Binocular Tracking: Integrating Perception and Control

Alexandre Bernardino and José Santos-Victor

Abstract— **This paper presents an active binocular tracking system using log-polar images with contributions in both the perceptual and control aspects.**

The control part is based on the visual servoing framework, including kinematics and dynamics. We introduce a fixation constraint that simplifies the tracking problem by decoupling the visual kinematics and allowing to express system dynamics in image coordinates. Simple dynamic controllers are designed for each degree of freedom directly from image features.

In the perceptual part, we use a space variant sensor that emphasizes the center of the visual field (log-polar geometry). We present a new disparity estimation algorithm for log–polar images and provide a theoretical analysis to illustrate the advantages of using space variant images.

The overall system is implemented in the *Medusa* binocular head without any specific processing hardware. The use of log–polar images allows real-time performance (50 Hz). Tracking experiments are presented to illustrate system performance with different control strategies and objects of different shapes and motions.

Keywords— **Binocular Tracking, Log-Polar Image Sensors, Vision Based Control.**

I. Introduction

OBJECT tracking and fixation are important for many robotic tasks, such as navigation and ego–motion estimation [1], [2], velocity estimation [3] or object recognition [4]. This paper addresses the visual tracking problem and its two major sub–problems: perception and control.

Regarding the control aspects, the *Visual Servoing* approach [5], [6], [7], [8] has established a formal way of designing the motor control scheme in the presence of well defined visual features of the target. However, most control strategies consider only the system kinematics and neglect dynamics [9]. Additionally, the kinematic relations between target and robot motions are rather complicated and highly coupled.

With respect to the perception aspects, the main difficulties arise from the extraction of reliable image information to use in visual servoing algorithms. In structured conditions, with constrained object shapes, motions or simple backgrounds, some visual servoing systems have been successfully employed [10], [11], [6]. However, in more complex and dynamic environments most systems lack robustness, reactivity and flexibility. Usual feature based matching algorithms are slow and dependent on environmental conditions (illumination, clutter, size of objects, etc). In most systems, the amount of "noise" generated by the perceptual processes is such that the control scheme must be slow and conservative to filter out this noise.

The visual tracking problem is often addressed with monocular clues [12]. However, in the absence of models for target shape or motion, only 2D information can be extracted. Binocular information can help in overcoming this problem but often relies on time consuming matching/reconstruction procedures. One difficult problem that remains is that of segmenting the motion of the target from the background or other moving objects. Some authors use very simple objects (bright spots, circles) [3], background suppression or ego–motion compensation [13], which restricts the range of applicable situations since prior knowledge must be used.

This paper presents contributions in both control and perception aspects, and their integration in a fully working system — the *Medusa* binocular head. At the control level we show that target fixation decouples head kinematics and simplifies dynamics, allowing the expression of control algorithms in image space. The fixation constraint was used in [2] for perceptual purposes while we also use it to simplify control aspects.

At the perceptual level, we present algorithms for target motion estimation using log–polar images, with particular emphasis on a new disparity estimation strategy. Additionally we provide a theoretical analysis when using log–polar *versus* cartesian images for disparity estimation.

Perception and control are integrated on a binocular system capable of tracking objects of different shapes and motions in real-time. Targets can occupy as few as 10% of the visual field and may have generic shapes, temporal deformations and motions. Fixation is attained using simple and fast processing routines which include correlation, optical flow and spatial moments, thus not depending on computational hard algorithms such as stereo. Furthermore, since disparities are kept within a small range, fixation simplifies sensorial tasks that are fundamental to tracking: vergence and figure/ground segmentation [14].

In this paper, instead of using cartesian images, we exploit the geometric properties of the log-polar image representation [15], inspired by the non-uniform resolution of the human retina. We present methods to extract the target motion parameters from the log-polar images. All the image measurements are obtained using large image regions, which improves robustness to spurious data.

Both computational and algorithmic improvements arise from the use of log-polar images, profiting from the higher resolution at the image center [16], [17]. On the computational side, the image data reduction provided by the log-polar mapping allows for much smaller processing times than using cartesian images with the same field of view.

This work was partially funded by projects ESPRIT-LTR 30185, NARVAL and PRAXIS 2/2.1/TPAR/2074/95, SIVA.

A. Bernardino and J. Santos-Victor are with the *Instituto de Sistemas e Robótica*, IST - Torre Norte, 1049-001 Lisboa, Portugal. E-mail: {alex,jasv}@isr.ist.utl.pt

This favors real-time implementation with available off–the–shelf hardware. On the algorithmic side, the focus of attention at the image center improves the acquisition of perceptual information for the binocular tracking process, particularly in disparity estimation. We provide a theoretical analysis of the relation between cartesian and space-variant image processing and illustrate this fact with experimental results.

In Section II we review the kinematics perspective of visual servo systems, introduce the fixation constraint and show that it decouples the binocular head kinematics. In Section III, robot dynamics and the fixation constraint are used to develop independent image based controllers for each joint. Section IV describes the space variant sensor geometry and the disparity and motion estimation algorithms. In Section V we show extensive results of tracking experiments in a real binocular head and finally, in Section VI, we draw some conclusions and envision future improvements.

II. Kinematic Visual Servoing

In this section we describe the kinematics control aspects of our system and introduce our notation and basic assumptions. Firstly, a general framework is presented and then specified for our setup. This is required to better understand the theoretical results presented here, since a significant mathematical formalism is used. Emphasis is given to the eye-in-hand visual servoing problem [8], closely related to our particular application.

A. Visual Servoing Framework

The eye-in-hand visual servoing problem is usually stated as the minimization of a task function [5], depending on the object's pose $\underline{P}$, (position and attitude of points, lines, surfaces), and on the pose of the cameras, $\underline{Q}$. Often, pose information is represented in 6D cartesian coordinates (position and orientation), by means of a translation vector $\underline{T} \in \Re^3$ and an orientation matrix $\mathbf{R} \in SO^3$, both relative to a world coordinate frame $\mathcal{W}$.

For the sake of generality, we assume that each pose is parameterized by generalized coordinate vectors $\underline{p} \in \mathcal{C}_p \subseteq \Re^m$ and $\underline{q} \in \mathcal{C}_q \subseteq \Re^n$, where $\mathcal{C}_p$ and $\mathcal{C}_q$ are the target and camera configuration spaces. We denote target pose and camera pose by $\underline{P} = \mathcal{P}\left(\underline{p}\right)$ and $\underline{Q} = \mathcal{Q}\left(\underline{q}\right)$, respectively. For instance, $\underline{p}$ can contain spherical coordinates to represent position, and roll–pitch–yaw angles to represent orientation. Usually, the components of $\underline{q}$ directly represent the system's degrees of freedom (joint angles or displacements) and define uniquely the cartesian pose of the cameras.

The task function (or error function) can be defined either in cartesian, configuration or feature spaces (image coordinates). The last choice is simpler and less sensitive to system calibration errors [8]. In this case, the visual servoing problem is stated as the minimization of:

$$e = \left\| \underline{f} - \underline{f}^0 \right\| \tag{1}$$

where $\underline{f}$ is the image feature parameter vector (e.g. position of points, orientation of lines, etc.) and $\underline{f}^0$ is the desired vector. The feature parameter vector is given by:

$$\underline{f} = \mathcal{F}\left(\mathcal{P}_c\left(\underline{q},\underline{p}\right)\right) \tag{2}$$

where $\underline{P}_c = \mathcal{P}_c\left(\underline{p},\underline{q}\right)$ is the relative pose between objects and cameras, and $\mathcal{F}$ is the projection mapping due to the imaging system.

1) Feature sensitivity: Usually, in image-based visual servoing, the control strategy is computed from image features, through the linearization of (2) about the operating point:

$$\delta\underline{f} = \mathbf{J}_q\left(\underline{q},\underline{p}\right) \cdot \delta\underline{q} + \mathbf{J}_p\left(\underline{q},\underline{p}\right) \cdot \delta\underline{p} \tag{3}$$

where $\mathbf{J}_q$ and $\mathbf{J}_p$ are the *image jacobians* or *feature sensitivity matrices*:

$$\begin{cases} \mathbf{J}_q\left(\underline{q},\underline{p}\right) = \frac{\partial\mathcal{F}}{\partial\mathcal{P}_c}\left(\mathcal{P}_c\left(\underline{q},\underline{p}\right)\right) \cdot \frac{\partial\mathcal{P}_c}{\partial\underline{q}}\left(\underline{q},\underline{p}\right) \\ \mathbf{J}_p\left(\underline{q},\underline{p}\right) = \frac{\partial\mathcal{F}}{\partial\mathcal{P}_c}\left(\mathcal{P}_c\left(\underline{q},\underline{p}\right)\right) \cdot \frac{\partial\mathcal{P}_c}{\partial\underline{p}}\left(\underline{q},\underline{p}\right) \end{cases}$$

When the target configuration parameters and their variations are known (or estimated), a suitable kinematic control solution to compensate for the feature motion expressed by (3) is:

$$\delta\underline{q} = -\mathbf{J}_q^{-1} \cdot \delta\underline{f} + \mathbf{J}_q^{-1}\mathbf{J}_p \cdot \delta\underline{p} \tag{4}$$

For non-square jacobians, least-squares solutions can be obtained using appropriate generalized inverses [8]. Despite their simplicity, solutions based solely on kinematics are not adequate for high-performance applications [9]. In our case, robot and target dynamics will be considered.

2) Target location determination: The computation of the jacobians $\mathbf{J}_q$ and $\mathbf{J}_p$, requires that the current configuration parameters, $\underline{q}$ and $\underline{p}$ are known. Even though $\underline{q}$ can be obtained from the robot's internal encoders, the object pose $\underline{p}$ can only be estimated using $\underline{q}$ and the image features $\underline{f}$. We call this process *Forward Visual Kinematics*:

$$\underline{p} = \mathcal{K}\left(\underline{q},\underline{f}\right) \tag{5}$$

With a single camera, one can only obtain 2D information about the position of points in 3D (depth is undetermined). To obtain 3D information, we must use more cameras or *a priori* knowledge of the object structure [3]. Forward visual kinematics usually requires a precise calibration of the robot-camera system and is very sensitive to modeling errors [6]. We will show that, for our visual tracking process, this computation can be avoided.

B. The fixation constraint

In the following discussion we assume that the desired feature vector is chosen such that the error function (1) can be arbitrarily small, i.e., $\underline{f} \approx \underline{f}^0$. This condition enforces a kinematics constraint (or virtual linkage [5]) between camera and target poses. By (5), this constraint can be expressed in the configuration space as:

$$\underline{p}^0 = \mathcal{K}\left(\underline{q},\underline{f}^0\right) \tag{6}$$

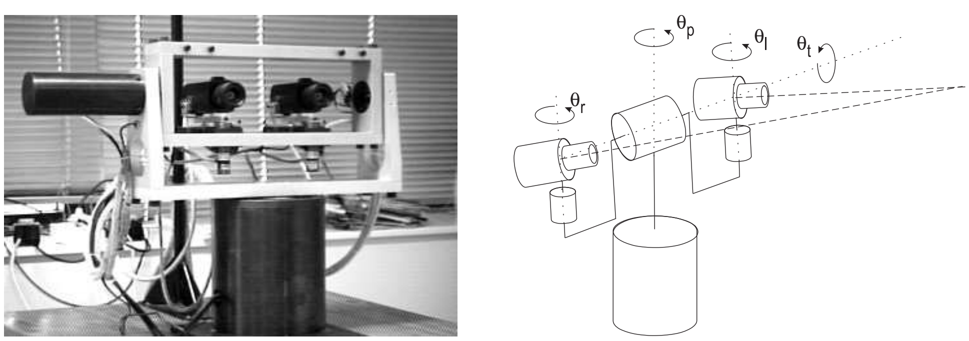

Fig. 1. A picture of the *Medusa* binocular head and its mechanical structure.

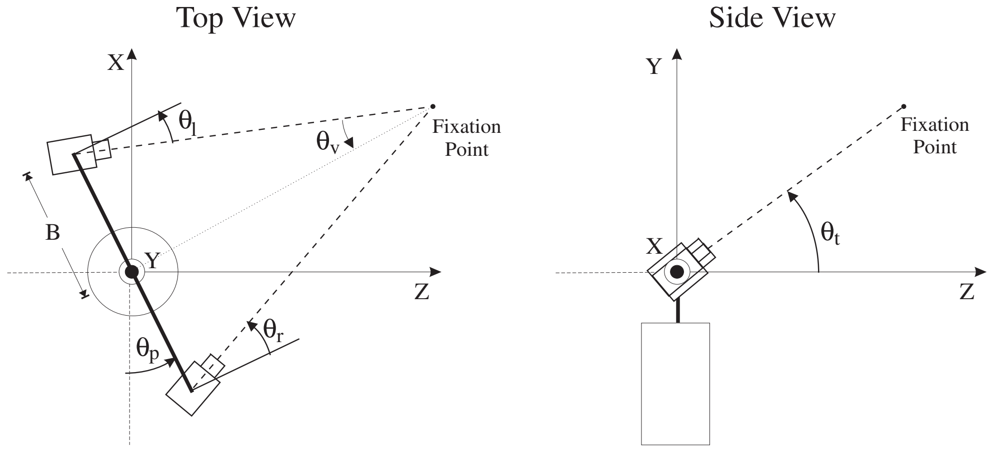

Fig. 2. The robot configuration parameters $(\theta_v, \theta_p, \theta_t)$ and the world reference frame $\{X, Y, Z\}$. The top view shows a frontal vergence fixation $(\theta_v = \theta_r = -\theta_l)$.

We define the *fixation space*, $\mathcal{E}$, as the set of all parameters $\left(\underline{q}, \underline{p}\right)$ where tracking is attained:

$$\mathcal{E} = \left\{ \left(\underline{q}, \mathcal{K}\left(\underline{q}, \underline{f}^0\right)\right); \forall \underline{q} \in \mathcal{C}_q \right\} \tag{7}$$

Points on this manifold are called *fixation points*. During tracking, we always expect to be near the fixation manifold, i.e., $\underline{p} \approx \underline{p}^0$. Then, in (4) we can approximate the general jacobians, $\mathbf{J}_q$ and $\mathbf{J}_p$, by the jacobians in fixation points, $\mathbf{J}_q^0 = \mathbf{J}_q\left(\underline{q}, \underline{p}^0\right)$ and $\mathbf{J}_p^0 = \mathbf{J}_p\left(\underline{q}, \underline{p}^0\right)$:

$$\delta\underline{q} \approx -\left[\mathbf{J}_q^0\right]^{-1} \cdot \delta\underline{f} + \left[\mathbf{J}_q^0\right]^{-1} \mathbf{J}_p^0 \cdot \delta\underline{p}$$

In our application this approximation leads to a substantial reduction of computation complexity.

C. Binocular Head Kinematics

Medusa is a binocular system with four degrees of freedom, designed to replicate the main motions performed by the human visual system. There are revolution joints for two cameras and for neck pan and tilt. The angular position of each joint is denoted by the angles θ_l, θ_r (left and right cameras), θ_p (pan) and θ_t (tilt). Fig. 1 represents the mechanical structure of the binocular head.

We assume frontal vergence configurations, i.e. $\theta_v = \theta_r = -\theta_l$, as represented in Fig. 2. The camera system configuration vector is defined as $\underline{q} = (\theta_v,\ \theta_p,\ \theta_t)$ and the target configuration vector as $\underline{p} = (\rho,\ \phi,\ \gamma)$ (see Fig. 3). Accordingly, target cartesian coordinates are given by:

$$\underline{P}_w = (\rho\cos\gamma\sin\phi,\ \rho\sin\gamma,\ \rho\cos\gamma\cos\phi)$$

We define the image feature vector to include binocular disparity and target average position:

$$\underline{f} = (d_t,\ x_t,\ y_t) = \left(\frac{x_t^1 - x_t^2}{2},\ \frac{x_t^1 + x_t^2}{2},\ \frac{y_t^1 + y_t^2}{2}\right) \tag{8}$$

where $\left(x_t^1, y_t^1\right)$, $\left(x_t^2, y_t^2\right)$ are the projections of the target on images 1 and 2. This feature vector f is computed from image processing algorithms described in Section IV. To obtain the feature sensitivity matrices, we must first express the world point in the head coordinate frame, i.e. estimate its relative pose.

1) Relative pose: In order to have a more compact notation, we define the extended vectors $\underline{P}$ and $\underline{P}_c$, to include information about the two cameras, as:

$$\underline{P} = (\underline{P}_w,\ \underline{P}_w)$$
$$\underline{P}_c = \left(\underline{P}_{c_1},\ \underline{P}_{c_2}\right) = (X_1,\ Y_1,\ Z_1,\ X_2,\ Y_2,\ Z_2)$$

The relative position between the target and cameras, $\underline{P}_c$, is given by the rigid transformation $\underline{P}_c = \mathbf{R}_c \cdot \underline{P} + \underline{T}_c$ with

$$\mathbf{R}_c = \begin{bmatrix} \mathbf{R}_{c_1} & 0 \\ 0 & \mathbf{R}_{c_2} \end{bmatrix} \quad \text{and} \quad \underline{T}_c = \left(\underline{T}_{c_1}, \underline{T}_{c_2}\right)$$

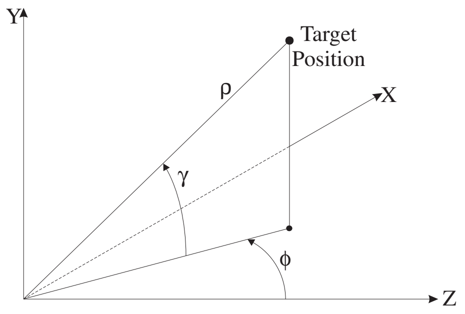

Fig. 3. The target configuration parameters: radius (ρ), azimuth (ϕ) and elevation (γ)

where $\mathbf{R}_{c_j}$ are rotation matrices and $\underline{T}_{c_j}$ are translation vectors. For the *Medusa* binocular head, $\underline{P}_c$ is given by:

$$\underline{P}_c\left(\underline{q},\underline{p}\right) = \begin{bmatrix} \rho\,c_\gamma\,c_v\,s_{\phi-p} + \rho\,c_\gamma\,c_t\,s_v\,c_{\phi-p} + \rho\,s_\gamma\,s_t\,s_v - c_v\,B \\ -\rho\,c_\gamma\,s_t\,c_{\phi-p} + \rho\,s_\gamma\,c_t \\ -\rho\,c_\gamma\,s_v\,s_{\phi-p} + \rho\,c_\gamma\,c_t\,c_v\,c_{\phi-p} + \rho\,s_\gamma\,s_t\,c_v + s_v\,B \\ \rho\,c_\gamma\,c_v\,s_{\phi-p} - \rho\,c_\gamma\,c_t\,s_v\,c_{\phi-p} - \rho\,s_\gamma\,s_t\,s_v + c_v\,B \\ -\rho\,c_\gamma\,s_t\,c_{\phi-p} + \rho\,s_\gamma\,c_t \\ \rho\,c_\gamma\,s_v\,s_{\phi-p} + \rho\,c_\gamma\,c_t\,c_v\,c_{\phi-p} + \rho\,s_\gamma\,s_t c_v + s_v\,B \end{bmatrix} \tag{9}$$

where B represents half the distance between left and right cameras, the symbols c, s are abbreviations for the trigonometric functions cos, sin and the subscripts p, t, v represent the joint angles θ_p, θ_t and θ_v, respectively.

2) Image formation: If we use the perspective camera model given by:

$$\left(x_t^1,\, y_t^1,\, x_t^2,\, y_t^2\right) = \left(-X_1/Z_1,\, Y_1/Z_1,\, -X_2/Z_2,\, Y_2/Z_2\right)$$

the feature parameter of (8) becomes:

$$\underline{f} = \mathcal{F}\left(\underline{P}_c\right) = \frac{1}{2}\left(-\frac{X_1}{Z_1}+\frac{X_2}{Z_2},\, -\frac{X_1}{Z_1}-\frac{X_2}{Z_2},\, \frac{Y_1}{Z_1}+\frac{Y_2}{Z_2}\right) \tag{10}$$

The computation of the partial derivatives of (9) and (10), show that, in general, the feature sensitivity matrices, $\mathbf{J}_q$ and $\mathbf{J}_p$, have a rather complex structure. We restrict our analysis to the fixation subspace $\mathcal{E}$, defined in (7) which is reasonable when dealing with tracking problems, and significantly simplifies the sensitivity matrices.

D. Kinematics under fixation

The goal of binocular tracking is to keep the target projection in the center of both images. Therefore, the desired feature vector is $\underline{f}^0 = \underline{0}$. It is assumed that when initializing the tracking process, the target projection is centered on the images. All target and head configurations for which this condition holds, are fixation points. Therefore, the tracking task consists of keeping the fixation constraint (6), despite target motion. The choice of system coordinates for robot and target shown in Figs. 2 and 3, provide a very simple fixation constraint for the *Medusa* binocular head. The robot pan and tilt angles, (θ_p, θ_t) in Fig. 2, must be equal to target elevation and azimuth, (γ, ϕ) in Fig. 3. The fixation distance, (ρ) in Fig. 3, is obtained by simple trigonometric calculations. The fixation constraint can be written as:

$$\underline{p}^0 = (\rho, \phi, \gamma) = \mathcal{K}\left(\underline{q}, \underline{0}\right) = \left(B\cot\theta_v, \theta_p, \theta_t\right)$$

with B and θ_v as shown in Fig. 2. The previous condition greatly simplifies the target location determination problem of eq. (5). The possible 3D location of fixation points are constrained to:

$$\begin{aligned} \underline{P}_w^0 &= \left(B\cot\theta_v\,c_t\,s_p,\; B\cot\theta_v\,s_p,\; B\cot\theta_v\,c_t\,c_p\right) \quad (11) \\ \underline{P}_c^0 &= \left(0,\; 0,\; B/s_v,\; 0,\; 0,\; B/s_v\right) \end{aligned}$$

Finally, computing the sensitivity matrices for this particular subspace, we obtain:

$$\mathbf{J}_q^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & c_t\,c_v^2 & 0 \\ 0 & 0 & -c_v \end{bmatrix} \qquad \mathbf{J}_p^0 = \begin{bmatrix} -s_v^2/B & 0 & 0 \\ 0 & -c_t\,c_v^2 & 0 \\ 0 & 0 & c_v \end{bmatrix}$$

Since cameras and tilt angles are always well below 90^o (see Fig. 2), $\mathbf{J}_q^0$ is invertible and well conditioned. Additionally, the previous jacobian matrices are **diagonal** and decouple the different motion parameters. We can use separate controllers for the kinematic visual servoing of each joint, which is a major advantage in terms of mathematical simplification and computation time.

III. Dynamic Control

Kinematics controllers have limited performance because they lack knowledge about the actuator dynamics and the robot mechanical structure. We will consider the dynamical aspects and include the effect of target motion in the system dynamic model. The fixation constraint also simplifies the analysis of the system dynamics, which can be represented in terms of the image features.

In the absence of disturbances and friction, a dynamic model can be written as [18]:

$$\mathbf{M}\left(\underline{q}\right)\underline{\ddot{q}} + \underline{h}\left(\underline{q},\underline{\dot{q}}\right) = \underline{\tau}$$

where $\mathbf{M}$ is the inertia matrix, $\underline{h}$ comprises centrifugal, coriolis and gravitational terms, and τ is the vector of applied torques. By mechanical design, gravitational, coriolis and centrifugal terms are negligible in the head dynamics and inertia terms are decoupled for each joint. Furthermore, we use high geared joints and velocity control for the motors, which is effective in linearizing the axis dynamics and eliminating much of the load variation [9]. In these circumstances, joint dynamics are approximated by a second order model. Considering all joints together, we have;

$$\underline{\ddot{q}} = -\Lambda^{-1}\cdot\underline{\dot{q}} + \Lambda^{-1}\cdot\underline{u} \tag{12}$$

with

$$\underline{u} = \begin{bmatrix} u_v \\ u_p \\ u_t \end{bmatrix} \quad \text{and} \quad \Lambda = \begin{bmatrix} \lambda_v & 0 & 0 \\ 0 & \lambda_p & 0 \\ 0 & 0 & \lambda_t \end{bmatrix}$$

where λ_j and u_j are the time constant and the velocity command for each joint j.

For image based visual servoing this model is of little use, as the dynamic model is not directly expressed in terms of the image feature vector. We consider the case of $dim\left(\underline{f}\right) = dim\left(\underline{q}\right) = n$, which includes our particular problem. Expressing (3) using time derivatives, we have:

$$\dot{\underline{f}} = \mathbf{J}_q\,\dot{\underline{q}} + \mathbf{J}_p\,\dot{\underline{p}}$$

Provided that $\mathbf{J}_q$ is invertible, we can write:

$$\dot{\underline{q}} = \mathbf{J}_q^{-1}\,\dot{\underline{f}} - \mathbf{J}_q^{-1}\mathbf{J}_p\,\dot{\underline{p}} \tag{13}$$

Time differentiating again, we obtain:

$$\ddot{\underline{f}} = \mathbf{J}_q\,\ddot{\underline{q}} + \mathbf{J}_p\,\ddot{\underline{p}} + \underline{\delta}$$

where $\underline{\delta}$ represents higher order terms that will be considered as external disturbances. Now, substituting (12) and (13) in the last expression, results:

$$\begin{aligned}\ddot{\underline{f}} = &-\mathbf{J}_q\,\Lambda^{-1}\,\mathbf{J}_q^{-1}\,\dot{\underline{f}} + \\ &+ \mathbf{J}_q\,\Lambda^{-1}\,\mathbf{J}_q^{-1}\,\mathbf{J}_p\,\dot{\underline{p}} + \mathbf{J}_q\,\Lambda^{-1}\,\underline{u} + \mathbf{J}_p\,\ddot{\underline{p}} + \underline{\delta}\end{aligned} \tag{14}$$

In general, this dynamic model is highly coupled and time-variant, because the image jacobians depend on the particular head configuration at a given time instant. However, by introducing the fixation constraint, we can rewrite it in a decoupled and time invariant fashion.

A. Dynamics under fixation

Specifying (14) for the fixation subspace, the matrix $\mathbf{J}_q^0$ is diagonal and can commute with Λ^{-1}:

$$\ddot{\underline{f}} = -\Lambda^{-1}\dot{\underline{f}} + \mathbf{J}_q^0\,\Lambda^{-1}\underline{u} + \Lambda^{-1}\mathbf{J}_p^0\,\dot{\underline{p}} + \mathbf{J}_p^0\,\ddot{\underline{p}} + \underline{\delta}$$

Defining a new control input $\underline{r} = \mathbf{J}_q^0\,\underline{u}$, and image target motion $\underline{w} = \mathbf{J}_p^0\,\dot{\underline{p}}$, both represented in local image plane coordinates, we have now:

$$\ddot{\underline{f}} = -\Lambda^{-1}\dot{\underline{f}} + \Lambda^{-1}\underline{r} + \Lambda^{-1}\underline{w} + \mathbf{J}_p^0\,\ddot{\underline{p}} + \underline{\delta} \tag{15}$$

This is a second order linear approximation of the dynamics, expressed in the image plane, that allows for the design of dynamic visual controllers directly from image measurements. Additionally, since all the matrices are diagonal, we can use separate controllers for each joint.

Therefore, with the fixation constraint we can easily design a variety of closed–loop controllers, whereas using the general model (14) it would be much more difficult. In the next section, we develop a dynamic controller with target motion prediction, as a way to compensate for the unknown term $\underline{w}$ in (15).

B. Independent Joint Control and Motion Estimation

Considering smooth trajectories for the target, we can assume a local constant velocity model (small acceleration) and disregard the term $\ddot{\underline{p}}$ in (15). Additionally, with decoupled dynamics, we can define separate controllers for each joint:

$$\ddot{f}_i = -\frac{1}{\lambda_i}\,\dot{f}_i + \frac{1}{\lambda_i}\,r_i + \frac{1}{\lambda_i}\,w_i$$

Defining the state vector as $\underline{x}_i = \left(f_i,\ \dot{f}_i\right)$, a state-space linear representation can be written as:

$$\begin{cases}\dot{\underline{x}}_i = \mathbf{F}\cdot\underline{x}_i + \mathbf{G}\cdot r_i + \mathbf{G}\cdot w_i \\ \underline{y}_i = \mathbf{H}\cdot\underline{x}_i + \underline{n}_i\end{cases}$$

where $\underline{y}_i$ are the observations and $\underline{n}_i$ the observation noise vector. Also we have:

$$\mathbf{F} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\lambda_i}\end{bmatrix};\quad \mathbf{G} = \begin{bmatrix} 0 \\ \frac{1}{\lambda_i}\end{bmatrix};\quad \mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1\end{bmatrix}$$

From this state-space model we can compute the regulator gains, $\underline{K}$, using standard regulator design techniques, and use the control law:

$$r_i = -\underline{K}\cdot\underline{x}_i$$

In a tracking problem, the target may execute many distinct maneuvers. Without target motion estimation, systematic tracking errors may occur even for simple trajectories. The problem is thus to define a sufficiently general model of the target motion. Assuming smooth motion, during sufficiently small time periods, the target velocity is approximately constant. Thus, for high sampling rates, an appropriate model for target motion is:

$$\dot{w}_i = 0$$

This strategy can provide good tracking performance to several types of motion. To estimate the target motion, we augment the system model with the motion model, yielding:

$$\begin{bmatrix}\dot{\underline{x}}_i \\ \dot{w}_i\end{bmatrix} = \begin{bmatrix}\mathbf{F} & \mathbf{G} \\ \mathbf{0} & 0\end{bmatrix}\begin{bmatrix}\underline{x}_i \\ w_i\end{bmatrix} + \begin{bmatrix}\mathbf{G} \\ \underline{0}\end{bmatrix}\cdot r_i$$

$$\underline{y}_i = \begin{bmatrix}\mathbf{H} & \underline{0}\end{bmatrix}\begin{bmatrix}\underline{x}_i \\ w_i\end{bmatrix} + \underline{n}_i$$

With this model, we are able to estimate (reconstruct) the state consisting of $\underline{x}_i$ and w_i, using a state-space structure [19]:

$$\begin{cases}\dot{\widehat{\underline{x}}}_i = \mathbf{F}\cdot\widehat{\underline{x}}_i + \mathbf{G}\cdot\underline{r}_i + \underline{L}_x\cdot\left(\underline{y}_i - \mathbf{H}\cdot\widehat{\underline{x}}_i\right) \\ \dot{\widehat{w}}_i = L_w\cdot\left(\underline{y}_i - \mathbf{H}\cdot\widehat{\underline{x}}_i\right)\end{cases}$$

The computation of the estimator gains, $\underline{L}_x$ and L_w can be done with standard state estimation techniques, where the system model is augmented by the target motion model. However notice that the previous model is not used for regulator design, which is obtained using the $\mathbf{F}$ and $\mathbf{G}$ matrices of the unaugmented system. Our plan is to use

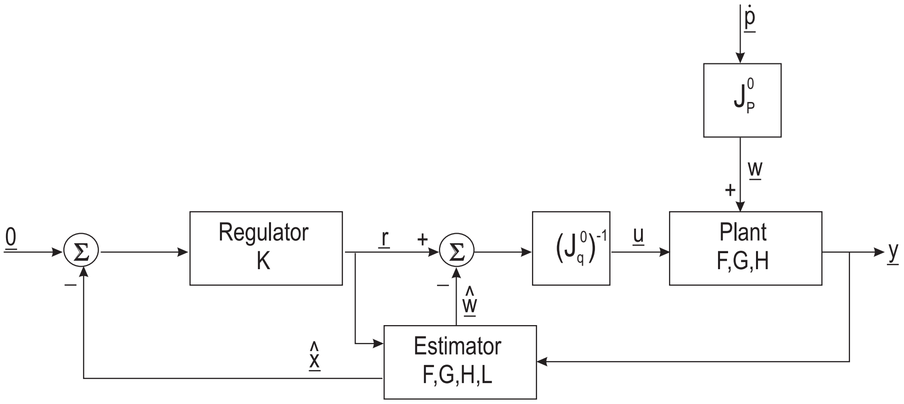

Fig. 4. Block diagram of the dynamic control system.

the estimated value of w_i in a feedforward control scheme to reduce errors. Fig. 4 presents a block diagram of the full control system.

In summary, this section described a control architecture which permits the use independent dynamic controllers for each joint of the binocular head, using as input the motion of visual features (positions and velocities) directly from the image plane – no intermediate (full or partial) 3D reconstruction is required. In the next section we describe how to extract the motion parameters from the images. We will estimate disparity, $\hat{d}_t$, horizontal and vertical positions and velocities, $\hat{x}_t, \hat{\dot{x}}_t, \hat{y}_t, \hat{\dot{y}}_t$. Recall that our feature vector was defined in Eq. (8) as $f = (d_t, x_t, y_t)$.

IV. Perceptual processes

We start by introducing the log–polar mapping that has the main advantage of defining a focus of attention at the center of the visual field, which perceptually "enhances" tracked objects. We present a new disparity estimation algorithm and show the advantages of the log–polar mapping in the acquisition of disparity estimates, both analytically and experimentally. We also explain how we obtain position and velocity measures from image target motion.

A. Perceptual processes on space variant images

In many biological systems, the main reason for ocular movements is the space variant resolution of the retina. In the human visual system, the receptors at the retina are distributed with increasing density toward the center of the visual field (the fovea). Eye movements are used by the human oculomotor system to point the central high resolution fovea in the direction of the scene areas which attract attention.

Apart from a few exceptions [20], [21], most active vision systems use cartesian image representations. Instead, the system described in this paper is based on a log-polar space variant image sampling. One advantage of this kind of sampling is data reduction. This is obtained by reducing the resolution at the image periphery, thus high resolution in the fovea improves the performance of the tracking process. We map 128x128 cartesian images to 64x32 log-polar images and achieve 8 times increase in efficiency (both storage and speed).

The log-polar transformation is a conformal mapping from the points on the *cartesian* plane (x, y) to points in the *log-polar* plane (ξ, η) [15]. The mapping is described by :

$$(\xi,\ \eta) = (\log(\rho),\ \theta)$$

where ρ and θ denote the pixel polar coordinates. The log-polar mapping is not defined in the neighborhood of the origin due to the singularity when ρ tends to zero:

$$\xi \in [\log(\rho_{min}), \log(\rho_{max})],$$

where ρ_{max} depends on the cartesian plane dimension. Fig. 5 shows the discretized cartesian and log-polar planes.

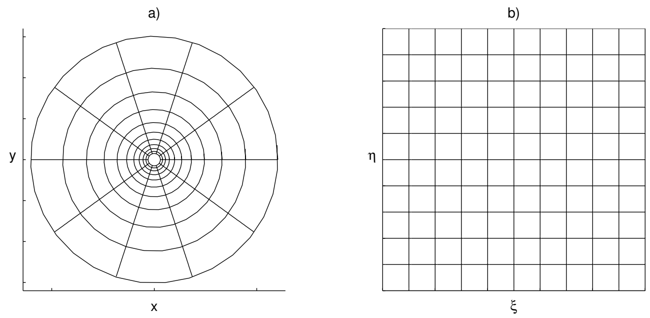

Fig. 5. Space variant mapping from the cartesian plane, (a), to the log-polar plane, (b).

For tracking, the main advantage of the log-polar sensor is that objects occupying the central high resolution part of the visual field become dominant over coarsely sampled background elements in the image periphery. This embeds an implicit focus of attention in the center of the visual field where the target is expected to be most of the time.

B. Disparity Estimation

Some algorithms estimate disparity by *cepstral* filtering [22] and phase correlation [23]. These techniques are very time consuming, and hardly suitable for real-time applications. Furthermore, with cartesian images, if the object of interest is small, the disparity of the background can lead to erroneous estimates. Alternatively, with space variant images, the target region becomes dominant [16]. In this paper we exploit the correlation of log-polar images to extract the disparity cues. Correlation is very fast to compute and we will show how to adjust the precision of the disparity estimation as a function of the computation time available.

1) Correlation in the Log-polar Domain: Various authors have shown experimentally the advantages of applying correlation techniques to log-polar images [20], [24], [21]. We now present a theoretical analysis that explains why much better results are obtained for log-polar images than for cartesian images.

Among the various correlation measures to compute the similarity between images, one of the most used is the Sum of Squared Differences (SSD). Consider the continuous form of the SSD:

$$\mathrm{SSD}\left(I_1, I_2\right) = \iint\limits_{x,y} \left(I_1\left(x,y\right) - I_2\left(x,y\right)\right)^2 dx\, dy \qquad (16)$$

This function determines the distance between I_1 and I_2 in the space of continuous images and will be used in the next theorem to derive an important relation between the correlation computed on cartesian images and on non-uniform resolution images.

Theorem 1: Consider two non-uniform resolution images, $I_1'(w,z)$ and $I_2'(w,z)$, obtained from cartesian images $I_1(x,y)$ and $I_2(x,y)$, through the differentiable mapping $(w,z) = \mathbf{T}(x,y)$. Let $|\mathbf{J}|$ be the jacobian of $\mathbf{T}$, and SSD the correlation operator defined by (16). Then, the correlation of the non-uniform resolution images is related to the correlation of the cartesian images by:

$$\mathrm{SSD}\left(I_1', I_2'\right) = \mathrm{SSD}\left(I_1\sqrt{|\mathbf{J}|}, I_2\sqrt{|\mathbf{J}|}\right)$$

In the particular case of log-polar images we have the following result, for $(x,y) \neq (0,0)$:

$$\mathrm{SSD}\left(I_1', I_2'\right) = \mathrm{SSD}\left(\frac{I_1\left(x,y\right)}{\sqrt{x^2+y^2}}, \frac{I_2\left(x,y\right)}{\sqrt{x^2+y^2}}\right) \qquad (17)$$

Proof: The correlation of two space variant images results in:

$$\mathrm{SSD}\left(I_1', I_2'\right) = \iint\limits_{w,z} \left(I_1'\left(w,z\right) - I_2'\left(w,z\right)\right)^2 dw\, dz$$

Applying the coordinate change $(w,z) = \mathbf{T}(x,y)$, we obtain:

$$\mathrm{SSD}\left(I_1', I_2'\right) = \iint\limits_{x,y} \left(I_1\left(x,y\right) - I_2\left(x,y\right)\right)^2 \left|\mathbf{J}\left(x,y\right)\right| \, dx\, dy = $$

$$= \mathrm{SSD}\left(I_1\sqrt{|\mathbf{J}|}, I_2\sqrt{|\mathbf{J}|}\right)$$

The log-polar mapping jacobian is given by:

$$\left|\mathbf{J}\left(x,y\right)\right| = \begin{vmatrix} \frac{\partial\xi}{\partial x} & \frac{\partial\xi}{\partial y} \\ \frac{\partial\eta}{\partial x} & \frac{\partial\eta}{\partial y} \end{vmatrix} = \begin{vmatrix} \frac{x}{x^2+y^2} & \frac{y}{x^2+y^2} \\ \frac{y}{x^2+y^2} & \frac{x}{x^2+y^2} \end{vmatrix} = \frac{1}{x^2+y^2}$$

which results in (17) ■

This theorem shows that the correlation between two log-polar images corresponds to the correlation of the cartesian images weighted by the inverse distance to the image center. This implicit weighing function is represented in Fig. 6. We can observe the dominance of the areas close to the image center, which implements a direct focus of attention.

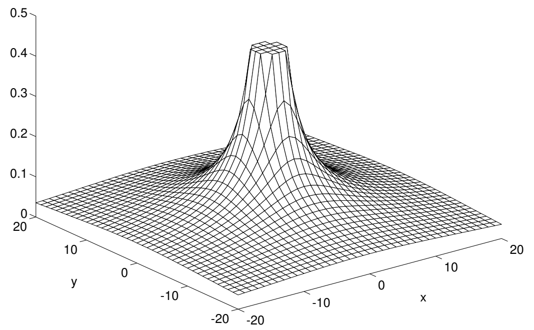

Fig. 6. Weighing window corresponding to the correlation of log-polar images. The center is truncated due to the mapping singularity.

Consider the case of a fixated target such that its projection is centered in both images and disparity is low for all object points. Assume that the image background has large disparities (typical situation in a tracking experiment). In this case, image correlation has one component due to the background and another due to the target:

cartesian: $\mathrm{SSD}\left(I_1, I_2\right) =$

$$\underbrace{\iint\limits_{B} \left(I_1 - I_2\right)^2}_{\text{background component}} + \underbrace{\iint\limits_{T} \left(I_1 - I_2\right)^2}_{\text{target component}}$$

log–polar: $\mathrm{SSD}\left(I_1', I_2'\right) =$

$$\underbrace{\iint\limits_{B} \left(I_1\sqrt{|\mathbf{J}|} - I_2\sqrt{|\mathbf{J}|}\right)^2}_{\text{background component}} + \underbrace{\iint\limits_{T} \left(I_1\sqrt{|\mathbf{J}|} - I_2\sqrt{|\mathbf{J}|}\right)^2}_{\text{target component}}$$

where B represents the background region and T is the target region.

Since, the target disparity is low, its component can be disregarded and the correlation value depend mainly on the background area and disparity. When the target size diminishes, the background component increases faster for the cartesian case than for the log-polar case due to the jacobian weighing function J. In the next pages we show

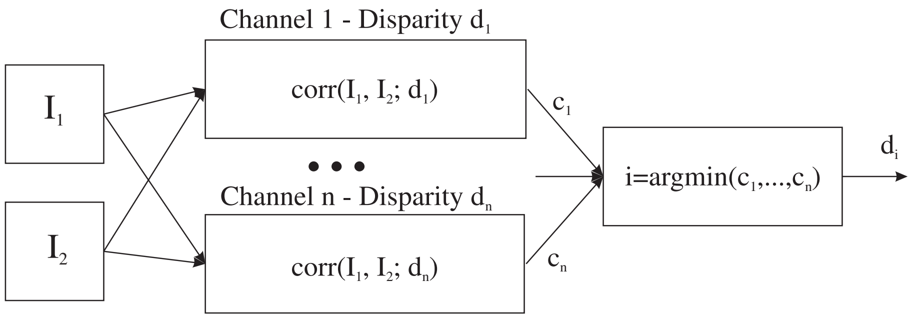

Fig. 7. Block diagram of the channel based disparity estimation algorithm.

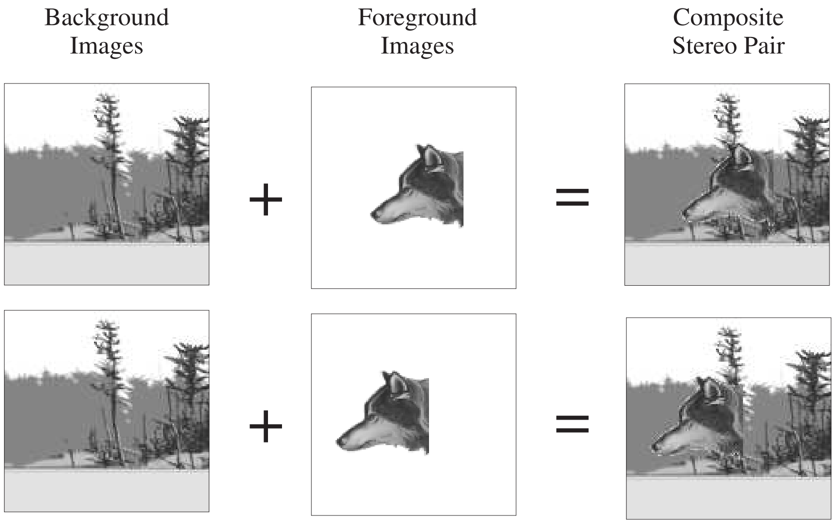

Fig. 8. Example of stereo image pairs created for disparity estimation tests.

that disparity estimation via correlation of log-polar images can deal with much smaller objects.

2) Channel based disparity estimation: To control the vergence angle of the cameras, we must compute the target horizontal disparity. Let us consider the discrete horizontal cross-correlation function, defined as:

$$\operatorname{corr}(I_1, I_2; d) = \operatorname{SSD}\left[I_1\left(x + \frac{d}{2}, y\right), I_2\left(x - \frac{d}{2}, y\right)\right] \quad (18)$$

where d represents disparity. This cross-correlation process consists of translating the cartesian images horizontally before computing the SSD. In the log-polar domain, this translation must be replaced by an equivalent but more complicated warping. If the target disparity d_t is the dominant horizontal disparity between the images, then the cross-correlation function (18) has a minimum for $d = d_t$. Some authors have proposed to locally minimize the cross-correlation function based on gradient–descent techniques [17]. This approach suffers from problems inherent to local optimization strategies: high frequency fluctuations on the correlation function lead to erroneous optimization steps and local minima. Additionally, the range of disparities that ensures convergence is small [16].

Our algorithm is based on a global search for the horizontal disparity that minimizes the SSD. This search is restricted to a limited set of hypotheses, $D = \{d_1, ..., d_n\}$ that we designate *disparity channels*. To improve the precision of the algorithm, the cardinality of D can be increased as a function of the available computation time. Fig. 7 shows a block diagram of this process. Each channel computes the correlation (SSD) on the image pair for its associated disparity. The estimated disparity is that of the channel with minimal output.

$$\hat{d}_t = \arg\min_{d \in D}\left[\operatorname{corr}(I_1, I_2; d)\right]$$

In [25] a similar theory is proposed to explain the behavior of the human vergence system.

As disparity is small most of the time, the set D samples small disparities in a dense manner, while large disparities are considered in a more coarse manner. Vergence is achieved in two phases: after initialization or sudden motion of the target, the vergence angle is roughly attracted to the highly sampled disparity range; within this range, precise servoing is achieved.

3) Results on disparity estimation: To assess the performance of the proposed disparity estimation method, and compare the results with cartesian and log-polar images, we performed experiments with sequences of image pairs with ground–truth data. Each pair of images consists of a common background and a foreground object, shown in

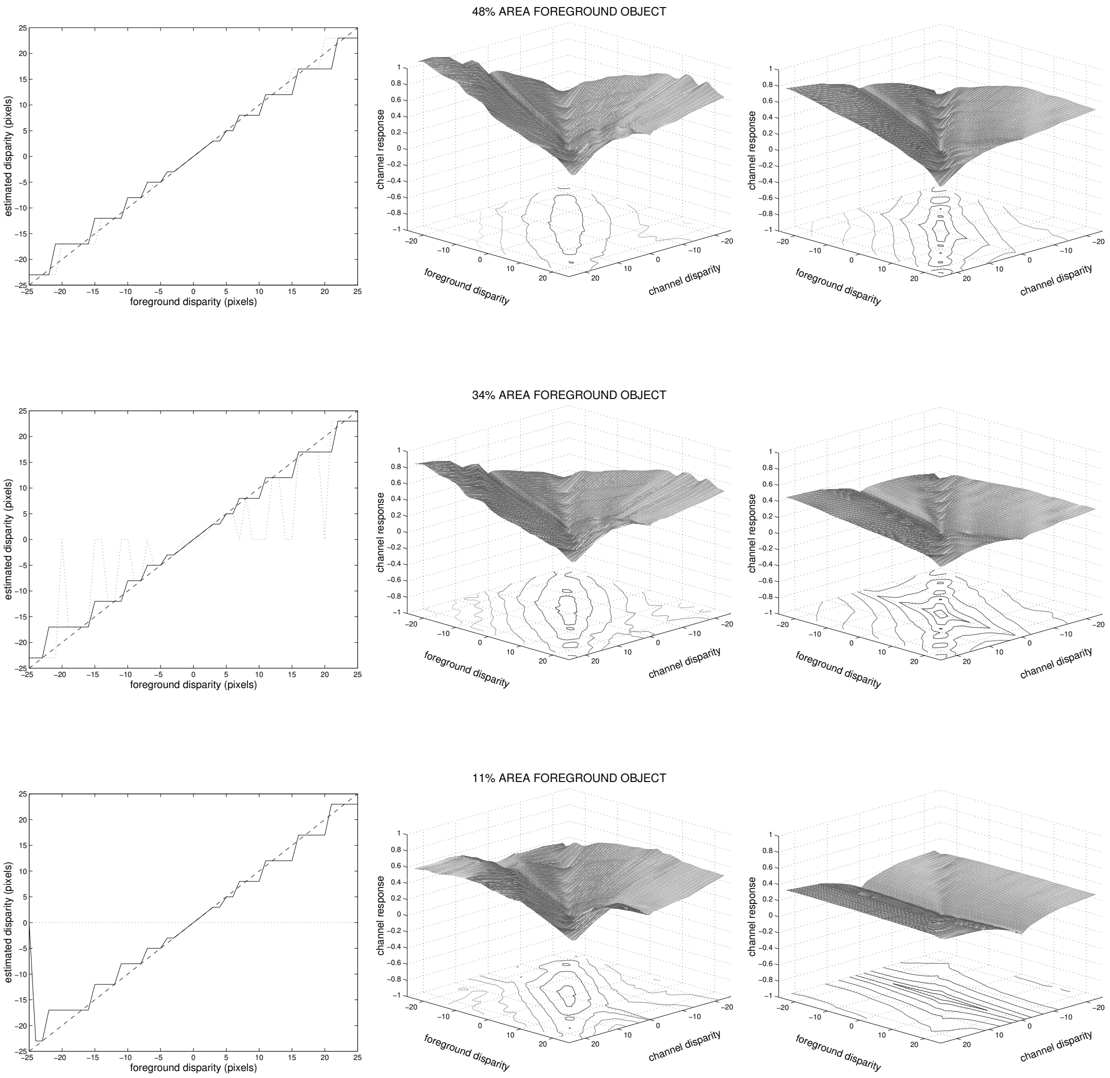

Fig. 9. Results of disparity estimation with foreground objects occupying 48% (top row), 34% (middle row) and 11% (bottom row) of image area, with true disparity ranging from -25 to 25 pixels. Left column: estimated disparity computed with log-polar (solid line) and cartesian images (dotted line); true disparity (dashed line) is also represented. Middle and right columns: correlation surfaces showing the output of each channel as a function of true foreground disparity (log-polar and cartesian cases, respectively).

Fig. 8. For all sequences, the background has zero disparity and the foreground disparity ranges from -25 to 25 pixels. We have built sequences with foreground objects of different sizes. The set of disparity channels considered is $D = \{0, \pm1, \pm2, \pm3, \pm5, \pm8, \pm12, \pm17, \pm23\}$.

For each image pair in a sequence, the algorithm is applied and we analyze the output value of each channel and the estimated disparity. Fig. 9, shows results both for log-polar and cartesian images: 2D plots of the estimated disparity and 3D/contour plots of the output of all channels, as the true disparity changes.

Three sequences are considered:

- The top row shows results obtained with a **large target**, (**48%** of the image). In general, the estimated disparity is close to the true disparity, both for cartesian and log-polar images. However, for log-polar images, the correlation surface is sharper in the neighborhood of the true disparity, which improves the robustness of the estimation process. In the cartesian case, the valley corresponding to zero disparity channel shows a strong response due to the disparity of the background.
- With a **medium target**, occupying **34%** of the image (middle row), the influence of the background becomes stronger with cartesian images and disparity estimation often fails. However, with log-polar images there is no noticeable difference to the previous case.
- The bottom row shows results for a **small target** (**11%** of the image). The valley corresponding to the zero disparity becomes dominant in the cartesian case but the log-polar plot still follows the true disparity except for -25 pixels. As the object size diminishes, there is a gradual degradation in the estimation of large disparities, which does not happen for small disparities. Even for targets occupying only 5% of the image, disparities between -10 and 10 are still detected.

In summary, the cartesian case cannot deal with small objects because the background pixels are dominant even in fixation situations. The non-uniform resolution representation is an elegant solution to this problem. Since the target usually occupies the high resolution region of the log-polar images, its area is dominant in the computation of the correlation measure, hence reducing the influence of the background.

C. Position and Velocity Estimation

Estimates of the target position and velocity are required to control the gaze direction, driving the *pan* and *tilt* angles. We must first segment the target points from the background or other objects in the visual field. Assuming that the target is under vergence, its location in each of the stereo images is approximately the same, and the segmentation problem can be addressed by zero disparity filtering [14]. Algorithms for the extraction of zero disparity points in the images were developed based on vertical gradient matching, adapted to log-polar images. Fig. 10 shows the output of the segmentation process on a pair of real images.

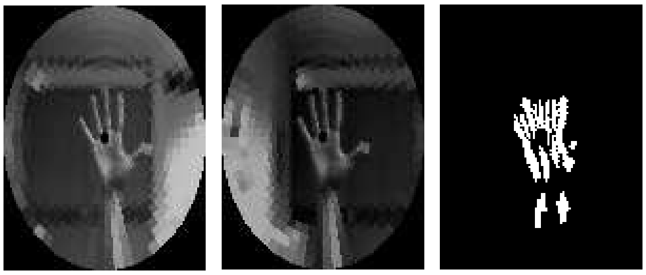

Fig. 10. Figure-ground segmentation using zero disparity filtering in log-polar images. From left to right: left camera image, right camera image and filter output.

Once the target points have been extracted, position and velocity estimates are taken from the segmented areas in the log-polar images and expressed in cartesian coordinates to be used by the dynamic controllers designed in Section III.

1) Position error: The position error is defined by the difference between the position of the object and the coordinates of the image center. We use the centroid of the segmented object points to determine its position:

$$\hat{x}_t = \frac{\iint_S x\,dx\,dy}{\iint_S dx\,dy} \qquad \text{and} \qquad \hat{y}_t = \frac{\iint_S y\,dx\,dy}{\iint_S dx\,dy}$$

where S is the image region corresponding to segmented points.

2) Velocity error: Velocity estimation can be done simply by differentiating position data. However, noise in position estimates would be amplified. Alternatively, we use the optical flow [26] which provides velocity estimates for each point in the image. The integration of the normal flow vectors[1] over the target area, provides a stable velocity measure and good noise rejection. Thus, our sensoring strategy relies on the computation of the average of the normal flow vectors, $\mathbf{v} = (v_x, v_y)$, in the segmented area S:

$$\hat{\dot{x}}_t = \frac{\iint_S v_x\,dx\,dy}{\iint_S dx\,dy} \qquad \text{and} \qquad \hat{\dot{y}}_t = \frac{\iint_S v_y\,dx\,dy}{\iint_S dx\,dy}$$

[1]Normal flow is the projection of the optical flow vector in the direction of the image gradient. Is the only observable component of the optical flow due to the aperture problem [26].

V. Results

In this section we present extensive experimental results of our approach. The first experiment illustrates the application of the fixation constraint to controllers with and without kinematics or dynamics compensation. With the use of simulated target trajectories, the results are independent of the imaging system used (cartesian or log-polar).

In the second set of experiments we use real targets and the log-polar image representation. Cartesian images are not used here because the integrated system is highly dependent on a correct vergence at the target, for which log–polar images are advantageous as shown in Section IV.

The system is implemented on the *Medusa* binocular head [27], shown in Fig. 1, running at full video rate (50 Hz) on a Pentium II 300MHz computer, without any special processing hardware. The log-polar mapping from 128x128 cartesian to 32x64 log-polar images takes about 3 ms. This delay is highly compensated by the reduction achieved (8 times) in the remaining computations.

A. Control under fixation

In this first set of experiments we simulate 3D target trajectories to perform repeatable tests on the kinematic and dynamic closed loop control of the binocular head. *The real robot head is used inside the control loop.*

1) Kinematics compensation under fixation: Under the fixation constraint, the image jacobian has a simple structure, and can be used in the control law. To illustrate the effect of including the inverse jacobian in the visual control loop, we consider the following kinematics control strategies:

$$\delta \underline{q} = -\mathbf{K}\left[\mathbf{J}_q^0\right]^{-1} \cdot \delta \underline{f} \qquad \textit{versus} \qquad \delta \underline{q} = -\mathbf{K} \cdot \delta \underline{f}$$

where $\mathbf{K}$ is a constant gain matrix, adjusted to obtain a good transient response. Recall that the jacobian $\mathbf{J}_q^0$ has a strong effect in the loop gain from the x_t coordinate to the pan angle, when tilt and vergence angles are large, whereas for small vergence and tilt angles the loop gain is approximately unitary:

$$\delta\theta_p = \cos(\theta_t) \cdot \cos(\theta_v)^2 \cdot \delta x_t$$

The results are shown in Fig. 11. We illustrate two cases: (1) a distant target moves in a step-like manner between the 3D points $(1, 0, 10)$ and $(-1, 0, 10)$, affecting only the x image coordinate and the pan angle; (2) a nearby target moves between points $(0.05, 0.4, 0.5)$ and $(-0.05, 0.4, 0.5)$, corresponding to large tilt and vergence angles. For the distant target both controllers behave similarly, since the jacobian gain is almost unitary. For targets close to the observer, kinematics compensation is advantageous and produces a faster response.

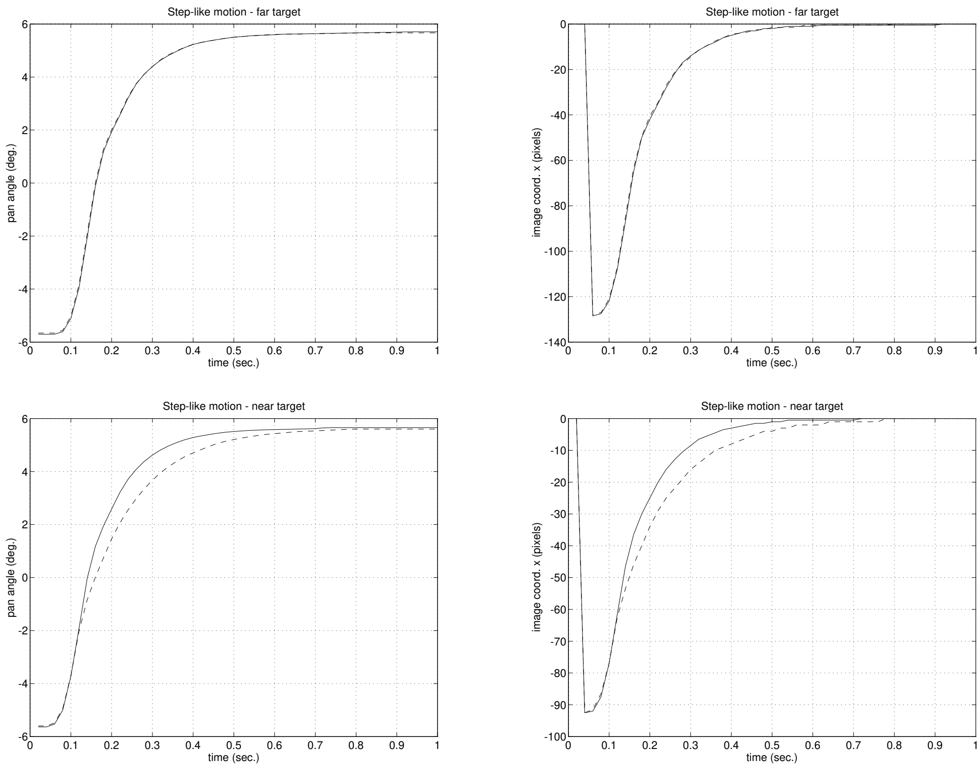

Fig. 11. Evolution of the pan angle and x_t image feature to a step-like input. Top row: distant target. Bottom row: nearby target. Solid line: with kinematics compensation. Dashed line: without kinematics compensation.

2) Dynamic controller under fixation: In Section III we used the fixation constraint to derive and image–based decoupled time–invariant model that was extended to incorporate a model of the target motion. Here we show that the dynamic controller developed in Section III-B is particularly suited to track slowly changing target motions. For comparison purposes we use the proportional controller with kinematics compensation as defined in the previous experiment.

The adopted model parameters are $\lambda = 0.05$ s and the sampling period is $T = 0.02$ s. The regulator and estimator gains, K and L, were obtained from LQR and LQE design [19].

2a) Constant velocity: Fig. 12 shows the evolution of the pan and tilt angles and the (x, y) image coordinates, for a piecewise constant velocity trajectory between the 3D points $(1, -1, 6)$ and $(-1, 1, 6)$. The error in the image features is smaller on average for the model based controller. Although, during the target velocity discontinuities, the error exhibits a short spike, for constant velocity periods it is much smaller than the error for the proportional controller.

2b) Elliptical motion – 1 rad/s: The target describes an elliptical trajectory with an angular velocity of 1 rad/s. Fig. 13 shows the error in the x image coordinate for each controller and also its estimated motion. In this situation, due to the absence of sudden changes in the target motion, the error for the model based controller is always much smaller than for the proportional controller.

2c) Fast motions: Fig. 14 shows a 15 rad/s elliptical and

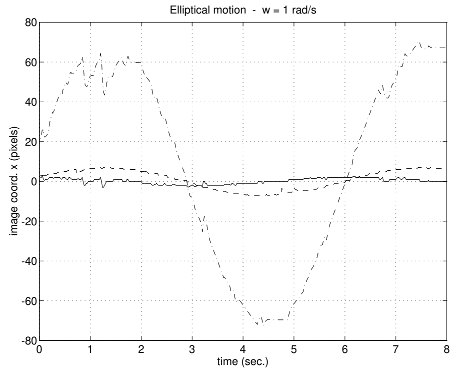

Fig. 13. Evolution of the x_t feature in response to an elliptical target motion. Solid line: dynamic controller. Dashed line: proportional controller. Dash-dot line: estimated target motion.

step-like trajectories. These situations represent sudden changes of target motion and position discontinuities. The elliptical case represents a motion with a higher frequency than the bandwidth of the system and both controllers are unable to track the target. For the step-like trajectory, the model based controller performs poorly because the slowly changing velocity model is no longer adequate at motion discontinuities.

In summary, for motions with slowly changing velocities, the proposed controller provides a better tracking since it estimates target motion with a constant velocity model and compensates for it in the control law. When strong velocity

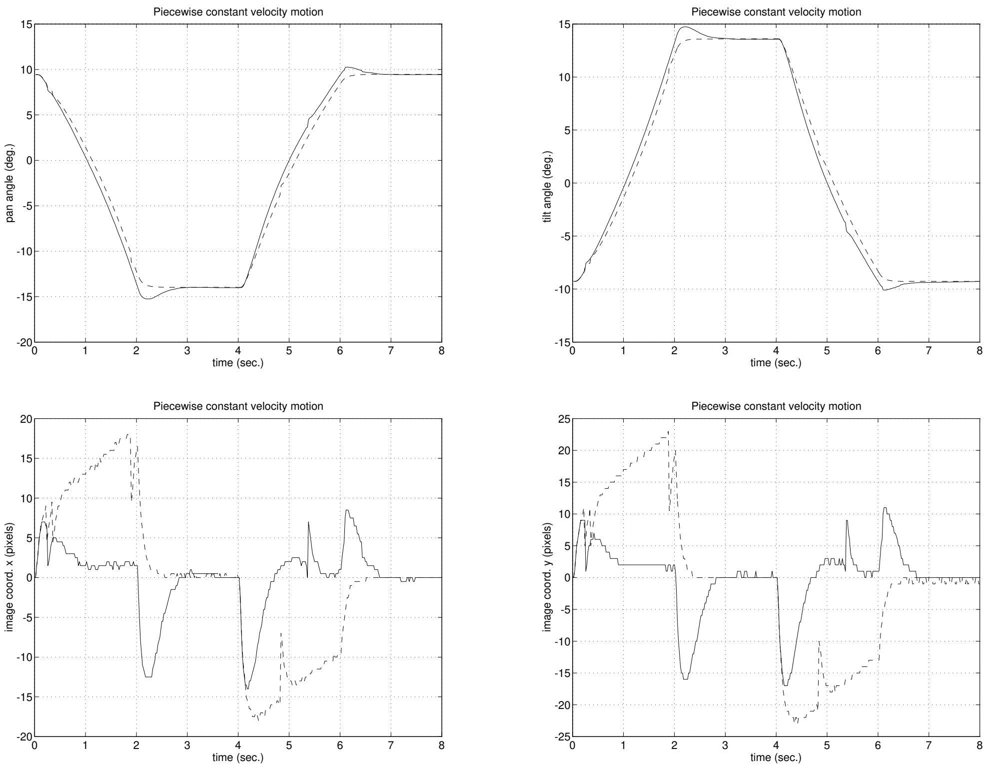

Fig. 12. Evolution of the pan and tilt angles and of x_t and y_t image features to a piecewise constant velocity input. Solid line: dynamic controller. Dashed line: proportional controller.

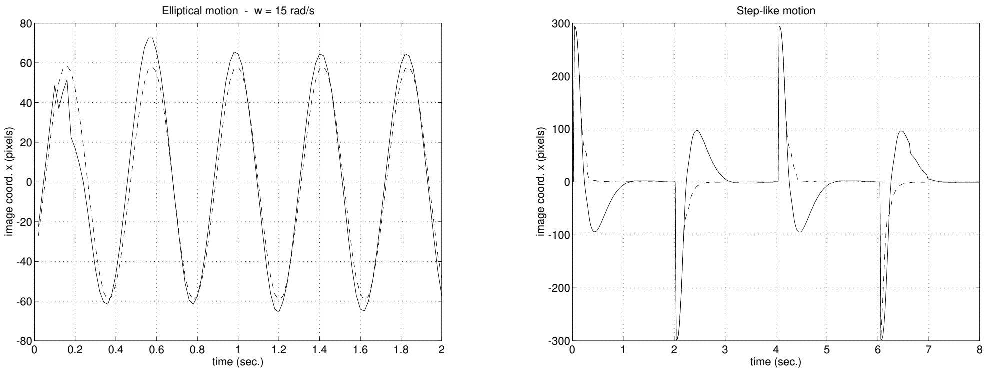

Fig. 14. Evolution of the x_t feature in response to fast changing motions. Solid line: dynamic controller. Dashed line: proportional controller.

changes or position discontinuities occur, its performance degrades. For very fast target motions other control strategies could be used. For instance, in the visual systems of many animals, fast redirections of gaze are performed by saccadic motions, executed in an open-loop manner (visual information is not acquired for control purposes during the saccade). Note that with the model based on the fixation constraint one can design other controllers for more general target motions.

B. *Real targets*

In this set of experiments the developed perceptual algorithms are used to extract the feature vector from the real images. As the target motion is not controlled accurately, we do not have quantitative ground truth information to compare with. However, qualitative evaluation of the results enables us to assess the system performance.

1) Linear motion: We first used an approximately linear target motion, and reconstructed the 3D trajectory[2], using Eq. (11) and the head joint angles. Figs. 15 to 17 show the time evolution of the joint angles and the reconstructed 3D trajectory. Fig. 15 corresponds to motion in depth and, as expected, only the vergence angle has significant variations. In Fig. 16 the target moves horizontally, perpendicularly to the initial gaze direction. Most of the tracking motion is achieved by the pan joint. Fig. 17 shows a diagonal

[2]Notice that system calibration and target motion were only made in a coarse fashion.

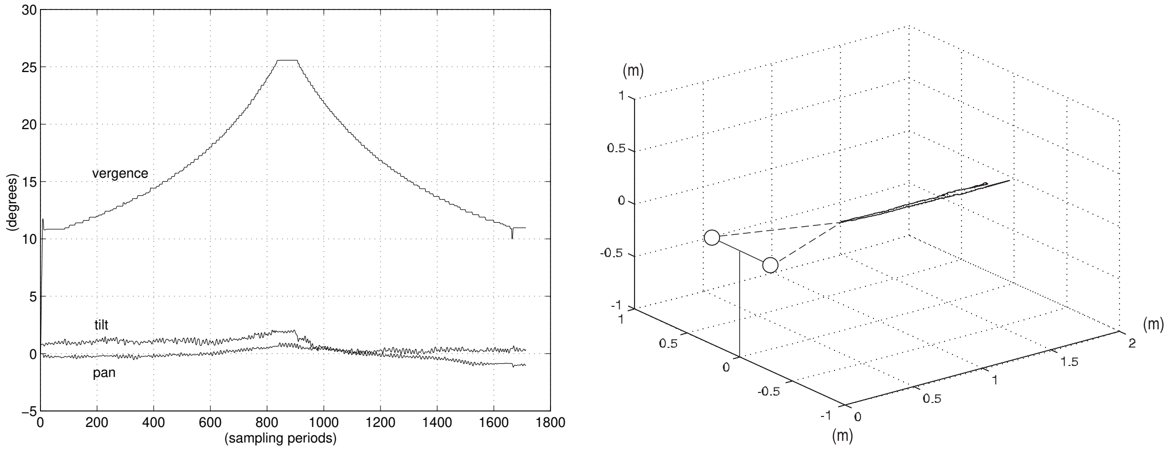

Fig. 15. Depth tracking. Time evolution of the joint angles (left) and trajectory reconstruction (right).

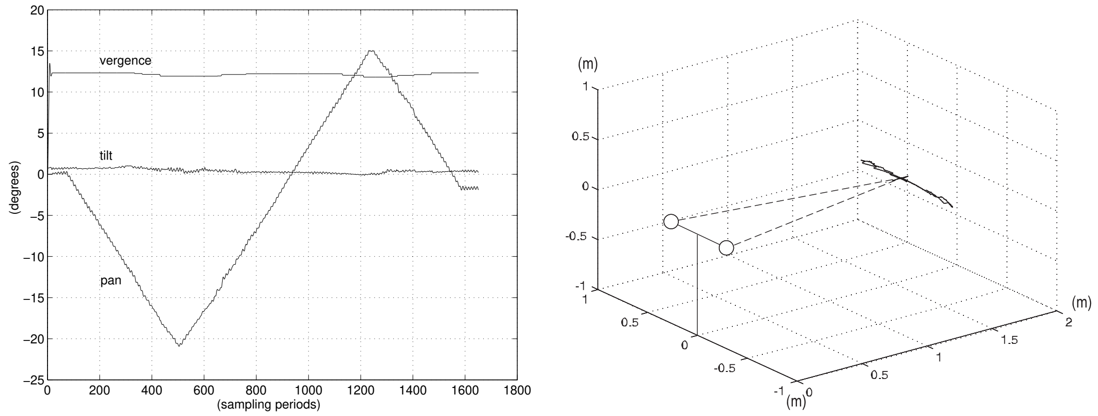

Fig. 16. Fronto-parallel tracking. Time evolution of the joint angles (left) and trajectory reconstruction (right).

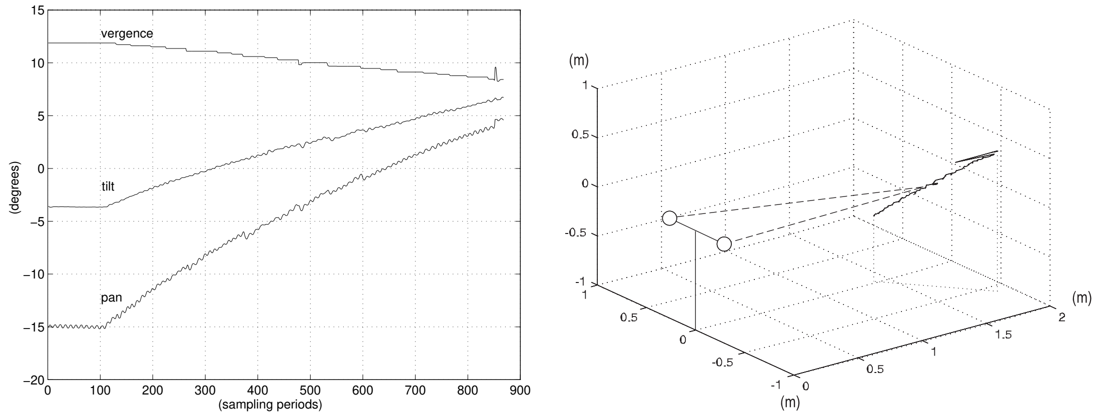

Fig. 17. Diagonal tracking. Time evolution of the joint angles (left) and trajectory reconstruction (right).

trajectory where tracking requires the contribution of all joints. In all these cases, the reconstructed trajectory is qualitatively close to the expected.

2) General motion: Experiments were also made for general target motions. Results are presented by a sequence of images obtained during the tracking process. Both sequences are presented in Fig. 18. The first sequence shows the tracking of a human hand. In the beginning, fixation is stable on the door at the back of the room (images 1 and 2). Once the hand gets close to the gaze direction, the system starts verging and tracking the target. Notice that the hand is always kept very close to the center of the images despite complex background motion and target rotation and scaling. The second sequence shows a reliable tracking of a human face with scale and pose changes.

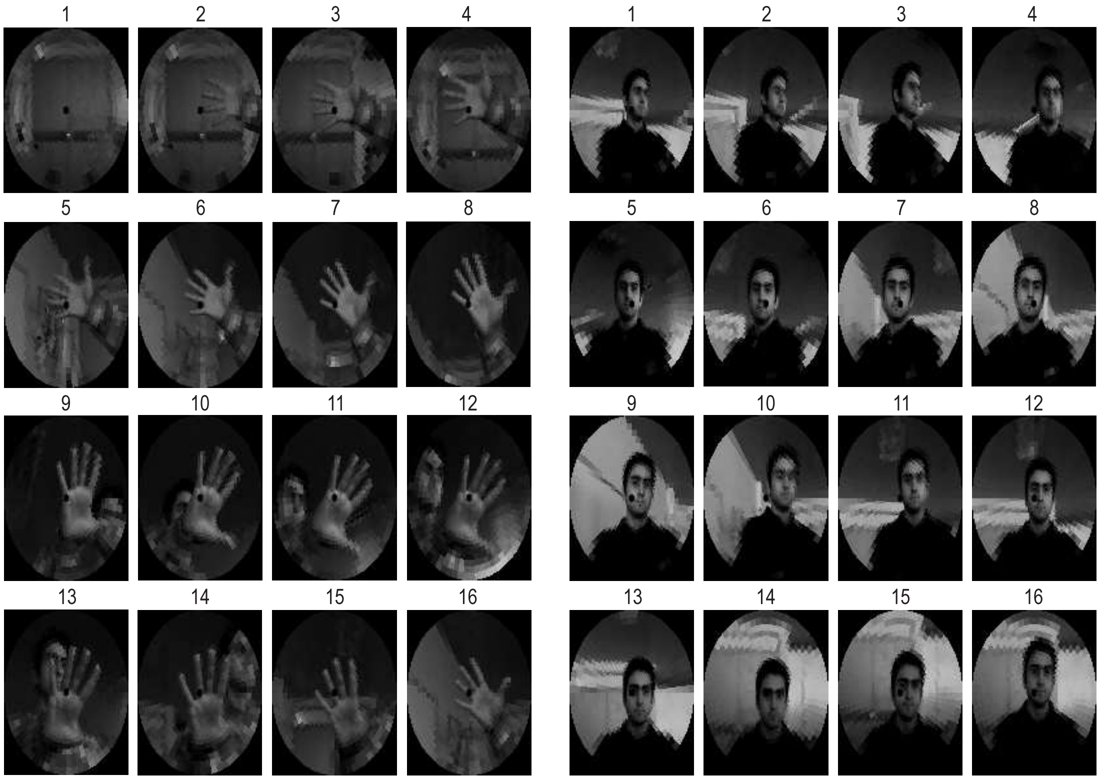

Fig. 18. Hand tracking (left) and face tracking (right).

VI. Conclusion

This paper presented a real-time binocular tracking system which copes with objects of different shapes and motions. The tracking problem is divided in two main issues: the control of the robot head and the estimation of the target motion from visual information.

In the control problem we addressed both the kinematics and dynamics aspects. Kinematics is analyzed for general configurations of the robot and objects in the environment. However, tracking imposes a particular position on the objects relative to the cameras – the *fixation constraint*. With this constraint we show that system kinematics can be decoupled and simplified, and system dynamics can be expressed using image coordinates. We derive a linear state-space dynamic model, which includes a target motion model, and design a dynamic controller with motion estimation.

The perception strategies use log-polar images that focus the attention at the center of the visual field, where the target is usually located. Distracting elements in the background become less important in the overall system performance. This aspect is illustrated with theoretical and experimental results in a new disparity estimation algorithm. Additionally, log-polar images allow much faster algorithms and real-time functionality.

Extensive experimental results were presented to illustrate perceptual and control strategies in a fully working system. Results show that a careful integration of perception and control strongly improves robustness in visual servoing tasks.

Future work will exploit the consequences of the fixation constraint to other problems, and use more general models of the target motion. We will also focus on using run-time estimates of the reliability of sensor signals directly in the control system parameters, to further improve the tracking performance.

References

[1] C. Fermüller and Y. Aloimonos, "The role of fixation in visual motion analysis," *IJCV*, vol. 11, no. 2, pp. 165–186, October 1993.

[2] Y. Aloimonos and A.Bandyopadhyay, "Active vision," in *Proc. ICCV*, 1987, pp. 35 – 54.

[3] F. Chaumette, P. Rives, and B. Espiau, "Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing," in *Proc. ICRA*, 1991, pp. 2248–2253.

[4] K. Brunnström, J. Eklundh, and T. Uhlin, "Active fixation for scene exploration," *IJCV*, vol. 17, no. 2, pp. 137–162, February 1996.

[5] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robotics and Automation*, vol. 8, no. 3, pp. 313–326, June 1992.

[6] G. Hager, W. Chang, and A. Morse, "Robot hand-eye coordination based on stereo vision," *IEEE Control Systems Magazine*, vol. 15, no. 1, pp. 30–39, 1995.

[7] N. Papanikolopoulos, B. Nelson, and P. Kosla, "Six degree-of-freedom hand/eye visual tracking with uncertain parameters," *IEEE Trans. Robotics and Automation*, vol. 11, no. 5, pp. 725–732, October 1995.

[8] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robotics and Automation*, vol. 12, no. 5, pp. 651–670, October 1996.

[9] P. Corke and M. Good, "Dynamic effects in visual closed-loop systems," *IEEE Trans. Robotics and Automation*, vol. 12, no. 5, pp. 671–683, October 1996.

[10] D. Gennery, "Visual tracking of known 3-dimensional objects," *IJCV*, vol. 7, no. 3, pp. 243–270, April 1992.

[11] R. Andersson, "Dynamic sensing in a ping-pong playing robot,"

IEEE J. Robotics and Automation, vol. 5, no. 6, pp. 728–739, December 1989.

[12] D. Koller, K. Daniilidis, and H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *IJCV*, vol. 10, no. 3, pp. 257–281, June 1993.

[13] I. Reid and D. Murray, "Active tracking of foveated feature clusters using affine structure.," *IJCV*, vol. 18, no. 1, pp. 41–60, 1996.

[14] D. Coombs and C. Brown, "Real-time binocular smooth pursuit," *IJCV*, vol. 11, no. 2, pp. 147–164, October 1993.

[15] E. Schwartz, "Spatial mapping in the primate sensory projection : Analytic structure and relevance to perception," *Biological Cybernetics*, vol. 25, pp. 181–194, 1977.

[16] A. Bernardino and J. Santos-Victor, "Vergence control for robotic heads using log-polar images," in *Proc. IROS*, Osaka, Japan, November 1996, pp. 1264–1271.

[17] C. Capurro, F. Panerai, and G. Sandini, "Dynamic vergence," in *Proc. IROS*, Osaka, Japan, November 1996, pp. 1241–1247.

[18] J. Craig, *Introduction to Robotics: Mechanics and Control*, Addison-Wesley, 1986.

[19] A. Gelb, *Applied Optimal Estimation*, The MIT Press, 1994.

[20] C. Capurro, F. Panerai, and G. Sandini, "Dynamic vergence using log-polar images," *IJCV*, vol. 24, no. 1, pp. 79–94, August 1997.

[21] G. Salgian and D. Ballard, "Visual routines for vehicle control," in *The Confluence of Vision and Control*, D. Kriegman, G. Hager, and S. Morse, Eds. Springer Verlag, 1998.

[22] Y. Yeshurun and E. Schwartz, "Cepstral filtering on a columnar image architecture: A fast algorithm for binocular stereo segmentation," *IEEE Trans. PAMI*, vol. 11, no. 7, pp. 759–767, July 1989.

[23] D. Fleet, A. Jepson, and M. Jenkin, "Image matching using the windowed fourier phase," *CVGIP: Image Understanding*, vol. 53, no. 2, pp. 198–210, March 1991.

[24] A. Bernardino and J. Santos-Victor, "Sensor geometry for dynamic vergence," in *Workshop on Performance Characteristics of Vision Algorithms*, Cambridge, UK, April 1996.

[25] M. Pobuda and C. Erkelens, "The relation between absolute disparity and ocular vergence," *Biological Cybernetics*, vol. 68, pp. 221–228, 1993.

[26] B. Horn and B. Shunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[27] J. Santos-Victor, F. van Trigt, and J. Sentieiro, "Medusa - a stereo head for active vision," in *Proc. of the Int. Symposium on Intelligent Robotic Systems*, Grenoble, France, July 1994.

Alexandre Bernardino was born in Lisboa, Portugal, in 1971. He received the MSc degree in Electrical and Computer Engineering in 1997 from the Instituto Superior Técnico (Lisboa), and is working towards the achievement of the PhD degree. He is also a teaching assistant at the Instituto Superior Técnico and a research assistant at the Instituto de Sistemas e Robótica (Lisboa). His main research interests focus on robot vision, autonomous systems, and real-time control.

José Santos-Victor obtained the Licenciatura degree in Electrical and Computer Engineering from IST - Instituto Superior Técnico - Lisbon, Portugal, in 1988, the MSc degree in 1991 and the Ph.D in 1995 from the same institution, in the areas of Active Computer Vision and Robotics. He is an assistant professor at the Departement of Electrical and Computer Engineering of IST, and he is the scientific responsible for the participation of IST in various european and national research projects, in the same areas. His research interests are in the areas of Computer and Robot Vision, Intelligent Control Systems, particularly in the relationship between visual perception and the control of action, in (land and underwater) mobile robots.