Uncalibrated Visual Tasks via Linear Interaction

Carlo Colombo

ARTS Lab
Scuola Superiore Sant'Anna
Via Carducci 40
56127 Pisa, ITALY

tel +39-50-883207
fax +39-50-883215
columbus@shamash.sssup.it

James L. Crowley

LIFIA-IMAG
Institut National Polytéchnique de Grenoble
46, Avenue Félix Viallet
38031 Grenoble Cedex, FRANCE

tel +33-76-574655
fax +33-76-574602
jlc@imag.fr

Abstract

In this paper, we propose an approach for the control and layering of space-time continuous visual tasks with an uncalibrated camera. The approach is based on the bi-dimensional appearance of the objects in the environment, it allows the design of both reactive (or reflexive) and active (purposive) tasks, and takes explicitly into account independent object motions. A linear model of camera-object interaction is embedded in the control scheme, which dramatically simplifies visual analysis and control by reducing the size of visual representation.

The implementation of three visual tasks of increasing complexity, obtained with the proposed scheme and based on active contour analysis and polynomial planning of image contour transformations, is described and discussed. Both simulations and real-time experiments with a robotic eye-in-hand configuration are shown, which demonstrate the feasibility of the approach for applications in the fields of visual navigation, active exploration and perception, and man-robot interaction.

KEYWORDS: Active and Real-Time Vision, Vision-Guided Robotics

CORRESPONDING AUTHOR: Carlo Colombo

This work has been submitted for publication in the
Fourth European Conference on Computer Vision ECCV'96,
14–18 April 1996, University of Cambridge, England

Contents

List of Figures

Uncalibrated Visual Tasks via Linear Interaction

Abstract

In this paper, we propose an approach for the control and layering of space-time continuous visual tasks with an uncalibrated camera. The approach is based on the bi-dimensional appearance of the objects in the environment, it allows the design of both reactive (or reflexive) and active (purposive) tasks, and takes explicitly into account independent object motions. A linear model of camera-object interaction is embedded in the control scheme, which dramatically simplifies visual analysis and control by reducing the size of visual representation.

The implementation of three visual tasks of increasing complexity, obtained with the proposed scheme and based on active contour analysis and polynomial planning of image contour transformations, is described and discussed. Both simulations and real-time experiments with a robotic eye-in-hand configuration are shown, which demonstrate the feasibility of the approach for applications in the fields of visual navigation, active exploration and perception, and man-robot interaction.

1 Introduction

As available computing power has increased, it has become possible to build and experiment vision systems that operate continuously. A crucial problem in a continuously operating vision system is dealing with the very large quantity of ambiguous and noisy data provided by cameras. An often overlooked property of the human visual system is that the perceptual processes are serial and highly restrictive about what data is processed at each instant. The human visual system can be seen as a pipeline of filters for eliminating unnecessary information. *Active vision* systems take inspiration from this "filtering" principle to limit the amount of data which must be attended to in order to provide a response within a fixed delay [1–3]. Active vision can be defined as *control of cameras and control of processing to aid the observation of the external world* [4]. According to this principle, amelioration of perception is the result of combining selective sensing strategies and motor control techniques into semi-autonomous *visual tasks*. The simplest visual tasks can be regarded as reactive transformations from perception to action [5], where motor actions are a direct consequence of incoming visual data. A number of reactive visual tasks such as saccadic shifts and target tracking has been recently implemented in a robot head [6]. Other tasks involve the purposive (active) planning of visuo-motor strategies, thereby requiring a deeper knowledge of the visual environment than reactive tasks. An example of active visual task is the recognition strategy recently proposed in [7], which adopts deliberate camera displacements so as to disambiguate object views.

Much work has been done in the last few years on the design of specific architectures for the control of camera motion in the visual environment, or *visual servoing* [8]. A new approach to visual servoing has been proposed in [9], in which the visual loop is closed at the image level instead than in space, with significant improvements in terms of decreased sensitivity to camera calibration and kinematic modeling uncertainties. Active gaze control based on the learning of visual appearance has been described in [10]. A theoretical framework which establishes a trait d'union between amelioration of visual perception and control task sensitivity optimization has been recently proposed in [11].

The human visual system provides a large number of examples in which visual tasks interact and cooperate together. An example of cooperation, at the reactive level, is when the visual system performs saccadic shifts so as to recover from pursuit errors on a target—[12], for an active vision implementation see [13]. An example of cooperation between active and reactive tasks is the first, reactive saccade which precedes an active recognition saccadic sequence, or "scanpath"—[14], for an active vision implementation see [15].

The problem of the integration of several tasks is of key importance for the design of complex visual systems. A general framework for the integration of reactive visual processes was presented recently, in which the problem of the hierarchical organization of control processes was addressed [16]. Layered architectures for the organization of generic robotic tasks and behaviors are also discussed in [17, 18].

In this paper, we present an approach for implementing space-time continuous tasks with an uncalibrated camera. The approach is based on bi-dimensional (2D) visual appearance, includes both feedforward and feedback sensori-motor strategies, and explicitly considers independent object motions. The mechanism of task layering is explained in terms of loop bandwidth of each control task. The proposed approach is equally usable for the design of active and reactive tasks, which are operationally defined in terms of presence/absence of a planning module. The assumption of a linear model of camera-object interaction, once

that its intrinsic ambiguities are solved, dramatically simplifies object representation and visual control. We discuss a system implementation with a manipulator-mounted camera which uses active contours as image primitives and includes three different tasks: fixation (reactive), mimicking object motion (reactive), and relative positioning (active). In the case of active positioning, we show how to generate affine position transformations by coupling a polynomial planning strategy and visual servoing from active contours. The techniques described can have applications in visual navigation by means of natural landmarks, active exploration and perception, and man-robot interaction.

The paper is organized as follows. In Sect. 2, we give an overview of our approach, focusing on the control aspects for the design of a generic visual task. Visual modeling and measurement issues, which are actually independent of control architecture, are introduced in Sect. 3. In Sect. 4 we illustrate the implementation and combination of three different visual tasks. We show finally experiments and results with an eye-in-hand robotic setup in Sect. 5.

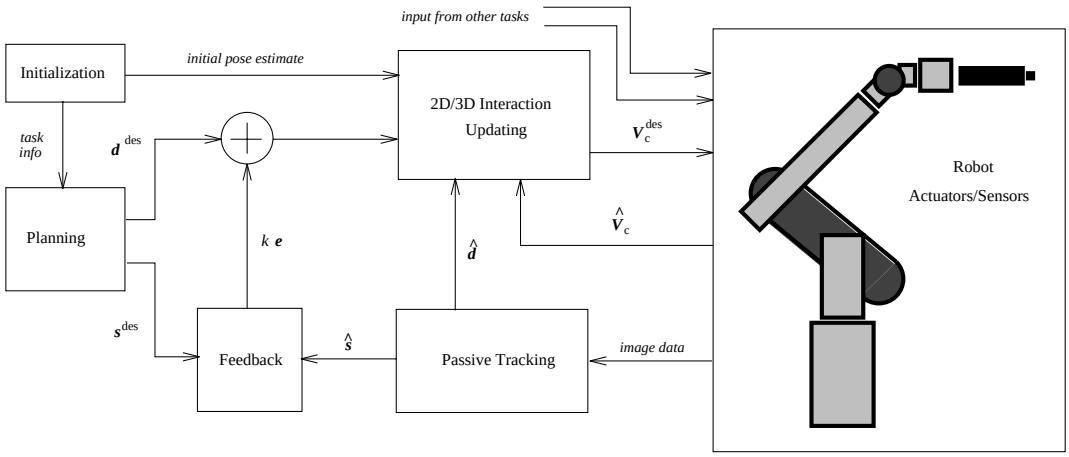

Figure 1: Control scheme for a generic visual task.

2 Overview and Control Strategy

Changes in viewpoint determined by relative motion of camera and object influence the raw image data according to the nature of both camera projection and object shape. Given a model of camera-object interaction, a *visual representation* $\{\boldsymbol{s}, \boldsymbol{d}\}$ can be defined where, at each time t:

- $\boldsymbol{s}(t)$ is an m-dimensional parameterization of object's visual appearance;
- $\boldsymbol{d}(t)$ is a suitable set of n image differential parameters that can be chosen in order to describe 2D changes of image appearance caused by the 3D relative velocity twist of camera and object.

The $n \times 6$ *interaction matrix* $\mathcal{L}$ encodes the differential transformation from relative twist $\Delta \boldsymbol{V}$ to appearance changes:

$$\boldsymbol{d} = \mathcal{L}\,\Delta \boldsymbol{V} = \mathcal{L}\,(\boldsymbol{V}_c - \boldsymbol{V}_o), \tag{1}$$

where $\boldsymbol{V}_c^{\mathrm{T}} = [\boldsymbol{T}_c^{\mathrm{T}}\ \boldsymbol{\Omega}_c^{\mathrm{T}}]$ is the velocity twist of the camera and $\boldsymbol{V}_o^{\mathrm{T}} = [\boldsymbol{T}_o^{\mathrm{T}}\ \boldsymbol{\Omega}_o^{\mathrm{T}}]$ is the velocity twist of the object. (The concept of interaction matrix is introduced in [9] in the particular case of $\boldsymbol{d} = \dot{\boldsymbol{s}}$.)

Any visual task can be described in the image plane as a desired evolution $\boldsymbol{s}^{\mathrm{des}}(t)$ of object appearance toward a goal one. In differential terms, the task can be represented in a synthetic way by a trajectory $\boldsymbol{d}^{\mathrm{des}}(t)$, the desired evolution of visual appearance. This is nonzero only in the case of active tasks, for which a purposive control task can be defined. In the case of a reactive task, this term vanishes identically, control resulting in a pure regulation-to-zero scheme. According to the above, reactive tasks are regarded as particular active tasks, which lack of a planning module.

According to active vision principles, the visual representation will be chosen as the minimal one to succesfully accomplish the task at hand. Such visual representation can be estimated as $\{\widehat{\boldsymbol{s}}^{\mathrm{T}}, \widehat{\boldsymbol{d}}^{\mathrm{T}}\}$ through visual analysis, according to a tracking process in the image plane, which we refer to as *passive tracking*. Passive tracking, which takes place also when the camera is motionless, closely resembles voluntary attentional shifts in the human visual system, which occur without changes in viewpoint [12].

Once that the structure of $\mathcal{L}$ has been identified, we adopt the following control strategy, which makes use of both feedforward and feedback information:

$$\boldsymbol{V}_c^{\mathrm{des}} = \widehat{\boldsymbol{V}}_o + \mathcal{L}^{+}(\boldsymbol{d}^{\mathrm{des}} + k\,\boldsymbol{e}(\widehat{\boldsymbol{s}}, \boldsymbol{s}^{\mathrm{des}})), \tag{2}$$

where $\boldsymbol{V}_c^{\mathrm{des}}$ is the required motion of the camera, $\widehat{\boldsymbol{V}}_o$ is an estimate of object motion, $\boldsymbol{e}(\widehat{\boldsymbol{s}}, \boldsymbol{s}^{\mathrm{des}})$ is an n-dimensional error signal derived from a suitable comparison between the estimated object appearance and the desired one, $k \in [0, 1]$ is the feedback gain, and finally $\mathcal{L}^{+}$ denotes the $6 \times n$ pseudo-inverse of $\mathcal{L}$. Control law (2) ensures a zero steady-state error for a constant object motion. Position feedback, if k is tuned properly, compensates for various modeling inaccuracies (manipulator kinematics, camera-object interaction model, camera parameters, finite differences approximation, etc.). The anticipation term $\widehat{\boldsymbol{V}}_o$ can be obtained from eq. (1) as:

$$\widehat{\boldsymbol{V}}_o = \widehat{\boldsymbol{V}}_c - \mathcal{L}^{+}\widehat{\boldsymbol{d}}, \tag{3}$$

where the camera motion estimate $\widehat{\boldsymbol{V}}_c$ is simply obtained from actuator encoders data.

Being our control scheme of differential nature, each task has to be provided with initial conditions, whose accuracy is not critical, due to the presence of feedback in the control. In the case of simple reactive tasks, initial conditions are the result of a pre-segmentation of the image region enclosing the object, that is of an attentive-like processing of raw visual data—see also [13, 10]. For complex active tasks, further information may be required for a full task specification. A raw initial guess of the camera extrinsic—3D relative position between camera and object—and intrinsic parameters must be also provided, in order to compute and update the interaction matrix which will have, of course, entries which depend on camera parameters and are time-dependent. Yet, *the camera needs not to be calibrated.* Indeed, as the control loop closes at the image level rather than in space, it can be shown that calibration parameters only affect the speed of convergence, not its stability (see also [19]), and control proves to converge even when a bad estimate of camera parameters is provided.

Notice also that when several tasks are executed independently in parallel, there is a danger of tasks issuing conflicting commands to hardware and computing resources. Such conflicts can be resolved by organizing the tasks into a *hierarchy* based on the processing time (or bandwidth) of the transformations and, in ultimate analysis, on the feedback gain of each task. With such techniques, slower tasks, working in more abstract reference spaces, provide the reference signal to lower level tasks. (A similar mechanism takes place, at a different control bandwidth scale, in the layering of proprioceptive and exteroceptive tasks [17]. The former, usually simple PIDs, are considered as virtually instantaneous with respect to the latter in terms of loop time.)

3 Models and Measurements

In this section we describe in detail both the modeling and implementation aspects of our approach, and how they enter in the control scheme. We first introduce two interaction matrices obtained from a linearized model of camera-object interaction. Such a model, which is obtained from some basic assumptions about the geometry of camera projection and of the visible surface of the object, is used also to define an object representation based on image contours, to initialize and update the interaction matrices, and to develop the passive tracking equations.

3.1 Interaction model

Let us refer to a pinhole camera with fixed focal length f and optical axis Z. (The interaction model being referred to camera centered coordinates, all geometrical entities but focal length depend on time. Therefore, unless this leads to an ambiguous notation, we avoid mentioning explicitly the variable t.) Let $Z(X,Y)$ the object's visible surface in camera-centered coordinates, and $[x\ y]^{\mathrm{T}}$ be the perspective projection of point $\boldsymbol{P} = [X\ Y\ Z(X,Y)]^{\mathrm{T}}$ onto the image plane:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z(X,Y)} \begin{bmatrix} X \\ Y \end{bmatrix}. \tag{4}$$

Suppose now the camera to have a *narrow field of view in the sense of vanishing squares*, i.e. that the transversal dimensions of the sensor are small enough with respect to focal length to assume that, for any two imaged object points $\boldsymbol{P}$ and $\boldsymbol{Q}$:

$$\frac{x^{\mathrm{P}}}{f}\frac{x^{\mathrm{Q}}}{f} \approx 0; \qquad \frac{x^{\mathrm{P}}}{f}\frac{y^{\mathrm{Q}}}{f} \approx 0; \qquad \frac{y^{\mathrm{P}}}{f}\frac{y^{\mathrm{Q}}}{f} \approx 0, \tag{5}$$

thus constraining also by eq. (4), in order for the object to be visible, its transverse dimensions to be small with respect to its depth. (In the case of a wide field of view, the approximation above holds approximately true in the case of an object almost centered in the visual field, and sufficiently far from the camera plane.) Specifically, assume the depth function $Z(X,Y)$ to be sufficiently regular for its quadratic and higher order terms to be negligible. The visible surface itself can then be approximated by a planar surface, referred to as *plane of attention*, of equation

$$Z(X,Y) = pX + qY + c, \tag{6}$$

the plane coefficients determining, up to a degree of freedom, the relative pose and distance of camera and object. By combining eqs. (4) and (6) and defining $z(x,y)$ by $z(fX/Z, fY/Z) = Z$, we obtain:

$$z(x,y) = \frac{c}{1 - p\frac{x}{f} - q\frac{y}{f}}, \tag{7}$$

which expresses object depth directly in terms of p, q and c and image coordinates.

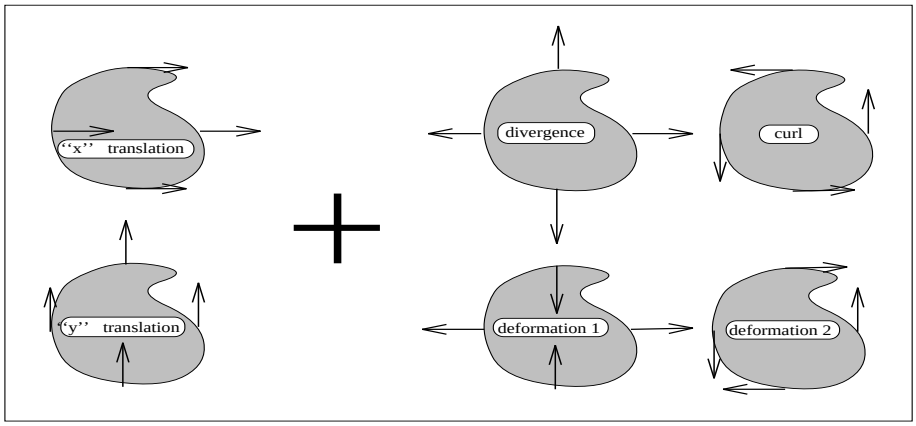

Figure 2: The six degrees of freedom of first-order image shape..

The *dynamic interaction* between camera and object can be expressed, at a generic image point $\boldsymbol{x}$, in terms of the 2D *motion field* $\boldsymbol{v}(x,y) = \dot{\boldsymbol{x}}$, arising in the image plane due to both surface's shape and 3D relative speed, as:

$$\boldsymbol{v}(x,y) = \mathcal{V}(x,y)\,\Delta\boldsymbol{V}. \tag{8}$$

To obtain the 2×6 *motion field matrix* $\mathcal{V}(x,y)$, we express the relative speed of an object point with respect to the camera frame as

$$[\dot{X}\ \dot{Y}\ \dot{Z}]^{\mathrm{T}} = -\Delta\boldsymbol{T} - \Delta\boldsymbol{\Omega} \wedge [X\ Y\ Z]^{\mathrm{T}}, \tag{9}$$

and we differentiate eq. (4) with respect to time, taking into account eqs. (9) and (7) and canceling out quadratic terms using eq. (5). We get

$$\mathcal{V}(x,y) = \begin{bmatrix} -f/z(x,y) & 0 & x/c & 0 & -f & y \\ 0 & -f/z(x,y) & y/c & f & 0 & -x \end{bmatrix}, \tag{10}$$

which is a linear function of image coordinates. (Notice that in the general case of full-perspective camera projection, the motion field matrix associated to a planar surface in motion (the plane of attention) would have been a quadratic function of image coordinates.) We rearrange now the terms in eq. (8) in order to emphasize this first-order spatial structure of motion field, or *motion parallax*, around an image point—say $\boldsymbol{x}^{\mathrm{B}} = [x^{\mathrm{B}}\ \ y^{\mathrm{B}}]^{\mathrm{T}}$, the centroid of the image patch enclosing the whole object:

$$\boldsymbol{v}(x,y) \approx \boldsymbol{v}^{\mathrm{B}} + \mathcal{M}_{\mathrm{B}}\,[x - x^{\mathrm{B}} \quad y - y^{\mathrm{B}}]^{\mathrm{T}}, \tag{11}$$

with $\boldsymbol{v}^{\mathrm{B}} = \boldsymbol{v}(x^{\mathrm{B}}, y^{\mathrm{B}})$. Note that the motion parallax, which is encoded in the 2×2 matrix $\mathcal{M}_{\mathrm{B}}$, is independent of the evaluation point $\boldsymbol{x}^{\mathrm{B}}$. Indeed, the instantaneous shape transformation

in the image due to a camera movement is constant in the neighborhood of each object point, depending only on time. Thus, according to our linearized interaction model, *the dynamic evolution of any image patch enclosing the object has six degrees of freedom*, namely the two components of $\boldsymbol{v}^{\mathrm{B}}$, which account for the rigid translation of the whole patch (two degrees of freedom), and the four entries of $\mathcal{M}_{\mathrm{B}}$, which account for the changes in the shape of the patch itself. As shown in Fig. 2, patch shape transformations are conveniently expressed in terms of the four *differential invariants* of the motion field divergence (div), curl ($curl$), and two components of deformation (def_1, def_2). The divergence accounts for changes in area, curl for rigid rotations and deformation components for expansions/compressions along mutually perpendicular axes, without changes in area [20]. Such invariants are in a one-one correspondence with the entries of $\mathcal{M}_{\mathrm{B}}$, since:

$$\mathcal{M}_{\mathrm{B}} = \frac{1}{2} \begin{bmatrix} div + def_1 & def_2 - curl \\ def_2 + curl & div - def_1 \end{bmatrix}. \tag{12}$$

As in the motion field case, we relate the motion parallax $\boldsymbol{w} = [div\ curl\ def_1\ def_2]^{\mathrm{T}}$ to the camera relative speed through a 4×6 *motion parallax matrix* $\mathcal{W}$:

$$\boldsymbol{w} = \mathcal{W}\, \Delta \boldsymbol{V} \tag{13}$$

where, as a consequence of eqs. (11) and (12):

$$\mathcal{W} = \begin{bmatrix} p/c & q/c & 2/c & 0 & 0 & 0 \\ -q/c & p/c & 0 & 0 & 0 & -2 \\ p/c & -q/c & 0 & 0 & 0 & 0 \\ q/c & p/c & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{14}$$

The motion field $\mathcal{V}^{\mathrm{B}} = \mathcal{V}(x^{\mathrm{B}}, y^{\mathrm{B}})$ and motion parallax $\mathcal{W}$ matrices, and the square 6×6 matrix

$$\mathcal{U} = \begin{bmatrix} \mathcal{V}^{\mathrm{B}} \\ \mathcal{W} \end{bmatrix} \tag{15}$$

such that

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{v}^{\mathrm{B}} \\ \boldsymbol{w} \end{bmatrix} = \mathcal{U}\, \Delta \boldsymbol{V} \tag{16}$$

can be effectively used as interaction matrices for the design of visual tasks, as we show in Sect. 4. When $\mathcal{U}$ is used, *a one-one correspondence is established between the six degrees of freedom which describe appearance evolution and those of camera motions*. Notice that the linearization of the model allows a compact representation of dynamic interaction, this being evident from the small dimensions of $\mathcal{V}^{\mathrm{B}}$ and $\mathcal{W}$. The same could not have been achieved with a full-perspective model. The interaction matrices depend on a number of camera parameters, namely f (intrinsic), and p, q and c (extrinsic). The initial and run-time value of these parameters must be known, even if approximately, to improve the convergence of the control scheme.

3.2 Initialization

A raw estimate of initial object pose and distance is obtained by assuming, for *static interaction*, a *weak perspective*, or *scaled orthography*, camera approximation [21]:

$$\begin{bmatrix} x \\ y \end{bmatrix} \approx \lambda \begin{bmatrix} X \\ Y \end{bmatrix}, \tag{17}$$

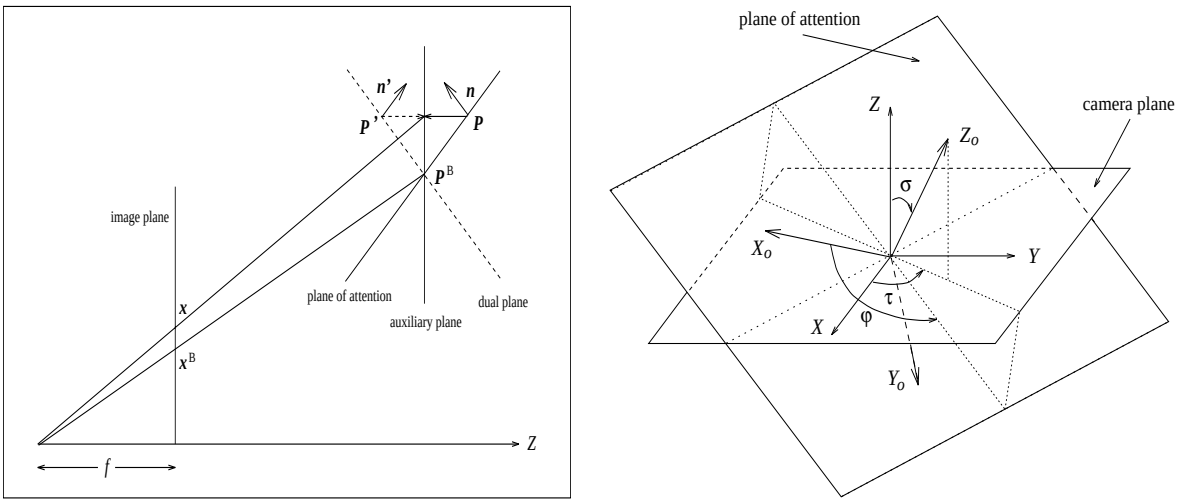

Figure 3: *Left*: weak perspective projection. *Right*: Definition of extrinsic parameters. The camera-centered frame has been translated for convenience in the object-centered frame's origin.

where $\lambda = f/z(x^{\mathrm{B}}, y^{\mathrm{B}})$ is an isotropic image scaling factor which is inversely proportional to object depth (Fig. 3, *left*). Such a model is easily obtained from eq. (7) and eq. (4), by taking into account the narrow field of view constraint of eq. (5) and the additional constraint $p\,x^{\mathrm{B}}/f + q\,y^{\mathrm{B}}/f \approx 0$. We express the model in terms of object-centered coordinates—that is, by explicitly taking into account the extrinsic camera parameters. To this end, we fix, as in Fig. 3 (*right*), an object-centered frame $\{X_o, Y_o, Z_o\}$ on the visible surface's centroid $[X^{\mathrm{B}}\ Y^{\mathrm{B}}\ Z^{\mathrm{B}}]^{\mathrm{T}}$, so that in terms of object-centered coordinates the object plane has simply equation $Z_o = 0$. This frame is uniquely determined by the three angles σ, τ and φ, the latter angle providing at each time the direction of the X_o axis with respect to the current direction, in the object plane, of maximum depth decrease:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos\tau & -\sin\tau & 0 \\ \sin\tau & \cos\tau & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\sigma & 0 & \sin\sigma \\ 0 & 1 & 0 \\ -\sin\sigma & 0 & \cos\sigma \end{bmatrix} \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_o \\ Y_o \\ 0 \end{bmatrix}. \qquad (18)$$

Notice that the parameters p and q give the object plane orientation (that is, the orientation of the Z_o axis) with respect to the camera, leaving undetermined, up to a rotation φ, the object-centered frame orientation. Explicitly, it holds:

$$p = -\tan\sigma\cos\tau; \qquad q = -\tan\sigma\sin\tau, \qquad (19)$$

where $\sigma \in [0, \pi/2]$ is the *slant* angle between the plane normal $[-p\ \ -q\ \ 1]^{\mathrm{T}}$ and the Z-axis, and the *tilt* angle $\tau \in [-\pi, \pi]$ gives the direction in the image plane of maximum depth decrease with respect to the X-axis (Fig. 3, *right*). From eqs. (17) and (18) we have:

$$[X - X^{\mathrm{B}}\ Y - Y^{\mathrm{B}}]^{\mathrm{T}} = \mathcal{T}^{\mathrm{wp}}\,[X_o\ Y_o]^{\mathrm{T}} \qquad (20)$$

with

$$\mathcal{T}^{\mathrm{wp}} = \mathcal{T}^{\mathrm{wp}}(\lambda, \tau, \sigma, \varphi) = \lambda \begin{bmatrix} \cos\tau & -\sin\tau \\ \sin\tau & \cos\tau \end{bmatrix} \begin{bmatrix} \cos\sigma & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix} \qquad (21)$$

where, besides the two planar rotations, the central matrix introduces an anisotropic scaling in the imaged pattern.

An approximation of the weak perspective matrix:

$$\widehat{\mathcal{T}}^{\text{wp}} = \frac{1}{2}\begin{bmatrix} \alpha+\gamma & \delta-\beta \\ \delta+\beta & \alpha-\gamma \end{bmatrix}, \tag{22}$$

can be easily obtained from the least squares comparison of the current appearance of the object and a frontoparallel view of the object. Once that this is known, we can estimate both pose and distance. We start computing the slant:

$$\cos\sigma = \rho - \sqrt{\rho^2 - 1} \in [0,1] \tag{23}$$

where $\rho = \frac{1+r}{1-r} \geq 1$, $r = \frac{\gamma^2+\delta^2}{\alpha^2+\beta^2} \in [0,1]$. Then we evaluate the scaling factor as:

$$\frac{f - p\,x^{\text{B}} - q\,y^{\text{B}}}{c} = \lambda = \sqrt{\frac{\det(\widehat{\mathcal{T}}^{\text{wp}})}{\cos\sigma}}, \tag{24}$$

from which c can be computed using eq. (19) after estimating τ. The linear system

$$\begin{cases} \tau - \varphi = \arctan(\beta/\alpha) \\ \tau + \varphi = \arctan(\delta/\gamma) + \pi \end{cases} \tag{25}$$

provides us actually with two dual solutions for τ, which differ by π. This results in the well-known *pose ambiguity* which is typical of any perspective linearization: there are two distinct object poses sharing the same visual appearance (see again Fig. 3, *left*). In the case of weak perspective, the ambiguity can be written as $\mathcal{T}^{\text{wp}}(\lambda,\tau,\sigma,\varphi) = \mathcal{T}^{\text{wp}}(\lambda,\tau+\pi,\sigma,\varphi+\pi)$. To disambiguate the pose, one can refer back to the full-perspective model, and choose as the "true" pose the one providing the best lest squares fit with image data [22].

3.3 Updating the interaction matrix

When $\mathcal{W}$ is used for control, it is convenient to re-estimate it at each control step to speed up convergence. For a robust estimate, we combine the current estimated values of the pose and distance coefficients with their predicted values, obtained using finite differences and the temporal dependence of pose and distance on relative speed. (Relative speed is also estimated at run time from eq. (1).) From eq. (13) we have:

$$\begin{cases} p = \dfrac{\Delta T_{\text{X}} def_1 + \Delta T_{\text{Y}} def_2}{\Delta T_{\text{X}}{}^2 + \Delta T_{\text{Y}}{}^2} \cdot c \\ q = \dfrac{-\Delta T_{\text{Y}} def_1 + \Delta T_{\text{X}} def_2}{\Delta T_{\text{X}}{}^2 + \Delta T_{\text{Y}}{}^2} \cdot c \end{cases} \tag{26}$$

where

$$c = \frac{2\Delta T_{\text{Z}}(\Delta T_{\text{X}}{}^2 + \Delta T_{\text{Y}}{}^2)}{(\Delta T_{\text{X}}{}^2 + \Delta T_{\text{Y}}{}^2) div - (\Delta T_{\text{X}}{}^2 - \Delta T_{\text{Y}}{}^2) def_1 - 2\Delta T_{\text{X}}\Delta T_{\text{Y}} def_2} \tag{27}$$

We have also, with computations akin to those carried out for the dynamic interaction model:

$$\begin{cases} \dot{p} = & & & -pq\Delta\Omega_{\text{X}} & +(1+p^2)\Delta\Omega_{\text{Y}} & +q\Delta\Omega_{\text{Z}} \\ \dot{q} = & & & -(1+q^2)\Delta\Omega_{\text{X}} & +pq\Delta\Omega_{\text{Y}} & -p\Delta\Omega_{\text{Z}} \\ \dot{c} = & p\Delta T_{\text{X}} & +q\Delta T_{\text{Y}} \; -\Delta T_{\text{Z}} & -cq\Delta\Omega_{\text{X}} & -cp\Delta\Omega_{\text{Y}}. & \end{cases} \tag{28}$$

3.4 Passive Tracking

The goal of passive tracking is to estimate at each time step the current visual representation of the object, both in terms of visual appearance and speed. In the current implementation, we measure the position and speed of objects using *active contours* [23]. Active contours enable the system to deal with quite generic object shapes, and allow the design of complex tasks, such as visual navigation by means of natural landmarks, or mimicking human gestures. Notice however that the choice of active contours is due primarily to its simplicity of implementation and relative robustness. Other object representations (e.g. image regions) and tracking techniques (e.g. correlation) could have equally served for our purpose.

Contours are represented by quadratic B-splines and used to estimate affine transformations which account for deformations of visual appearance:

$$\boldsymbol{x}(s) = \sum_{i=1}^{M} f_i(s)\boldsymbol{x}^i, \tag{29}$$

where the $f_i(s)$, $s \in [0,1]$ are the spline basis functions and the $\boldsymbol{x}^i$ are the spline control points.

The measurement of "contour speed" $\boldsymbol{u}$, that is of the six degrees of freedom of the affine transformation "in the small" between two successive contour instances $\{\boldsymbol{x}_t^i\}$ and $\{\boldsymbol{x}_{t+1}^i\}$, is done in two steps, by means of the spline control points. First, the contour centroids are evaluated $\left(\widehat{\boldsymbol{x}}^{\mathrm{B}} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{x}^i\right)$, then the 2×2 transformation about the origin is estimated via least squares:

$$\boldsymbol{x}_{t+1}^i - \widehat{\boldsymbol{x}}_{t+1}^{\mathrm{B}} = \delta\mathcal{A}_{\mathrm{B}} \left(\boldsymbol{x}_t^i - \widehat{\boldsymbol{x}}_t^{\mathrm{B}}\right) \tag{30}$$

and, using finite differences, we obtain:

$$\widehat{\boldsymbol{v}}^{\mathrm{B}} = \widehat{\boldsymbol{x}}_{t+1}^{\mathrm{B}} - \widehat{\boldsymbol{x}}_t^{\mathrm{B}}; \qquad \widehat{\mathcal{M}}_{\mathrm{B}} = \widehat{\delta\mathcal{A}}_{\mathrm{B}} - \mathcal{I}, \tag{31}$$

and from the latter we compute $\boldsymbol{w}$ with eq. (12).

Notice also that to enhance the quality of all visual measurements (visual representation, object motion), simple IIR filters are adopted, with gain $k^{\mathrm{IIR}} \in [0,1]$.

3.5 Feedback

The feedback error is evaluated at each control step by a comparison of the desired visual appearance, $\{\boldsymbol{x}^{i,\mathrm{des}}\}$ and the estimated one, $\{\widehat{\boldsymbol{x}}^i\}$. Thus the centroid feedback error is $\boldsymbol{v}_e^{\mathrm{B}} = \boldsymbol{x}^{\mathrm{B,des}} - \widehat{\boldsymbol{x}}^{\mathrm{B}}$, while in the case of contour shape evolution, a 4-dimensional error vector $\boldsymbol{w}_e$ is computed using the spline contour control points. Such an error is computed from eq. (12) by evaluating the *error matrix* $\mathcal{M}_e = \delta\mathcal{A}_e - \mathcal{I}$, with $\delta\mathcal{A}_e$ such that:

$$\boldsymbol{x}^{i,\mathrm{des}} - \boldsymbol{x}^{\mathrm{B,des}} = \delta\mathcal{A}_e(\widehat{\boldsymbol{x}}^i - \widehat{\boldsymbol{x}}^{\mathrm{B}}). \tag{32}$$

3.6 Planning

Planning is required to produce a shift of viewpoint. In our approach, such a shift is associated with an according *smooth, progressive change of object's visual appearance*, from an

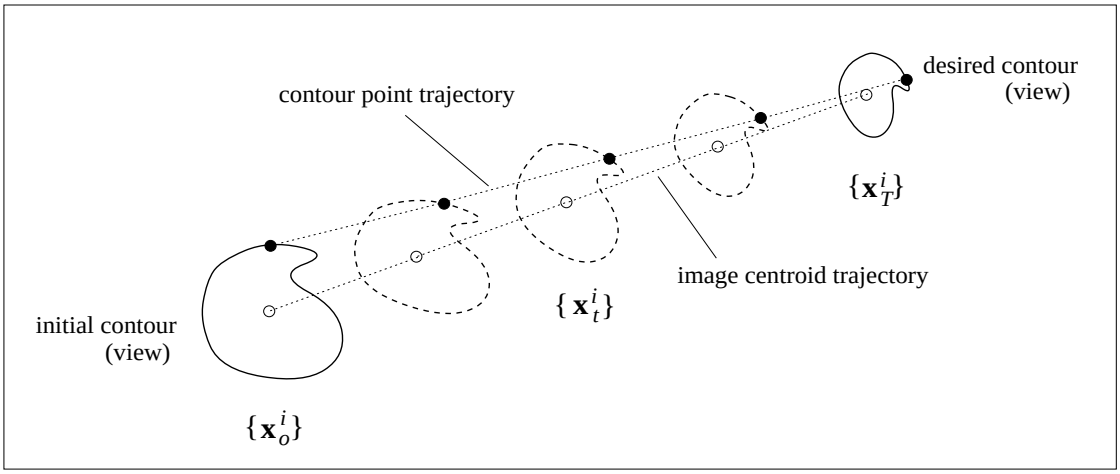

Figure 4: Planning a sequence of affine mappings which smoothly transforms one contour into the other. Each contour point follows a linear trajectory in the image, with a specific speed profile.

initial contour $\{\boldsymbol{x}_o^i\}$ to a final desired contour $\{\boldsymbol{x}_{\mathrm{T}}^i\}$ (see Fig. 4). The mapping "in the large" between the contours is evidently affine:

$$\boldsymbol{x}_{\mathrm{T}}^i = \boldsymbol{x}_{\mathrm{T}}^{\mathrm{B}} + \mathcal{A}_{\mathrm{T}}(\boldsymbol{x}_o^i - \boldsymbol{x}_o^{\mathrm{B}}), \tag{33}$$

as the result of a sequence of affine transformations "in the small"—a sequence of affine transformations producing always an affine transformation. Indeed, the general fact that *the mapping between any two object views* $\{\boldsymbol{x}_1\}$ *and* $\{\boldsymbol{x}_2\}$ *is affine* is a direct consequence of the static interaction model, and it holds:

$$[x_2 - x_2^{\mathrm{B}}\ y_2 - y_2^{\mathrm{B}}]^{\mathrm{T}} = \mathcal{A}_{12}[x_1 - x_1^{\mathrm{B}}\ y_1 - y_1^{\mathrm{B}}]^{\mathrm{T}}, \tag{34}$$

with

$$\mathcal{A}_{12} = \mathcal{T}_2^{\mathrm{wp}}(\mathcal{T}_1^{\mathrm{wp}})^{-1}. \tag{35}$$

The reference contour evolution used for feedforward control is planned as follows according to a polynomial trajectory approximation. A 2-vector $\boldsymbol{a}(t)$ and a 2×2 matrix $\mathcal{A}(t)$ are defined for $t \in [0, T]$, so that the generic contour control point trajectory is:

$$\boldsymbol{x}_t^{\mathrm{i,des}} = [\boldsymbol{a}(t) + \boldsymbol{x}_o^{\mathrm{B}}] + \mathcal{A}(t)\,(\boldsymbol{x}_o^i - \boldsymbol{x}_o^{\mathrm{B}}). \tag{36}$$

It is easy to show that, adopting a finite difference approximation for derivatives, the desired appearance evolution evaluates as:

$$\boldsymbol{x}_{t+1}^{i,\mathrm{des}} = \boldsymbol{x}_t^{i,\mathrm{des}} + \boldsymbol{v}_t^{\mathrm{B,des}} + \mathcal{M}_t^{\mathrm{des}}(\boldsymbol{x}_t^{i,\mathrm{des}} - \boldsymbol{x}_t^{\mathrm{B,des}}), \tag{37}$$

with $\boldsymbol{v}_t^{\mathrm{B,des}} = \dot{\boldsymbol{a}}(t)$ and $\mathcal{M}_t^{\mathrm{des}} = \dot{\mathcal{A}}(t)\mathcal{A}^{-1}(t)$, this last to be transformed into a desired evolution vector $\boldsymbol{w}^{\mathrm{des}}$ using eq. (12). The following boundary conditions on the trajectory ensure that the contour evolution starts with the initial contour and terminates with the desired contour:

$$\boldsymbol{a}(0) = \mathbf{o}, \quad \boldsymbol{a}(T) = \boldsymbol{x}_{\mathrm{T}}^i - \boldsymbol{x}_o^i; \qquad \mathcal{A}(0) = \mathcal{I}, \quad \mathcal{A}(T) = \mathcal{A}_{\mathrm{T}}. \tag{38}$$

Additional constraints on the functions $\boldsymbol{a}$ and $\mathcal{A}$—with beneficial effects on contour tracking, visual analysis and camera velocities and accelerations—can be imposed by means of cubic or higher order polynomial approximations of the trajectory, in terms of a desired dynamical behavior of contour points:

$$\boldsymbol{v}_t^{i,\mathrm{des}} = \dot{\boldsymbol{a}}(t) + \dot{\mathcal{A}}(t)(\boldsymbol{x}_o^i - \boldsymbol{x}_o^{\mathrm{B}}). \tag{39}$$

For example, a cubic polynomial trajectory

$$\boldsymbol{a}(t) = \sum_{j=0}^{3} \boldsymbol{c}_j t^j; \qquad \mathcal{A}(t) = \sum_{j=0}^{3} \mathcal{C}_j t^j, \tag{40}$$

where $\boldsymbol{c}_j$ and $\mathcal{C}_j$ are constants, must be chosen so as to impose zero boundary conditions on contour speed:

$$\boldsymbol{v}_o^{i,\mathrm{des}} = \boldsymbol{v}_{\mathrm{T}}^{i,\mathrm{des}} = \mathbf{o} \qquad \Longrightarrow \qquad \dot{\boldsymbol{a}}(0) = \dot{\boldsymbol{a}}(T) = \mathbf{o}; \qquad \dot{\mathcal{A}}(0) = \dot{\mathcal{A}}(T) = \mathrm{O}, \tag{41}$$

and get:

$$\boldsymbol{a}(t) = \xi(t)(\boldsymbol{x}_{\mathrm{T}}^{\mathrm{B}} - \boldsymbol{x}_o^{\mathrm{B}}); \qquad \mathcal{A}(t) = \xi(t)\mathcal{A}_{\mathrm{T}} + [1 - \xi(t)]\mathcal{I}, \tag{42}$$

with $\xi(t) = \chi^2(t)[3 - 2\chi(t)] \in [0, 1]$, being $\chi(t) = (t/T) \in [0, 1]$. A quintic polynomial approximation, with the further constraints of zero boundary accelerations would yield instead $\xi(t) = \chi^3(t)[6\chi^2(t) - 15\chi(t) + 10] \in [0, 1]$, thus achieving a smoother trajectory at the expense of a slower convergence.

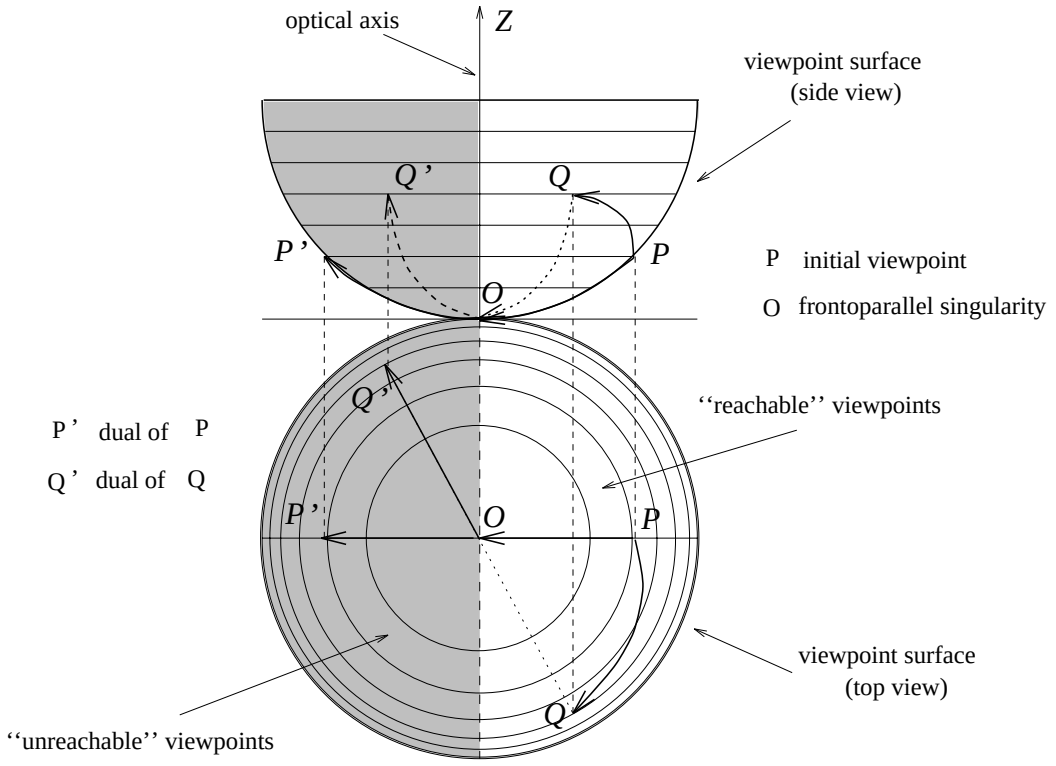

Figure 5: Viewpoint surface, pose ambiguity and frontoparallel singularity for weak perspective.

Embedding the planning strategy in the control scheme provides us with an "engine" which produces a 3D viewpoint change based on a desired change of visual appearance.

VISUAL TASKS	*Fixation Pursuit*	*Active Tracking*	*Active Positioning*
Task type	reactive	reactive	active
Initial conditions	$\{\boldsymbol{x}_o^i\}$	$\{\boldsymbol{x}_o^i\}$	$\{\boldsymbol{x}_o^i\}$
Visual representation	$(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{v}^{\mathrm{B}})$	$(\{\boldsymbol{x}^i\}, \boldsymbol{u})$	$(\{\boldsymbol{x}^i\}, \boldsymbol{u})$
Task description	$\boldsymbol{x}^{\mathrm{B},\,\mathrm{des}} = \mathbf{o}$	$\{\boldsymbol{x}^{i,\,des}\} = \{\boldsymbol{x}_o^i\} \quad \forall t$	$\{\boldsymbol{x}^{i,\,des}\} = \{\boldsymbol{x}_{\mathrm{T}}^i\}$ at $t = T$
Interaction matrix	$\mathcal{V}^{\mathrm{B}}$	$\mathcal{U}$	$\mathcal{U}$

Table 1: Synthetic description for the three tasks.

Let us introduce the *viewpoint surface* as the semi-sphere whose points (corresponding one-one to the innerwise unit normals) represent all possible relative orientations of the plane of attention with respect to the camera (Fig. 5). Any smooth change of orientation will correspond to a curvilinear path on this surface. Our planning engine selects automatically as goal viewpoint, between the two dual solutions which arise due to the pose ambiguity for any given and 2D goal appearance, the one which is closer to the initial 3D viewpoint moving along a geodesic path of the viewpoint surface. For instance, let $\boldsymbol{P}$ be the initial viewpoint, and $\boldsymbol{P'}$ be its dual, placed at its opposite side. It is evident that $\boldsymbol{P'}$ cannot be reached from $\boldsymbol{P}$ via a one-step planning strategy (the initial and goal 2D appearances do actually coincide) and that the relative orientation will not change. To reach $\boldsymbol{P'}$, we can complicate somewhat the planning strategy, by splitting the path in two parts, namely $\boldsymbol{PO}$ and $\boldsymbol{OP'}$, where $\boldsymbol{O}$ is the frontoparallel view of the object used also for raw pose estimation. In the general case, given $\boldsymbol{P}$ and a goal visual appearance corresponding to the two dual views $\boldsymbol{Q}$ and $\boldsymbol{Q'}$, we proceed as follows (refer again to Fig. 5):

1. determine the final viewpoint, and establish whether it is "reachable" ($\boldsymbol{Q}$) or "unreachable" ($\boldsymbol{Q'}$) via a single planning iteration;
2. in the first case, plan $\boldsymbol{PQ}$ and execute;
3. in the second case, split planning into $\boldsymbol{PO}$ and $\boldsymbol{OQ'}$.

Notice that the frontoparallel pose corresponds to the only control algorithmic singularity, in that the determinant of $\mathcal{U}$ vanishes for $p = q = 0$. In order to avoid desired camera speeds arbitrary large, we open the loop as $p^2 + q^2$ becomes too small. Then, as $\boldsymbol{O}$ has been crossed and $p^2 + q^2$ is large enough again, we close the loop and plan towards $\boldsymbol{Q'}$, which has now become "reachable."

4 Three Uncalibrated Visual Tasks

We show here how to implement three different visual tasks by means of the approach proposed above. The task characteristics are summarized in Tab. 1: notice that the first two are reactive, while the third one, which contains also a planning strategy, is active. The tasks are introduced in order of complexity, this being related to computational burden and speed, and degree of object representation required.

4.1 Fixation Pursuit

The simplest task is *fixation pursuit*, that is, the tracking of an object point. The importance of fixation in active vision has already been emphasized in several recent works (see e.g. [24, 25, 5]). Fixation is one of the most basic tasks also in the human visual system [26]; it allows one to concentrate the major part of visual resources in the processing of foveal data, and to keep the high-resolution fovea centered on the object of interest, while using the image periphery for attentive processing and monitoring. For this reason, fixation could be even more effective if anthropomorphic sensors were used in the place of traditional cameras [4].

The object centroid is chosen as fixation point, whose 3D tracking can be accomplished, as a result of our linearized interaction model, through a simple 2D tracking of the image centroid. Indeed, due to linear approximation, the 2D centroid of the imaged object corresponds to the projection of the 3D visible surface centroid. Note from eq. (8) that, under fixation, a constraint is determined among the translational and rotational relative speed components parallel to the image plane:

$$c \cdot [\Delta\Omega_{\mathrm{X}} \quad \Delta\Omega_{\mathrm{Y}}]^{\mathrm{T}} = [\Delta T_{\mathrm{Y}} \quad -\Delta T_{\mathrm{X}}]^{\mathrm{T}}. \tag{43}$$

This means that *to all tasks which "run slower" than fixation, the number of independent camera degrees of freedom is reduced from six to four*, thereby simplifying further the interaction model. Specifically, under the fixation constraint (43), the motion field matrix of eq. (10) becomes:

$$\mathcal{V}'(x,y) = \begin{bmatrix} -f\left(\frac{1}{z(x,y)} - \frac{1}{c}\right) & 0 & x/c & 0 & 0 & y \\ 0 & -f\left(\frac{1}{z(x,y)} - \frac{1}{c}\right) & y/c & 0 & 0 & -x \end{bmatrix}, \tag{44}$$

with of course $\mathcal{V}'(x^{\mathrm{B}}, y^{\mathrm{B}}) = \mathcal{V}'(0,0) = \mathrm{O}$. (Note that no change occurs to the parallax matrix instead. In fact, according to our linearized model of interaction, only the rotational component along the optical axis affects image shape changes—see eq. (13).) Similar simplifications occur also in the equations for the updating of the interaction matrices.

4.2 Active Tracking

With the term *active tracking* we refer to a visual task in which the object motion is tracked by means of active movements of the camera (instead than by means of pseudo-attentive movements as in the case of passive tracking). In other words, the goal of active tracking is *mimicking the motion of an object in the visual environment.* This kind of task can prove to be useful in the design of human-robot interfaces (mimicking human gestures), and provide also—by simply reading from encoders the 3D speed of the camera—an estimate of 3D object motion. Notice that also in this case there exists a fixation point—in fact, the centroid's speed in constrained to be zero—but that this in general is a point of the visual periphery, due to the fact that the centroid's initial position for this task is not necessarily zero. As a consequence, the direction of gaze does not coincide with the direction of attention. Such an attentive shift is indeed possible in humans, as previously recalled talking about passive tracking, only at condition to be voluntary.

4.3 Active Positioning

The *active positioning* task consists in purposively changing the relative spatial configuration (pose, distance) of the camera with respect to a fixed or moving object. As such, active positioning can be of extreme importance for the optimal execution of complex perceptive and explorative tasks.

To run the task, we compute via least squares the transformation $\mathcal{A}_{\mathrm{T}}$ between the initial and the goal weak-perspective views of the object (eq. (33)).

On the basis of the above interpretation of active tracking in terms of attentional shifts, we can relate the movements of object's visual appearance in the image to corresponding "attentional movements" from a region to another of the image periphery. Better still, we can regard the active tracking task itself as a particular case of active positioning, where the desired final contour coincides with the initial contour, thereby requiring simply to extract the object model directly from raw image data.

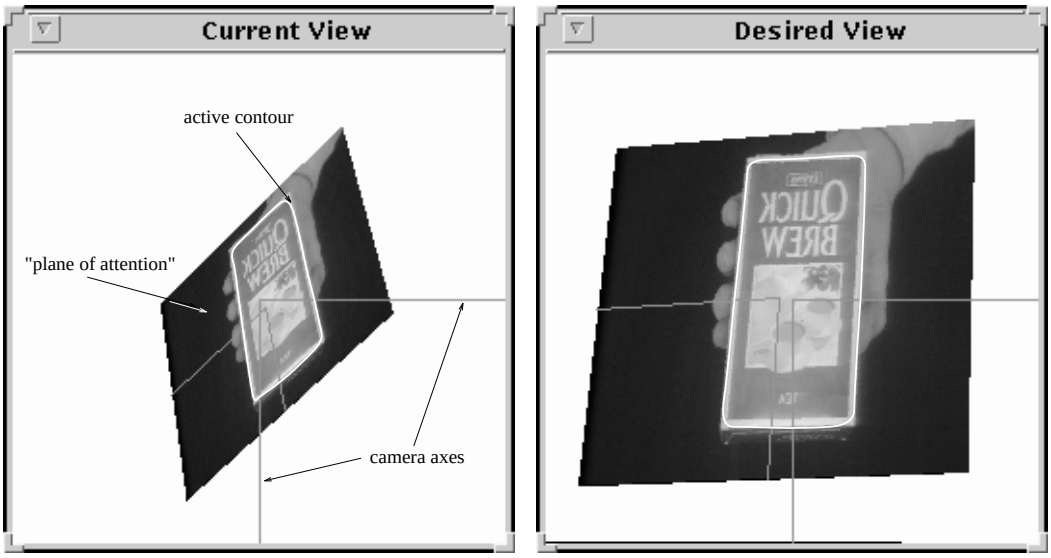

Figure 6: *Left*: the initial (before fixation) view of the experiment. *Right*: the desired configuration for active positioning.

4.4 An example

An simulation environment running under X11 was developed for testing visual tasks and their composition. In Fig. 6, the left window shows the current view of the object's plane of attention, together with the associated active contour, while the right window shows a desired object appearance as needed for active positioning. Simulations are carried out with a camera's focal length of $f = 18$ mm, and an object speed of $\boldsymbol{T}_o = [1.8\ 3.6\ 5.4]^{\mathrm{T}}$ mm/steps and $\boldsymbol{\Omega}_o = [0.01\ 0.02\ 0.03]^{\mathrm{T}}$ °/steps. The initial configuration is $c_o = 21600$ mm, $\sigma_o = 60°$, $\tau_o = 30°$, $\varphi_o = 150°$, and the goal configuration has parameters $c_{\mathrm{T}} = 16200$ mm, $\sigma_{\mathrm{T}} = 30°$,

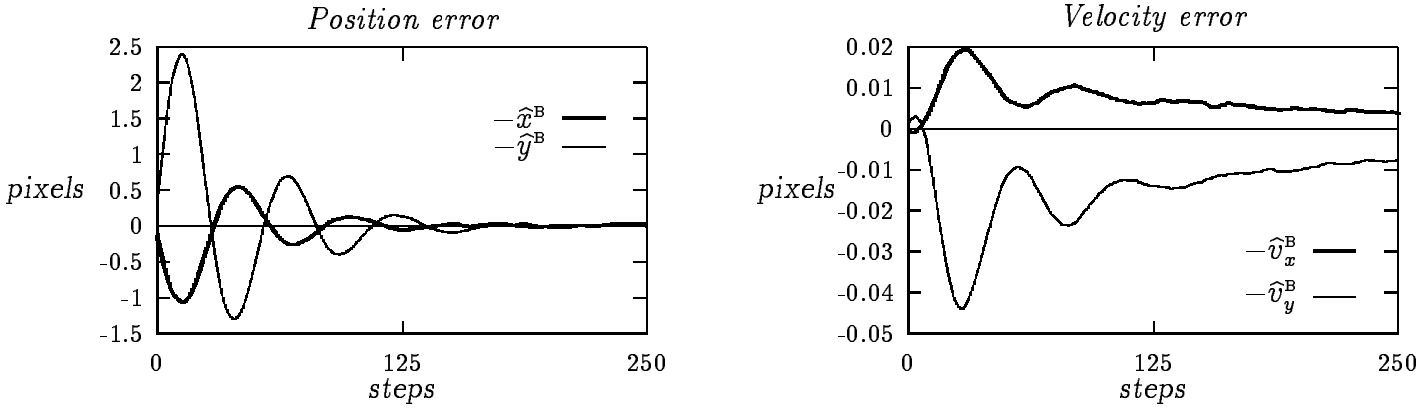

Figure 7: Fixation pursuit (steps $0 \div 250$). *Left*: image centroid position error. *Right*: image centroid velocity error.

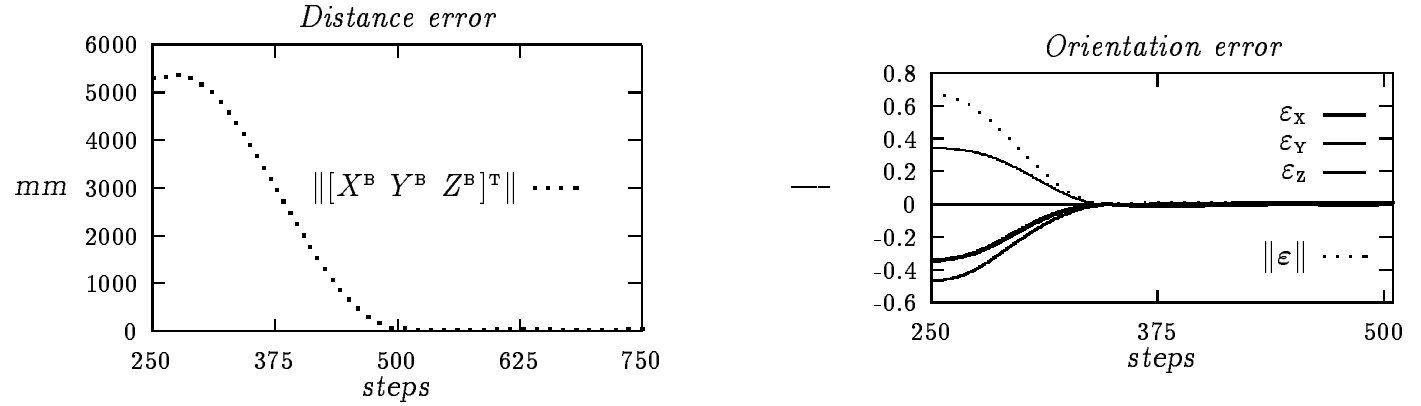

Figure 8: Active positioning (steps $250 \div 500$) and active tracking (steps $500 \div 750$). *Left*: distance error. *Right*: pose error.

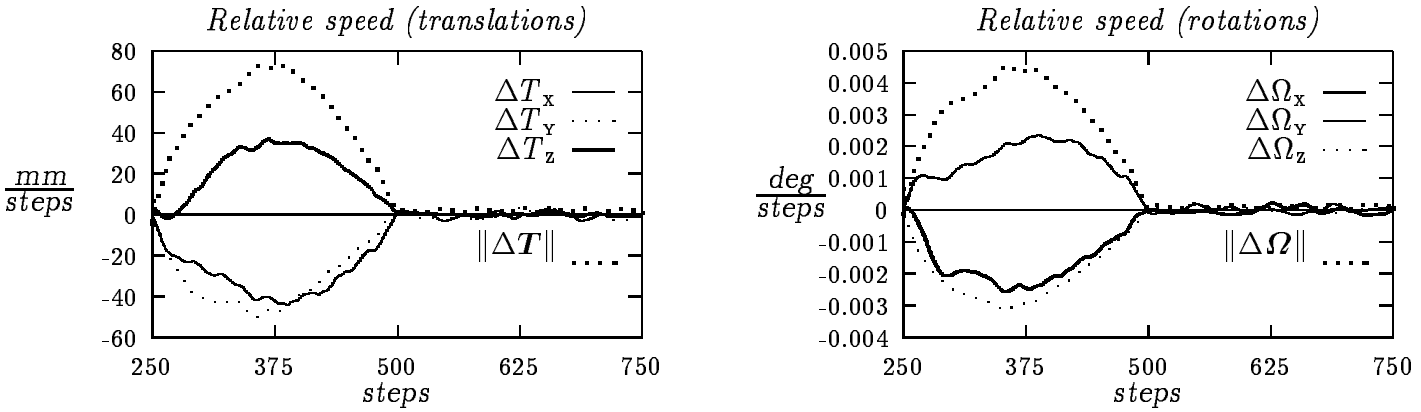

Figure 9: Relative speed of camera and object during active positioning (steps $250 \div 500$) and active tracking (steps $500 \div 750$). *Left*: translations. *Right*: rotations.

$\tau_{\mathrm{T}} = 60°$, $\varphi_{\mathrm{T}} = 150°$. Position measurements are smoothed using an IIR filter with gain $k_{\mathrm{p}}^{\mathrm{IIR}} = 0.05$, while the IIR velocity filter gain is $k_{\mathrm{v}}^{\mathrm{IIR}} = 0.005$.

Results for a sequence of 750 steps are shown in Figs. 7 through 9. In the first part of the sequence (steps $0 \div 250$), the *fixation pursuit* task is let to run alone, with a feedback gain set to $k^{\mathrm{fix}} = 0.2$, chosen so as to have a slightly underdamped control. Fig. 7 (*left*) shows that after less than 250 steps the initial position error of Fig. 6 (*left*) is completely recovered, due to the large associated feedback gain; the velocity error has a slower convergence instead (see Fig. 7, *right*).

After step 250, while fixation pursuit continues running, an *active positioning* task is started, which attempts to change the object view as in Fig. 6 (*right*) based on a cubic contour planning. This provides an example of task layering: as the active positioning feedback gain is less than the fixation gain by one order of magnitude, the system exhibits globally an *active positioning task with respect to a fixated object*. The positioning strategy consists thus actually in suitably commanding camera translation $\boldsymbol{T}$ and cyclotorsion Ω_{z} through the control of motion parallax, using the simplified motion field interaction matrix of eq. (44). Fig. 8 shows the 3D relative pose and distance error for the task, obtained from the comparison of current and desired camera frames $\{\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{a}\}$ and $\{\boldsymbol{n}^{\mathrm{des}}, \boldsymbol{s}^{\mathrm{des}}, \boldsymbol{a}^{\mathrm{des}}\}$. Specifically, the pose error is evaluated according to the formula [27] $\boldsymbol{\varepsilon} = \frac{1}{2}[\boldsymbol{n} \wedge \boldsymbol{n}^{\mathrm{des}} + \boldsymbol{s} \wedge \boldsymbol{s}^{\mathrm{des}} + \boldsymbol{a} \wedge \boldsymbol{a}^{\mathrm{des}}]$. As a typical performance, the goal configuration is computed with an error of within a few degrees (orientation angles) and millimeters (relative distance).

Fig. 9 shows the relative speed of camera and object. Notice that, due to the cubic-based planning and to a good tracking and filtering of object's independent motion, the relative speed profile varies gracefully, with gradual relative accelerations and decelerations. Notice also that, since the fixation task is still under execution, relative tranlations and rotations are related according to the fixation constraint of eq. (43).

When, after step 250, the planning goal is reached, the active task degenerates into an *active tracking* task, which ensures (steps $500 \div 750$, see again Figs. 8 and 9) that the new relative configuration of camera and object be maintained properly.

5 Robotic experiments

Experiments have been performed using an eye-in-hand robotic setup. The hardware setup consists of a PUMA 560 manipulator with MARK III controller equipped with a wrist-mounted camera, and a PC equipped with a frame grabber and a graphic accelerator. The PC features a 80486-66 MHz processor. The MARK III controller runs VAL II programs and communicates with the PC via the ALTER real-time protocol using an RS232 serial interface. The ALTER protocol allows us to modify the Cartesian setpoint of the robot arm every 28 ms. Due to the burden of calculation of tracking algorithms, a multirate real-time control has been implemented. New velocity setpoints are generated by the PC with a sampling rate $T_2 = N\ T_1$, where $T_1 = 28$ ms is the sampling rate of the ALTER protocol and N is an integer, depending on the number of control points of the B-spline contour. The computation time when using 10 control points is less than 100 ms. Hence, the sampling time T_2 has been chosen equal to $4\,T_1$. A "fast" communication process maintains the handshake with the MARK III controller sending the most recent velocity twist setpoints generated by the high level "slow" process, available in a mailbox.

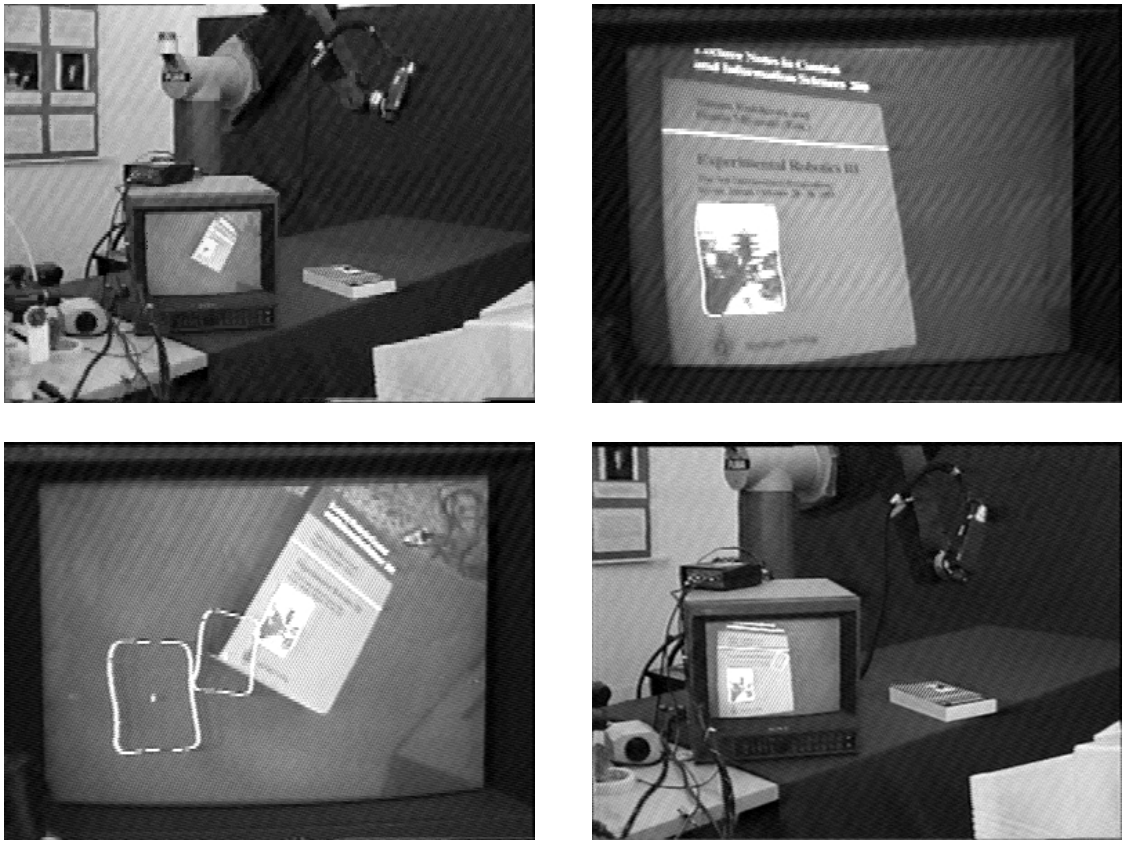

Figure 10: A positioning experiment. The monitor upon the table displays the current scene as seen by the camera. *Top left*: Initial configuration. *Top right*: Goal image appearance. *Bottom left*: Initial and goal contours, and an intermediate planned contour. *Bottom right*: The reached final configuration.

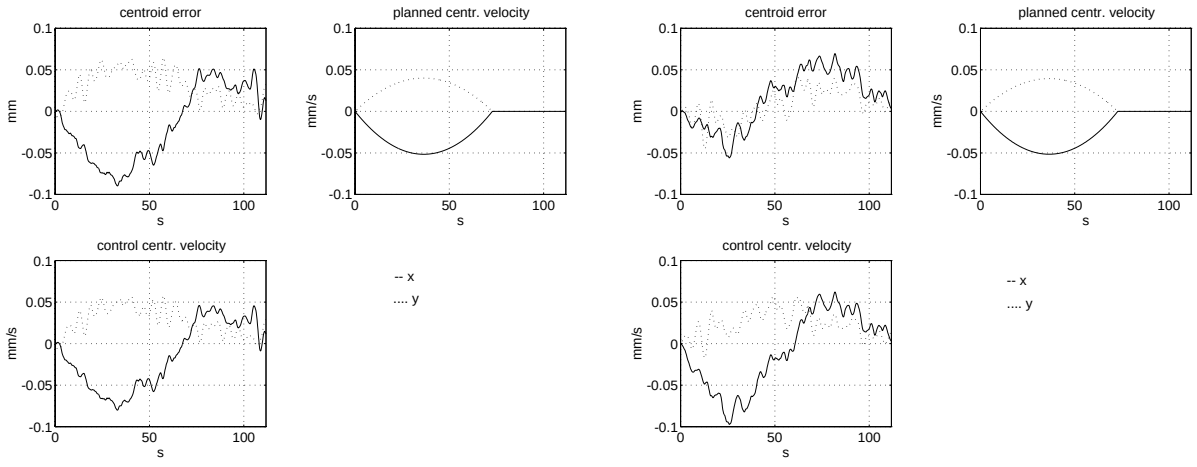

Figure 11: Comparison between the servoing (*left*) and planning (*right*) modes. Centroid.

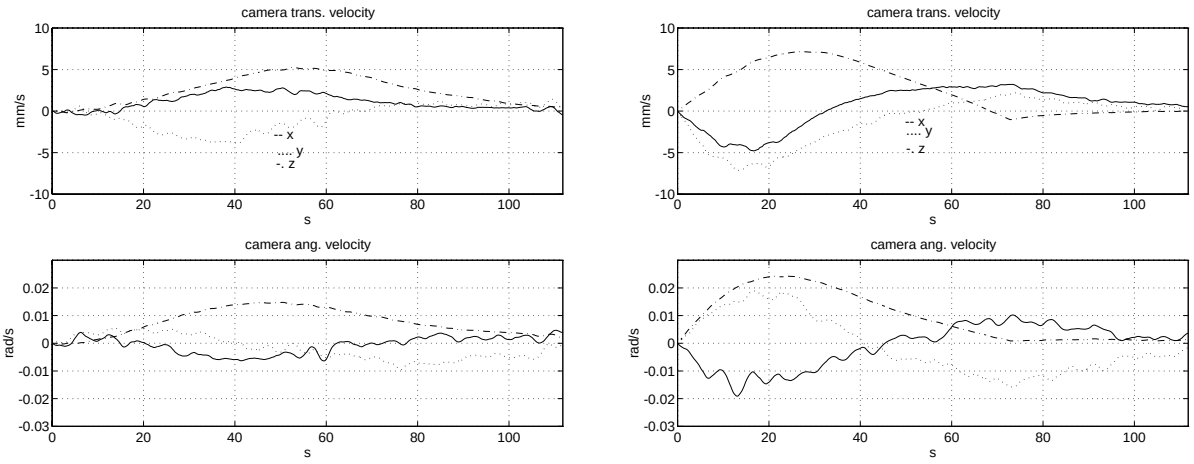

Figure 12: Comparison between the servoing (*left*) and planning (*right*) modes. Velocities.

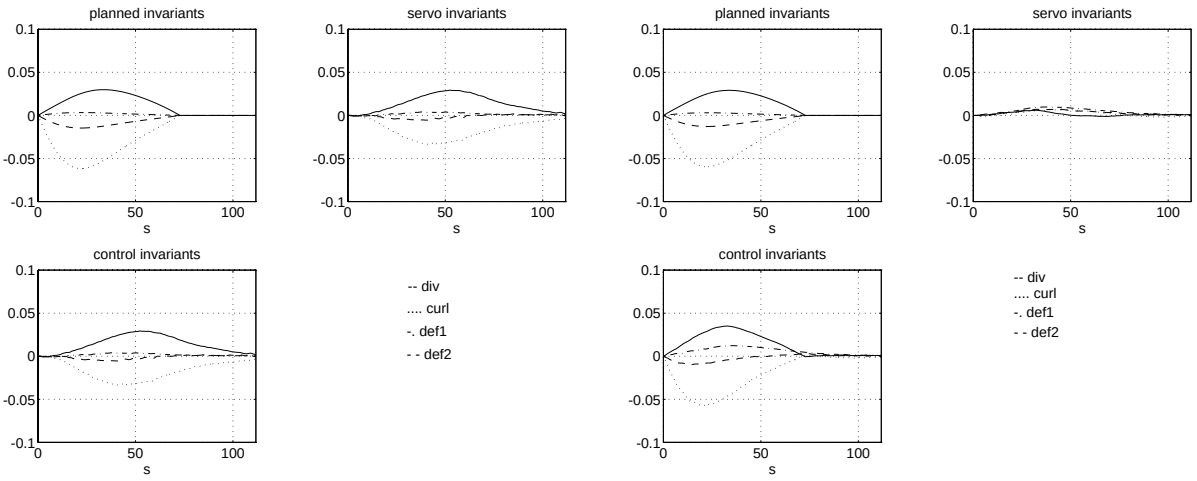

Figure 13: Comparison between the servoing (*left*) and planning (*right*) modes. Invariants.

We report on active positioning experiments with respect to a still planar object (a book upon a table, see Fig. 10). In order to assess the convergence and stability characteristics of the control scheme and tune up control parameters, the scheme for active positioning, which is the most complex of the three, is decomposed into its main "modes," which are tested separately. In increasing order of complexity, we have the modes:

1. *Output regulation mode.* This mode, which can be used to set properly the control gains in order to have stable behaviors, is relative to no desired contour evolution planning: the system has to move simply from an initial configuration to the desired one (output regulation task). In this case, the camera velocity commands derive from the 2D position error between the desired and the current contour configurations, thus yielding a system behavior closely similar to that described in [9]. The gain can be tuned up so as to obtain a slightly underdamped behavior of the closed loop system.

2. *Servoing mode.* With this mode, only the planned feature evolution (position) is provided for feedback error estimation, while the feedforward speed command is kept to zero. This is used to assess the tracking performance of the control scheme (servoing task). If the feedback gain is chosen according to the stability criterion above, it is likely that it is too small, so that residual errors may be present at the end of the planned trajectory. A number of pure regulation cycles may thus be taken into account, in order to achieve a complete convergence of the servoing task.

3. *Planning mode.* The normal mode, corresponding to the complete scheme. A considerable improvement with respect to the previous case (no lag) is obtained by reintroducing the feedforward term in the control.

Figs. 11 through 13 show the results of the comparison between the servoing and complete schemes, respectively. The same IIR digital filters have been used for smoothing sensory data. The gains of the filters as well as those of the feedback control term where tuned experimentally and were the same in the two experiments. Despite the fact that the interaction matrix $\mathcal{L}$, is only roughly computed (in these experiments, we did not even performed on-line estimation of 3D parameters), both the control schemes seem to be effective.

In Fig. 11, the centroid error is shown as well as the planned centroid velocity in the image plane and the velocity resulting from the control algorithm.

In Fig. 12, the translational and angular velocities of the camera are shown. As one can see, by using feedforward information in addition to feedback (planning mode), at the end of the planning phase (75 s), the Z-components of translational and rotational velocity are almost zero. This because inaccuracies in the estimate of 3D parameters p, q and c have a great influence only on the mapping between image centroid velocity and differential invariants on one side and T_{X}, T_{Y}, Ω_{X} and Ω_{Y} components of camera Cartesian velocity twist.

In Fig. 13, the planned motion parallax is shown as well as the invariants generated by the feedback (servo invariants) and those used to produce camera Cartesian velocity twist (control invariants). The effect of feedforward control is that of significantly reducing the job of the feedback.

References

1. R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):996–1005, 1988.
2. J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *International Journal of Computer Vision*, pages 333–356, 1988.
3. D.H. Ballard. Animate vision. *Artificial Intelligence*, 48:57–86, 1991.
4. M. J. Swain and M. A. Stricker. Promising directions in active vision. *International Journal of Computer Vision*, 11(2):109–126, 1993. Written by the attendees of the NSF Active Vision Workshop, University of Chicago, August 5–7, 1991.
5. J.L. Crowley, J.M. Bedrune, M. Bekker, and M. Schneider. Integration and control of reactive visual processes. In J.O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden, 1994*, pages II:47–58, 1994.
6. K. Pahlavan and J.O. Eklundh. A head-eye system—analysis and design. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 56(1):41–56, 1992.
7. S.J. Dickinson, H.I. Christensen, J. Tsotsos, and G. Olofsson. Active object recognition integrating attention and viewpoint control. In J.O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden, 1994*, pages II:3–14, 1994.
8. L.E. Weiss, A.C. Sanderson, and C.P. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, 3(5):404–417, 1987.
9. B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, 1992.
10. S.K. Nayar, H. Murase, and S.A. Nene. Learning, positioning and tracking visual appearance. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, California, 1994*, 1994.
11. E. Grosso. On perceptual advantages of eye-head active control. In J.O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden, 1994*, pages II:123–128, 1994.
12. M. Tistarelli and G. Sandini. Dynamic aspects in active vision. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 56(1):108–129, 1992.
13. I.D. Reid and D.W. Murray. Tracking foveated corner clusters using affine structure. In *Proceedings of the 4th IEEE International Conference on Computer Vision, Berlin, Germany, 1993*, pages 76–83, 1993.
14. D. Noton and L. Stark. Eye movements and visual perception. *Scientific American*, 224(6):34–43, 1971.
15. C. Colombo, M. Rucci, and P. Dario. Attentive behavior in an anthropomorphic robot vision system. *Robotics and Autonomous Systems*, 12(3–4):121–131, 1994.
16. J.L. Crowley and H.I. Christensen. *Vision as Process*. Springer Verlag Basic Research Series, 1994.
17. James S. Albus. *Brains, Behavior, and Robotics*. BYTE Books, 1981.
18. R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, pages 14–23, 1986.
19. B. Espiau. Effect of camera calibration errors on visual servoing in robotics. In *Preprints of the 3rd International Symposium on Experimental Robotics, Kyoto, Japan, 1993*, 1993.
20. R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. In *Proceedings of the 2nd European Conference on Computer Vision, S. Margherita Ligure, Italy, 1992*, pages 187–202, 1992.

21. J.L. Mundy and A. Zisserman. Projective geometry for machine vision. In J.L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*. MIT Press, 1992.
22. R. Horaud, S. Christy, and F. Dornaika. Object pose: The link between weak perspective, para perspective, and full perspective. Technical Report 2356, INRIA, Grenoble, 1994.
23. A. Blake, R. Curwen, and A. Zisserman. A framework for spatiotemporal control in the tracking of visual contours. *International Journal of Computer Vision*, 11(2):127–145, 1993.
24. G. Sandini and M. Tistarelli. Active tracking strategy for monocular depth inference over multiple frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):13–27, 1990.
25. E. Grosso and D.H. Ballard. Head-centered orientation strategies in animate vision. In *Proceedings of the 4th IEEE International Conference on Computer Vision, Berlin, Germany, 1993*, pages 395–402, 1993.
26. A.L. Yarbus. *Eye Movements and Vision*. Plenum Press, New York, NY, 1967.
27. J.Y.S. Luh, M.W. Walker, and R.P.C. Paul. Resolved acceleration control of mechanical manipulators. *IEEE Transactions on Automatic Control*, 25:468–474, 1980.