting the scene move with a velocity T_o with respect to the camera frame. We now have:

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

The control law (2) leads in this case to a tracking error, the size of which decreases with λ. For suppressing this tracking error, we may introduce in the control law a model $\widehat{\frac{\partial e}{\partial t}}$ of $\frac{\partial e}{\partial t}$ and we finally obtain:

$$T_c = -\lambda\ e - \left(\widehat{\frac{\partial e}{\partial \bar{r}}}\right)^+ \widehat{\frac{\partial e}{\partial t}} \tag{7}$$

$\widehat{\frac{\partial e}{\partial \bar{r}}}$ is obviously chosen as $CL^T_{|s=s^*}$. $\frac{\partial e}{\partial t}$ may also be estimated on-line and re-injected in (7). This leads to an adaptative control scheme of indirect type. For example, when taking a model with constant velocity, it is possible to show that an adequate estimation scheme is simply:

$$\dot{\widehat{\frac{\partial e}{\partial t}}} = \mu\ (t)\ e \tag{8}$$

• **Remark:** Taking $\mu\ (t) = \mu$ and considering the discrete time case, $\left(\widehat{\frac{\partial e}{\partial t}}\right)_{k+1}$ is given at the iteration $k+1$ by:

$$\begin{aligned}\left(\widehat{\frac{\partial e}{\partial t}}\right)_{k+1} &= \left(\widehat{\frac{\partial e}{\partial t}}\right)_k + \mu\ e_k \text{ with } \left(\widehat{\frac{\partial e}{\partial t}}\right)_0 = 0 \\ &= \mu \sum_{j=0}^{k} e_j \end{aligned} \tag{9}$$

As we can see on that last equation, the estimation algorithm takes effect as a simple integrator.

Furthermore, since we are provided with an estimate of $\widehat{\frac{\partial e}{\partial t}}$, we can obtain an estimate $\widehat{T_o}$ of the object velocity through:

$$\widehat{T_o} = -(CL^T_{|s=s^*})^+ \, \widehat{\frac{\partial e}{\partial t}} \qquad (10)$$

Indeed, we have $\dot{e} = CL^T \; T_c + \frac{\partial e}{\partial t} = CL^T \; (T_c - T_o)$.

• **Remark:** If the matrix $CL^T_{|s=s^*}$ is not of full rank, an estimation of the components of the object velocity which belong to the null space Ker $(CL^T_{|s=s^*})$ is impossible. Indeed, if $s = s^*$, such motion does not modify the value in the image of the visual features used in e.

3 Application to a positioning task

Let us suppose that it is wished to set the camera with respect to a plane object which may be characterized by four points defining a square. Adequate visual features for this positioning task are the image coordinates of the four points: $s = (X_1, \cdots, X_4, Y_1, \cdots, Y_4)$.

Let us consider a camera as a perspective projection model (see Figure 3). Without loss of generality, the focal length is assumed to be equal to 1, such that any point m with coordinates $(x\,y\,z)$ is projected on the image plane as a point M with coordinates $(X\,Y)$ where:

$$X = x/z \;,\; Y = y/z \qquad (11)$$

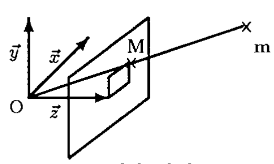


Figure 3: Model of the camera

If the goal position is such that the image plane is parallel to the object plane with the four image points forming a centred square, the image feature is obtained using (11): $s^* = (-a, a, a, -a, a, a, -a, -a)$ where $a = l/2z^*$, l being the vertex length and z^* the final wished range.

As we have seen in the previous section, we need to compute the interaction matrix associated to s^*. By differentiating (11), we can derive the well known equation relating optical flow measurement to 3D structure and camera motion:

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = \begin{pmatrix} -1/z & 0 & X/z & XY & -1-X^2 & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{pmatrix} T_c \qquad (12)$$

The interaction matrix L^T associated to s^* is thus immediately obtained through (12):

$$L^T_{|s=s^*} = \begin{pmatrix} l_1 & 0 & -a/z^* & -a^2 & -l_2 & a \\ l_1 & 0 & a/z^* & a^2 & -l_2 & a \\ l_1 & 0 & a/z^* & -a^2 & -l_2 & -a \\ l_1 & 0 & -a/z^* & a^2 & -l_2 & -a \\ 0 & l_1 & a/z^* & l_2 & a^2 & a \\ 0 & l_1 & a/z^* & l_2 & -a^2 & -a \\ 0 & l_1 & -a/z^* & l_2 & a^2 & -a \\ 0 & l_1 & -a/z^* & l_2 & -a^2 & a \end{pmatrix} \qquad (13)$$

with $l_1 = -1/z^*$ and $l_2 = 1 + a^2$. Since $L^T_{s=s^*}$ is of full rank, it is possible to compute its pseudo inverse C such that $CL^T_{s=s^*} = \mathbb{I}_6$:

$$C = \begin{pmatrix} c_1 & c_1 & c_1 & c_1 & -c_2 & c_2 & -c_2 & c_2 \\ -c_2 & c_2 & -c_2 & c_2 & c_1 & c_1 & c_1 & c_1 \\ -c_3 & c_3 & c_3 & -c_3 & c_3 & c_3 & -c_3 & -c_3 \\ -c_4 & c_4 & -c_4 & c_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_4 & -c_4 & c_4 & -c_4 \\ c_5 & c_5 & -c_5 & -c_5 & c_5 & -c_5 & -c_5 & c_5 \end{pmatrix} \qquad (14)$$

$$\text{with} \left\{ \begin{array}{l} c_1 = -z^*/4 \;,\; c_2 = z^*(1+a^2)/4a^2 \\ c_3 = z^*/8a \;,\; c_4 = 1/4a^2 \;,\; c_5 = 1/8a \end{array} \right.$$

At this step, the control law given by (7) in order to perform the positioning task over a square is completed.

• **Remark:** If the length of square vertex is unknown, it is not possible to set the final range z^* from camera to objet. Positioning at an unknown range is however possible by setting the desired square length in the image, $2a$, and taking as coefficients of C:

$$\begin{array}{l} c_1 = -1/4 \;,\; c_2 = (1+a^2)/4a^2 \\ c_3 = c_5 = 1/8a \;,\; c_4 = 1/4a^2 \end{array} \qquad (15)$$

Indeed, the convergence condition (4) is ensured for $s = s^*$:

$$(CL^T)_{|s=s^*} = \begin{pmatrix} \mathbb{I}_3/z^* & 0 \\ 0 & \mathbb{I}_3 \end{pmatrix} > 0 \qquad (16)$$

4 Experimental results

4.1 Description

This task has been implemented on an experimental testbed including a CCD camera mounted on the end effector of a six d.o.f. robot (Figure 4).

A camera calibration step allowing to obtain the model of Figure 3 was done, and the transformation matrix from the camera frame to a frame linked to the end effector was identified using methods given in [1]. This allows the desired velocity T_c of the camera to be transformed in a desired joint velocity vector

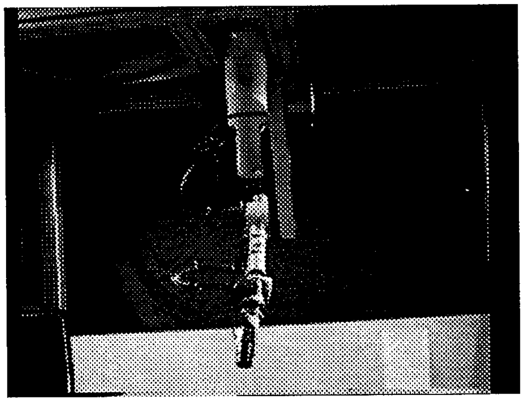


Figure 4: Experimental cell

$\dot{q}_c$ using the inverse jacobian matrix of the robot J^{-1} ($\dot{q}_c = J^{-1}\ T_c$). The computation of J^{-1} is realized on a 68020-based dedicated board with a sampling rate of 5 ms. The useful part of the scene is a set of four white coplanar disks on a dark background. Owing to this simplicity, mass centres of image disks and velocity screw T_c are computed in less than 20 ms, which ensures that the video rate is respected.

4.2 Positioning with respect to a motionless square

Figure 5 shows an image sequence taken during a positioning task. The dimensions of the square are assumed to be known (a = 3 cm) and z^* is set equal to 25 cm. The used value of λ is 0.1 and, since the object is motionless, we have $\mu = 0$. The plottings of the time variation of the components, in pixels, of $s - s^*$ (on the left) and T_c, in cm/s and dg/s, (on the right) show the stability and the convergence of the control law.

Of course, we can specify an other position to be reached between the camera and the square. For instance, we can choose to position the camera to 20 cm of the object with an attitude of 45 degrees. We then have:

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & -\sqrt{2}/2 & \sqrt{2}/2 & 0.2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (17)$$

where M is the desired homogeneous transformation matrix between the camera frame and the object frame. Since the dimensions of the square are known, it is possible to compute the coordinates of the points to be reached in the image, the pseudo inverse C of the interaction matrix $L^T_{|s=s^*}$, and to apply the control law (7). The results are given on Figure 6, structured as previously.

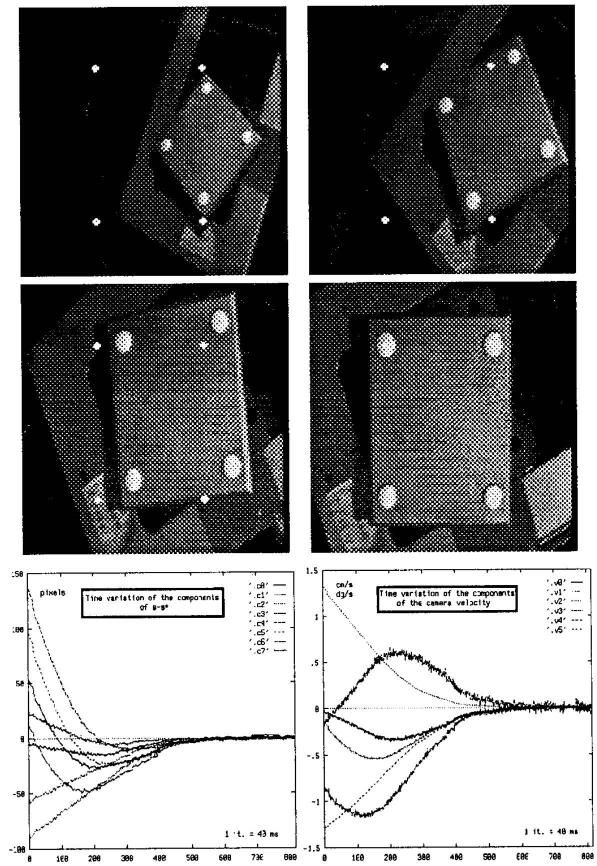


Figure 5: Positioning with respect to a square

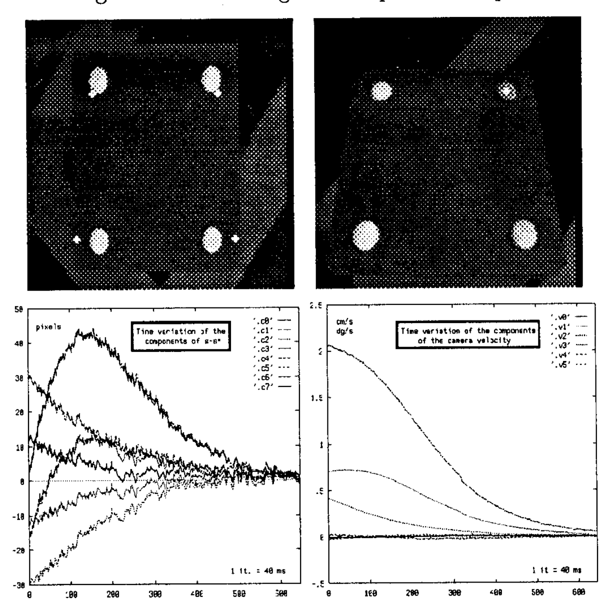


Figure 6: Reaching an other position

4.3 Following the square and estimating its velocity

From this position, the object moves with a constant velocity V_1 of 0.4 cm/s in the direction described Figure 7, then stops and moves in the opposite direction with a constant velocity V_2 of 1.7 cm/s. Since we know the transformation matrix between the camera frame and the object frame, we can easily obtain (see Figure 8a) the time variation of the components of the object velocity T_o expressed in the camera frame.

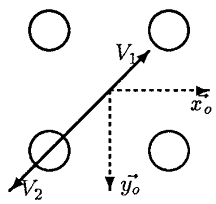


Figure 7: Motion of the square (in the object frame)

When no estimation scheme of $\frac{\partial e}{\partial t}$ (i.e. $\mu = 0$) is used, we logically observe, on the left plot of Figure 8b which represents the time variation of $s - s^*$, a tracking error, constant for a constant velocity of the object, increasing with the motion size and which disappears after some iterations when the object stops. The time variation of the camera velocity (on the right plot) approximatively looks like the object velocity.

If we now introduce the estimation of $\frac{\partial e}{\partial t}$ given by (9) with $\mu = 0.1$, we observe on Figure 8d structured as previously, that the tracking error is almost zero for a constant velocity of the object. Owing to the effect of the estimator, an overshoot occurs during the sharp variations of motion of the object.

Finally, let us remark that, since we have $C L^T_{|s=s^*} = \mathbb{I}_6$, the estimated velocity $\widehat{T_o}$ of the object is simply given by (see (10)):

$$\widehat{T_o} = -\frac{\widehat{\partial e}}{\partial t} \quad (18)$$

which is plotted on the Figure 8c and which we can compare with the real one given on the Figure 8a. Let us emphasize that none assumption on the direction or on the norm of the object velocity has been done.

5 Concluding remarks and future issues

As we try to show in this paper, visual servoing approach seems to be an original and powerful way for solving some "classical" problems in robotics and 3D vision: positioning of a robot with respect to the environment, tracking a 3D moving object, 3D motion estimation and so on. Characterizing this approach can be made in terms of robustness and generality. It may then be shown that some ill-conditionned problems becomes well-posed. In recent papers, [3], [2], we proposed a general way to compute the interaction matrix (which is the central point of the approach) from any 3D geometric primitives which can be expressed as a parametric equation. The knowledge of this interaction matrix embedded in a general robot control formalism, the task function approach [9], allows to derive a robust closed loop control scheme for realizing positioning tasks with respect to static 3D objects. In this paper, the previous approach was extended in the case of moving objects by introducing an estimation scheme which allows to take into account the velocity of the object. A simple estimation scheme was tested given good results which are discussed in the paper. As a secondary but important result, we showed that such an approach provides with an estimate of the instantaneous velocity of the object.

From this approach, several issues remain to be investigated: firstly, the accuracy of the object velocity estimate might be improved by taking into account positioning and tracking errors explicitly. Secondly, we have to consider the case corresponding to an inital position of the camera far from the desired one at the beginning of the tracking. Finally, a natural extension will be to revisit the *structure and motion problem* from this new point of view and to embed it in a more general scheme based on an *active vision* approach.

References

[1] F. Chaumette, P. Rives : *Réalisation et calibration d'un système expérimental de vision composé d'une caméra embarquée sur un robot manipulateur*, Research report, n. 994, INRIA, France, March 1989.

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

[3] B. Espiau, F. Chaumette, P. Rives: *A new approach to visual servoing in robotics*, Research Report, n. 542, IRISA/INRIA Rennes, France, July 1990.

[4] J. T. Feddema, C. S. G. Lee and O. R. Mitchell: *Automatic selection of image features for visual servoing of a robot manipulator*, Conf. IEEE Ro-

botics and Automation, Scottsdale, Arizona, USA, May 14-19, 1989.

[5] J. T. Feddema, O. R. Mitchell: *Vision-Guided Servoing with Feature - Based Trajectory Generation*, IEEE Transaction on Robotics and Automation, Vol. 5, n.5, October 1989.

[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.

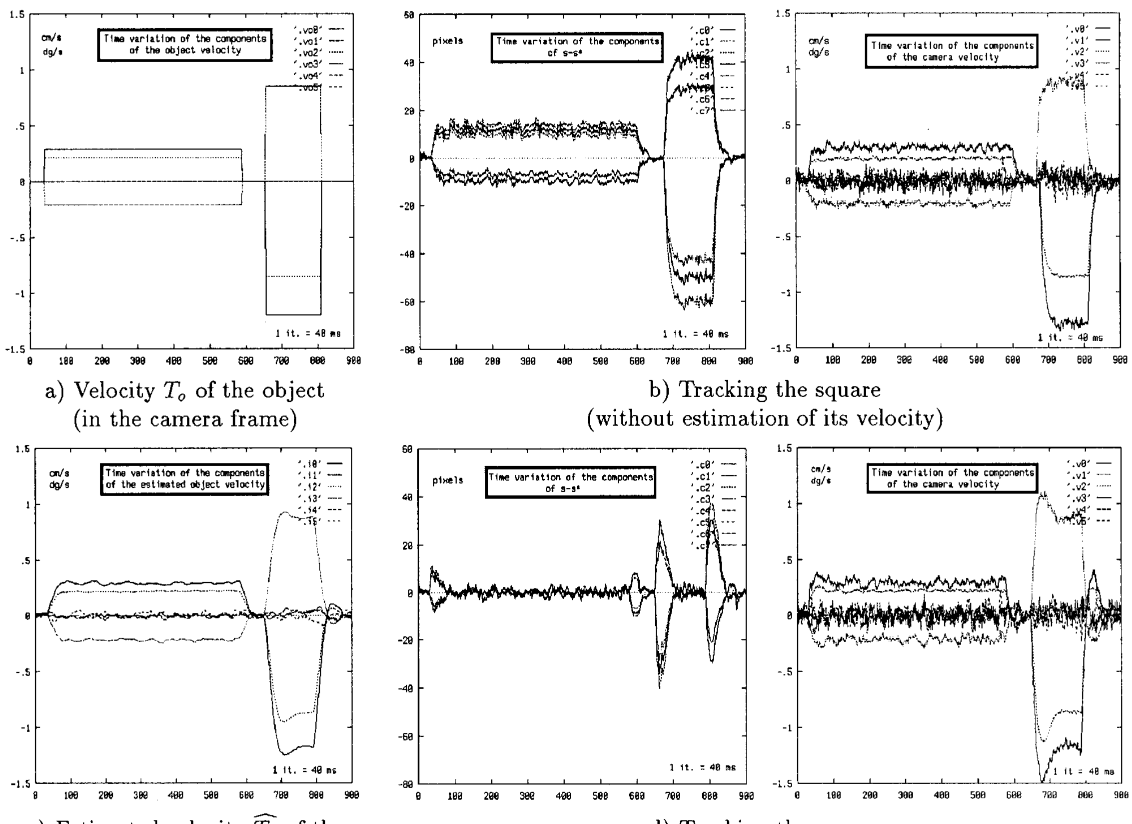
[7] P. Rives, F. Chaumette, B. Espiau: *Visual Servoing Based on a Task Function Approach*, First International Symposium on Experimental Robotics, Montréal, Canada, June 1989.

[8] C. Samson, B. Espiau: *Application of the Task Function Approach to Sensor-Based-Control of Robot Manipulators*, IFAC, Tallin, USSR, July 1990.

[9] C. Samson, B. Espiau, M. Le Borgne: *Robot Control: the Task Function Approach*, Oxford University Press, fall 1990.

[10] L. E. Weiss: *Dynamic Visual Servo Control of Robots. An Adaptive Image based Approach*, Technical Report, CMU-RI-TR-84-16; Carnegie Mellon, 1984.

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



a) Velocity T_o of the object (in the camera frame)

b) Tracking the square (without estimation of its velocity)

c) Estimated velocity $\widehat{T_o}$ of the object (in the camera frame)

d) Tracking the square (with estimation of its velocity)

Figure 8: Tracking the square