1. EVALUATION OF THE ROBUSTNESS OF VISUAL BEHAVIORS THROUGH PERFORMANCE CHARACTERIZATION

João P. Barreto, Paulo Peixoto, Jorge Batista, and Helder Araujo
Institute of Systems and Robotics
Dept. of Electrical Engineering
University of Coimbra
Polo II - Pinhal de Marrocos
3030 Coimbra - PORTUGAL

Abstract

The robustness of visual behaviors implemented in an active vision system depends both on the vision algorithms and control structure. To improve robustness an evaluation of the performance of the several system algorithms must be done. Performance analysis can be done within the framework of control theory since notions such as stability and controllability can contribute to a better understanding of the algorithms and architectures weaknesses. In this paper we discuss the generation of reference test target trajectories and we characterize the performance of smooth pursuit and vergence. The responses to motion are used to accurately identify implementation problems and possible optimizations. The system evaluation leads to solutions to enhance the global performance and robustness.

1.1 Introduction

Robust visual control of motion depends on issues related both to vision processing and control. Robustness of a specific visual behavior is a function of the performance of vision and control algorithms as well as the overall architecture [1, 2, 3]. Performance characterization of both vision and control aspects should be performed within a common framework. This would enable a global view of the performance of a specific approach. For example, when dealing with the problem of uncertainties and coping with varying environments (which are difficult or impossible to model) one can, in principle, choose to use more complex vision algorithms and/or more robust control algorithms. Good decisions and choices can only be made if all the aspects can be characterized in a common framework. Control theory has a number of tools that enable a common global characterization of the performance in visual servoing and active vision systems [4]. Several different measures and concepts can be used to perform such common characterization.

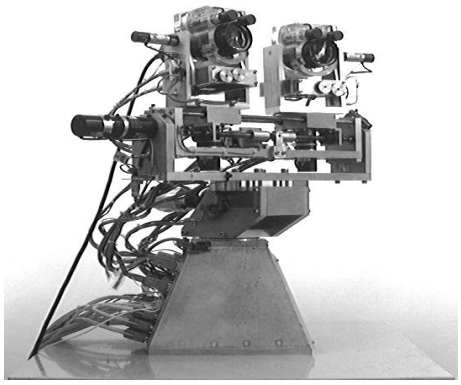

Figure 1.1: The MDOF binocular system

Many aspects related to visual servoing have been studied and several systems demonstrated [5, 6]. One of these aspects is the issue of system dynamics. System dynamics is essential to enable the performance optimization of the system [7, 8]. Other aspects are related to stability and the system latencies [9, 10, 11]. In [11] Corke shows that dynamic modeling and control design are very important for the improved performance of visual closed-loop systems. One of his main conclusions is that a feedforward type of control strategy is necessary to achieve high-performance visual servoing. Nonlinear aspects of system dynamics have also been addressed [12, 13]. In [12] Kelly discusses the nonlinear aspects of system dynamics and proves that the overall closed loop system composed by the full nonlinear robot dynamics and the controller is Lyapunov stable. In [13] Hong models the dynamics of a two-axis camera gimbal and also proves that a model reference adaptive controller is Lyapunov stable. In [14] Rizzi and Koditschek describe a system that takes into account the dynamical model of the target motion. They propose a novel triangulating state estimator and prove the convergence of the estimator. In [15, 16] the control performance of the Yorick head platform is also presented. In special it is considered the problem of dealing with the inherent delays and in particular with variable delays. Problems associated with overcoming system latencies are also discussed in [17, 18, 19].

1.2 Control of the MDOF Binocular Tracking System

The binocular MDOF robot head is a high-performance active vision system with a high number of degrees of freedom [20]. Real-time complex visual behaviors were implemented after careful kinematics modeling and adequate selection of basic visual routines[21, 22]. Binocular tracking of 3D motion was achieved by controlling neck pan/tilt and vergence.

In most cases visual servoing systems are analyzed as servo systems that use

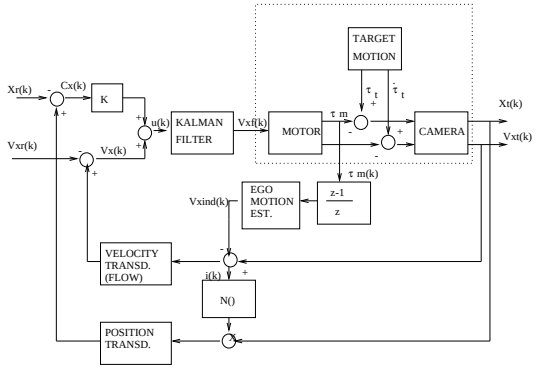

Figure 1.2: Monocular smooth pursuit block diagram. The dotted box encloses the analog components of the structure. Block $N(i(k))$ represents a non-linear function. $V_{xf}(k)$ is the command sent to the motor, obtained by filtering $u(k)$, the sum of the estimated velocity with the position error multiplied by a gain $K.V_{xind}(k)$ is the velocity induced in image by camera motion

vision as a sensor [23, 24]. Therefore the binocular tracking system should be considered as a servomechanism whose reference inputs are the target coordinates in space and whose outputs are the motor velocities and/or positions. In this case binocular vision is used to directly estimate target 3D motion parameters.

However in the case of this system, and as a result of both its mechanical complexity and its goal (tracking of targets with unknown dynamics), we decided to relate the system outputs with the data measured from the images. Thus this system can be considered as a regulator whose goal is to keep the target in a certain position in the image (usually its center). As a result of this framework target motion is dealt with as a perturbation. If the perturbation affects the target position and/or velocity in the image it has to be compensated for. The changes in the head geometry during the tracking process can be used to estimate the target 3D motion parameters.

1.2.1 Monocular Smooth Pursuit. Pan/Tilt Block Diagram

Each camera joint has two independent rotational degrees of freedom: pan and tilt. Even though pure rotation can not be guaranteed we model these degrees of freedom as purely rotational. A schematic for one of the these degrees of freedom is depicted in Fig 1.2 (both degrees of freedom are similar and decoupled). Notice that 2 inputs and 2 outputs are considered. Both position and velocity of the target in the image are to be controlled or regulated. Even though the two quantities are closely related, this formal distinction allows for a better evaluation of some aspects such as non-linearities and limitations in performance.

$$\begin{cases} i(k) = V_{xt}(k) - V_{xind}(k) \\ N(i(k)) = 1 \Longleftarrow i(k) \neq 0 \\ N(i(k)) = 0 \Longleftarrow i(k) = 0 \end{cases} \tag{1.1}$$

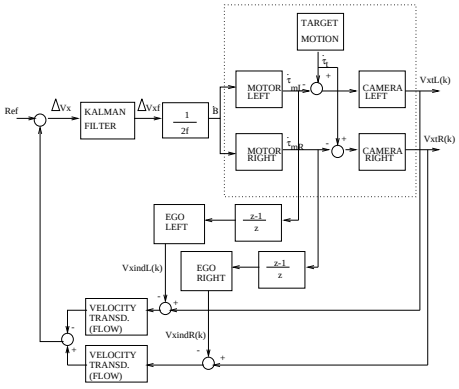

Figure 1.3: Vergence block diagram. Egomotion is estimated for each camera. After that target velocities in both left and right images are computed using differential flow. Estimated horizontal disparity (Δv_{xf}) is obtained by filtering the difference of measured velocities in both images

Considering that the motion computed in the image is caused by target motion and by camera motion, the computation of the target velocity requires that the effects of egomotion are compensated for. The egomotion is estimated based on the encoder readings and on the inverse kinematics. Once egomotion velocity ($V_{xind}(k)$) is compensated for, target velocity in the image plane is computed based on an affine model of optical flow. Target position is estimated as the average location of the set of points with non-zero optical flow in two consecutive frames (after egomotion having been compensated for). This way what is actually computed is the center of motion instead of target position. The estimated value will be zero whenever the object stops, for it is computed by using function $N(i(k))$ (equation 1.1) .

1.2.2 Vergence Block Diagram

In this binocular system, pan and tilt control align the cyclopean Z (forward-looking) axis with the target. Vergence control adjusts both camera positions so that both target images are projected in the corresponding image centers. Retinal flow disparity is used to achieve vergence control. Vergence angles for both cameras are equal and angular vergence velocity is computed in equation 1.2 where Δv_{xf} is the horizontal retinal motion disparity and f the focal length.[25]

$$\frac{\partial \beta}{\partial t} = \frac{\Delta v_{xf}}{2f} \tag{1.2}$$

A schematic for vergence control is depicted in Fig.1.3. Horizontal target motion disparity is regulated by controlling the vergence angle.

Both in smooth pursuit and vergence control, target motion acts as a perturbation that has to be compensated for. To study and characterize system regulation/control performance usual control test signals must be applied. Two problems

have to be considered:

- The accurate generation of perturbation signals;
- The generation of perturbation signals functionally defined, such as steps, ramps, parabolas and sinusoids;

1.3 Reference Trajectories Generation Using Synthetic Images

To characterize the system ability to compensate for the perturbations due to target motion, specific signals have to be generated. Instead of using real targets, we decided to use synthetic images so that the mathematical functions corresponding to reference trajectories could be accurately generated. These images are then used as inputs in the binocular active vision system. Given a predefined motion, captured frames will depend, not only on the target position, but also on the camera orientation. Due to the change on the system geometry as a result of its operation, images have to be generated on line to take into account the specific geometry at each time instant. Therefore at each time instant both target position and camera orientation have to be known in the same inertial coordinate system. The former is calculated using a specific motion model that enables the computation of any kind of motion in space. Camera orientation is computed by taking into account the motor encoders readings and the inverse kinematics. The inertial coordinate system origin is placed at optical center (monocular case) or at the origin of the cyclopean referential (binocular case).

To accurately describe the desired target motion in space the corresponding equations are used. Motion coordinates are converted into inertial Cartesian coordinates by applying the suitable transformation equations. Target coordinates in the inertial system are converted in camera coordinates. This transformation depends on motor positions that are known by reading the encoders. Perspective projection is assumed for image formation. These computations are performed at each frame time instant.

1.4 The reference trajectories equations.

To characterize control performance, target motion correspondent to a step, a ramp, a parabola and a sinusoid should be used to perturb the system.

1.4.1 Smooth Pursuit. Pan/Tilt Control System

Reference Trajectories Defined for the Actuators

Consider the perturbation at actuator/motor output. The reference trajectories are studied for both a rotary and a linear actuator.

In the former the actuator is a rotary motor and the camera undergoes a pure rotation around the Y (pan) and X (tilt) axis. Consider target motion equations defined in spherical coordinates (ρ, ϕ, θ), where ρ is the radius or depth, ϕ the

elevation angle and θ the horizontal angular displacement. The target angular position $\theta(t)$ at time t is given by one of:

$$\theta(t) = \begin{cases} Const \Longleftarrow t > 0 \\ 0 \Longleftarrow t = 0 \end{cases} \tag{1.3}$$

$$\theta(t) = \omega.t \tag{1.4}$$

$$\theta(t) = \frac{\gamma}{2}.t^2 \tag{1.5}$$

$$\theta(t) = A\sin(\omega.t) \tag{1.6}$$

Equations 1.3, 1.4, 1.5 and 1.6 describe a step, a ramp, a parabola and a sinusoid for the pan motor. For instance, if the target moves according to equation 1.4, the motor has to rotate with constant angular velocity ω to track the target. These definitions can be extended to the tilt motor by making $\theta = 0$ and varying ϕ according to equations 1.3 to 1.6.

Assume now a linear actuator and camera moving along the X axis. Cartesian equations 1.7 to 1.10 are the equivalent to spherical equations 1.3 to 1.16. In all cases the depth z_i is made constant.

$$x_i(t) = \begin{cases} Const \Longleftarrow t > 0 \\ 0 \Longleftarrow t = 0 \end{cases} \tag{1.7}$$

$$x_i(t) = v.t \tag{1.8}$$

$$x_i(t) = \frac{a}{2}.t^2 \tag{1.9}$$

$$x_i(t) = A\sin(v.t) \tag{1.10}$$

Reference Test Signals Defined in Image. Static Camera Situation

To relate the system outputs with the data measured from the images, control test signals must be generated in the image plane. Thus a step (in position) is an abrupt change of target position in image. A ramp/parabola (in position) occurs when the 3D target motion generates motion with constant velocity/acceleration in the image plane. And a sinusoid is generated whenever the image target position and velocity are described by sinusoidal functions of time (with a phase difference of 90 degrees).

$$x_{img} = f.\tan(\theta) \tag{1.11}$$

$$\frac{dx_{img}}{dt} = f.\frac{d\theta}{dt}.\frac{1}{\cos^2(\theta)} \tag{1.12}$$

$$\frac{d^2x_{img}}{dt^2} = f.\frac{d^2\theta}{dt^2}.\frac{1}{\cos^2(\theta)} + 2.f.(\frac{d\theta}{dt})^2.\frac{\tan(\theta)}{\cos^2(\theta)} \tag{1.13}$$

Consider that the camera is static. The observations depicted in Fig.1.4 agree with equations derived in 1.11, 1.12, 1.13 which relate angular position (θ) in space with target image coordinates $(x_{img}, y_{img}, z_{img})$ (f is the focal length and perspective projection is assumed). Notice in Fig.1.4 that, despite the inexistence of an angular acceleration, a residual acceleration can be observed in target image motion due to the second term of equation 1.13. Target motion described by spherical

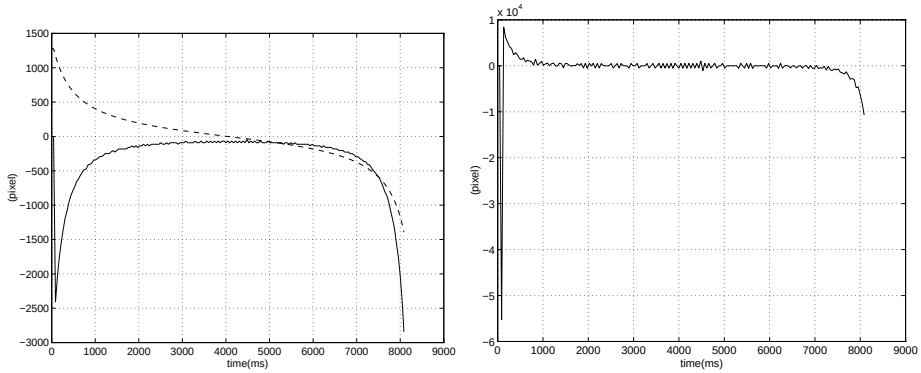

Figure 1.4: Motion projection in a static camera. The target moves along a circular path with constant angular velocity $\omega = 5(degrees)$ (equation 1.4). Left: Target position(- -) and velocity (-) in image. Right: Target acceleration(-) in image

equations 1.4 to 1.6, does not generate the desired perturbations in the image plane when the camera is static. Fig.1.4 shows that image motion distortion is significant when θ is above 50 degrees.

$$x_{img} = f.\frac{x_i}{z_i} \tag{1.14}$$

$$\frac{dx_{img}}{dt} = \frac{f}{z_i}.\frac{dx_i}{dt} - \frac{f.x_i}{z_i^2}.\frac{dz_i}{dt} \tag{1.15}$$

$$\frac{d^2x_{img}}{dt^2} = \frac{f}{z_i}.\frac{d^2x_i}{dt^2} - \frac{f.x_i}{z_i^2}.\frac{d^2z_i}{dt^2} - \frac{2.f}{z_i^2}.\frac{dx_i}{dt}.\frac{dz_i}{dt} - \frac{2.f.x_i}{z_i^3}.(\frac{dz_i}{dt})^2 \tag{1.16}$$

However, in the case of a target moving in a rectilinear trajectory parallel to the image plane (constant depth), the standard perturbations are obtained. Whenever images are obtained with a static camera, linear motion described by equations 1.7 to 1.10 is adequate to generate the standard control test signals. This conclusion is confirmed by equations 1.14, 1.15 and 1.16 (z_i remains constant) that relate image coordinates with Cartesian motion coordinates. This result is still true if camera moves along a linear path.

Reference Test Signals Defined in Image. Camera Undergoing Pure Rotation

The MDOF binocular system uses rotary eye joints. Thus, considering the monocular situation, the camera moves along a circular trajectory. We assume camera rotation around X_c axis (pan), the target moving along a circular/spherical path (see Fig.1.5) and perspective projection modeling image formation.

$$x_{img} = f.\tan(\theta - \alpha_p) \tag{1.17}$$

$$\frac{dx_{img}}{dt} = f.\frac{d\theta}{dt}.\frac{1}{\cos^2(\theta - \alpha_p)} - f.\frac{d\alpha_p}{dt}.\frac{1}{\cos^2(\theta - \alpha_p)} \tag{1.18}$$

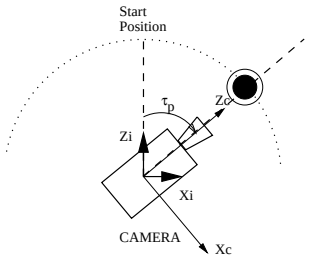

Figure 1.5: Tracking using the pan motor. α_p is the motor angular position and $\theta(t)$ the target angular position

$$\frac{d^2x_{img}}{dt^2} = f.(\frac{d^2\theta}{dt^2} - \frac{d^2\alpha_p}{dt^2}).\frac{1}{\cos^2(\theta-\alpha_p)} + 2f.(\frac{d\theta}{dt} - \frac{d\alpha_p}{dt})^2.\frac{\tan(\theta-\alpha_p)}{\cos^2(\theta-\alpha_p)} \quad (1.19)$$

Target position ($x_{img}(t)$) in the image is dependent both on the camera angular position ($\alpha_p(t)$) and target angular position ($\theta(t)$) (equation 1.17). To compute the target velocity in the image, equation 1.18 is derived by differentiating equation 1.17. Notice that the target and camera angular positions are time dependent. By differentiating equation 1.18 the expression for target acceleration in image is obtained (equation 1.19).

As can be noticed in these equations, motion in the image is caused both by target motion and camera motion. For a perfect tracking situation the former is compensated by the latter and no motion is detected in the image. Whenever perfect tracking does not happen there will be image motion as a result of tracking error. Therefore, the objective of tracking is to move the camera in such a way that egomotion compensates for the motion induced in the image by the target. From this point of view the system perturbation will be the motion induced by the target.

$$\omega_i = f.\frac{d\theta}{dt}.\frac{1}{\cos^2(\theta-\alpha_p)} \quad (1.20)$$

$$\gamma_i.t = f.\frac{d\theta}{dt}.\frac{1}{\cos^2(\theta-\alpha_p)} \quad (1.21)$$

$$A\omega_i\cos(\omega_i.t) = f.\frac{d\theta}{dt}.\frac{1}{\cos^2(\theta-\alpha_p)} \quad (1.22)$$

The reference trajectories that generate a perturbation in ramp, parabola and sinusoid are derived by solving the differential equations 1.20, 1.21 and 1.22 in order to $\theta(t)$ (ω_i, γ_i and A are the desired induced velocity, acceleration and amplitude). The difficulty is that the reference trajectories ($\theta(t)$) will depend on the system reaction to the perturbation ($\alpha_p(t)$). That is due to the fact that the image is not only function of target position in space, but also of camera orientation. Thus to induce a constant velocity in image during operation, target angular velocity must be computed at each frame time instant as a function of the the tracking error.

Consider that perfect tracking is going to occur. The tracking error will be null and $\alpha_p(t) = \theta(t)$. With this assumption the solutions of differential equations 1.20 to 1.21 are given by equations 1.4 to 1.6 (making $\omega = \frac{\omega_i}{f}$ and $\gamma = \frac{\gamma_i}{f}$). These are the

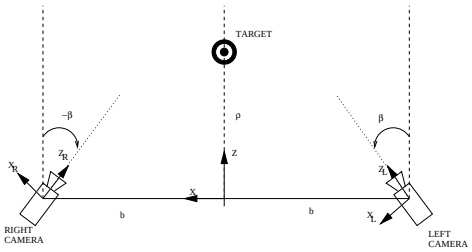

Figure 1.6: Top view of binocular system. The distance between the cameras is $2b$ and symmetric vergence is assumed. $\rho(t)$ is the target Z coordinate.

reference trajectories that we are going to use to characterize the system. It is true that for instance, trajectory of eq.1.4 (the ramp) only induces a constant velocity in image if tracking error is null (small velocity variation will occur otherwise). However it is independent of the system reaction and the generated perturbation allows the evaluation of system ability to recover from tracking errors.

1.4.2 The Vergence Control System

Taking into account the considerations of last section, the reference trajectories for vergence control characterization of the binocular system depicted in Fig. 1.6 are now discussed. The distance between the cameras is $2b$ and symmetric vergence is assumed. The Z coordinate of the target position (in the cyclopean coordinate frame) is ρ.

$$\Delta x_{img} = 2f.\frac{-\rho.\sin(\beta) + b.\cos(\beta)}{\rho.\cos(\beta) + b.\sin(\beta)} \tag{1.23}$$

$$\beta = \arctan(\frac{b}{\rho}) \tag{1.24}$$

Vergence control is achieved using retinal disparity. The differences of target position and velocity in the images of both cameras are the system stimuli. The position retinal disparity is calculated in equation 1.23. Perfect tracking is achieved when β is computed by equation 1.24. In this case $\Delta x_{img} = 0$.

$$\Delta V x_{img} = -\frac{2fb}{\sqrt{\rho^2 + b^2}}.\frac{d\rho}{dt} \tag{1.25}$$

Deriving equation 1.23 the expression for velocity retinal disparity is obtained. Suppressing the egomotion effect (considering $\frac{d\beta}{dt} = 0$), the stimulus generated by target motion is computed in equation 1.25 assuming a perfect tracking situation.

$$2fb.\frac{d\rho}{dt} + v.\rho^2 = -v.b^2 \tag{1.26}$$

$$a = -\frac{2fb}{\rho^2 + b^2}.\frac{d^2\rho}{dt^2} + \rho.\frac{4fb}{(\rho^2 + b^2)^2}.(\frac{d\rho}{dt})^2 \tag{1.27}$$

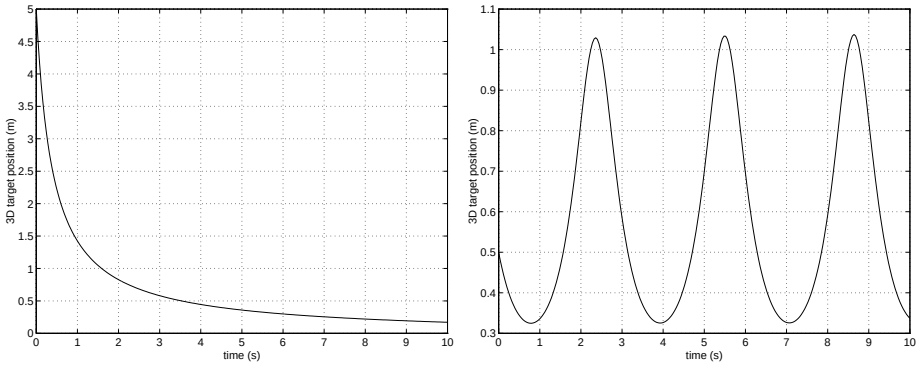

Figure 1.7: Left: Ramp perturbation. Target motion to generate a constant disparity of 1 pixel/frame ($\rho(0) = 5$(m)).Right: Sinusoidal Perturbation. Target motion that generates a sinusoidal velocity disparity in images($A = 2$(pixel), $\omega = 2$(rad/s) and $\rho(0) = 1$(m))

$$2fb.\frac{d\rho}{dt} + Aw\cos(wt).\rho^2 = -Aw\cos(wt).b^2 \tag{1.28}$$

The target motion equation $\rho(t)$ that generates a motion corresponding to a ramp in image target position (constant velocity disparity v) is determined solving equation 1.26 derived from 1.25. For a parabola (constant acceleration disparity a) equation 1.27 must be solved. In the case of a sinusoidal stimulus, the relevant target motion equation $\rho(t)$ can be computed by solving equation 1.28. Test signals obtained by solving differential equations 1.26 and 1.28 are depicted in Fig.1.7. Notice that to induce a constant velocity disparity in the images the 3D target velocity increases with depth. This is due to the perspective projection.

1.5 System Response to Motion

In this section we analyze the system ability to compensate for perturbations due to target motion.

1.5.1 Smooth Pursuit. Pan/Tilt Control Algorithm

As shown previously spherical/circular target motion must be used to generate the standard control test signals. Pan and tilt control algorithms are identical except for some of the parameter values. Therefore we will consider only the pan axis.

Step Response

A step in position is applied to the system. Fig. 1.8 shows the evolution of the target position (X_t) in the image. An overshoot of about 10% occurs. The regulation is done with a steady state error of about 1.5 pixels. These observations are in agreement with the observed positional servo-mechanical performance. This

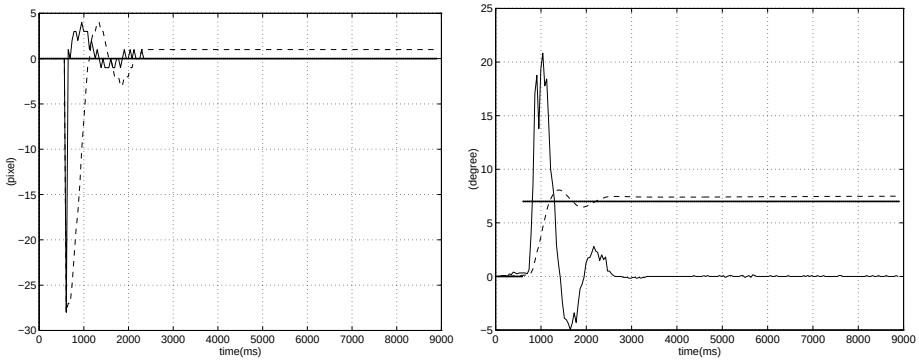

Figure 1.8: Left: Regulation performance. Target position (- -) and velocity (-) in the image. Right: Servo-mechanical performance. Target angular position (.), motor position (- -) and velocity (-)

is a typical second order step response of a type 0 system. In experiments done with smaller amplitude steps the system fully compensates for target motion. In these situations the regulation error is 0 and we have a type 1 system. The type of response depends on the step amplitude which clearly indicates a non-linear behavior. One of the main reasons for the non-linear behavior is the way position feedback is performed. After compensating for egomotion, target position is estimated as the average location of the set of points with non-zero optical flow in two consecutive frames. Thus the center of motion is calculated instead of the target position. If the target stops, any displacement detected in the image is due to camera motion. In that case target velocity ($V_{xt}(k)$) is equal to the induced velocity ($V_{xind}(k)$) and the position estimate C_x will be 0. Therefore target position would only be estimated at the step transition time instant. Only with egomotion as a pure rotation would this occur. In practice sampling and misalignment errors between the rotation axis and the center of projection introduce small errors.

A step in position corresponds to an impulse perturbation in velocity. Fig 1.8 shows the system ability to cancel the perturbation. Note that only the first peak velocity is due to real target motion.

Ramp Response

Fig.1.9 exhibits the ramp response for a velocity of 10 deg/s (1.5 pixel/frame). The target moves about 6 pixels off the center of image before the system starts to compensate for it. It clearly presents an initial inertia where the action of the Kalman filter plays a major role. The Kalman filtering limits the effect of measurement errors and allows smooth motion without oscillations.

Considering the motor performance we have a type 1 position response to a ramp and a second order type 1 velocity response to a step. The position measurement error

$$e(k) = X_t(k) - C_x(k) \tag{1.29}$$

will be directly proportional to the speed of motion.

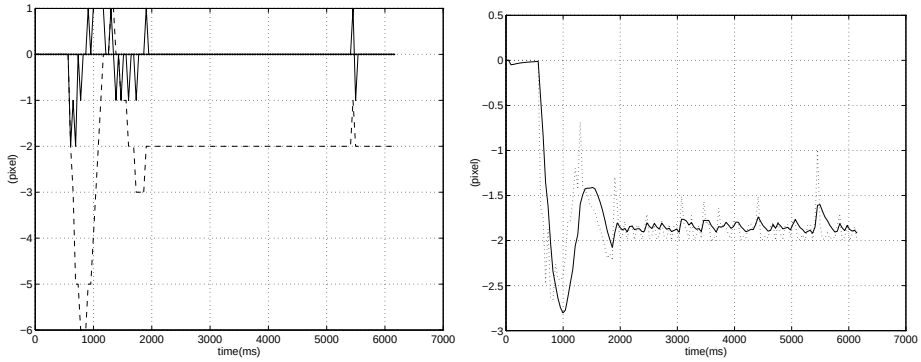

Figure 1.9: Left:Regulation performance.Target position (- -) and velocity (-) in the image. Right: Kalman filtering. Kalman input $u(k)$ (.) and output $V_{xf}(k)$(-)

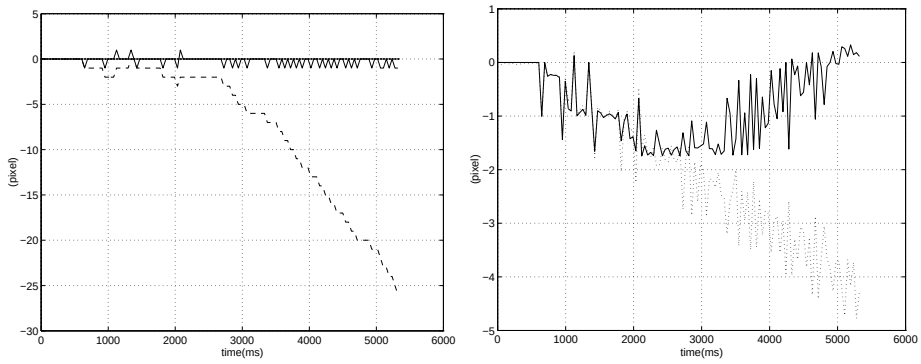

Figure 1.10: Left: Regulation performance. Target position (- -) and velocity (-) in the image). Right: Velocity estimation. Target velocity (.) and flow (-)

The algorithm for velocity estimation using optical flow only performs well for small velocities (up to 2 pixels/frame). For higher motion speeds the flow is clearly underestimated. This represents a severe limitation that is partially compensated for by the proportional position error component on the motor commands. Experiments were performed that enabled us to conclude that the system only follows motions with constant velocities of up to $20\,\mathrm{deg}/s$.

Parabola Response

The perturbation is generated by a target moving around the camera with a constant angular acceleration of $5\,\mathrm{deg}/s^2$ and an initial velocity of $1\,\mathrm{deg}/s$. When the velocity increases beyond certain values flow underestimation bounds the global performance of the system. The system becomes unable to follow the object and compensate for its velocity. As a consequence the object image is increasingly off the image center and the error in position increases.

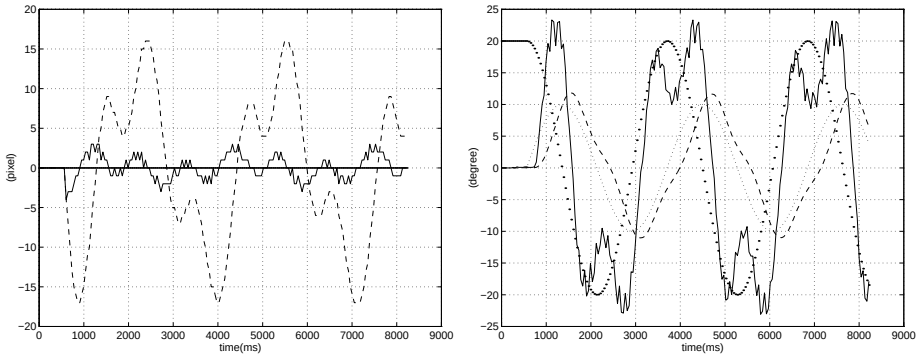

Figure 1.11: Left: Regulation Performance–Target position(- -) and velocity (-) in the image. Right: Servo-mechanical performance in position. Motor position (- -) and velocity (-). Target position (:) and velocity (.)

Sinusoidal Response

System reaction to a sinusoidal perturbation of angular velocity $2rad/s$ is studied. Fig. 1.11 shows target position X_t and velocity V_x in the image. Non-linear distortions, mainly caused by velocity underestimation, can be observed. Notice the phase lag and the gain in position motor response in Fig. 1.11.

1.5.2 Vergence Control System

Fig.1.12 depicts the vergence performance in compensating for a perturbation in step and ramp. The test signals are obtained as explained in section 4.2. These new observations confirm that visual processing is limiting the global performance. In the next section we describe some solutions that were explored to improve position and velocity estimation in image.

1.6 Improvements in the Visual Processing

1.6.1 Target Position Estimation in Image

Target position estimation in the image is fundamental to keep the position regulation error small and to reduce the effects of occasional erroneous velocity prediction.

$$C_x[k] = C_x[k-1] + V_{xind}[k] \tag{1.30}$$

Some problems in position estimation, that interfere with global system performance, were detected. The center of motion is estimated only when the target induces motion in the image. When no target motion is detected (after egomotion compensation) it can be considered that the target did not move. Thus the new position estimate should be equal to the previous estimate compensated for the induced displacement due to camera motion(equation 1.30). Another problem is that the center of motion is computed instead of the target position. The position

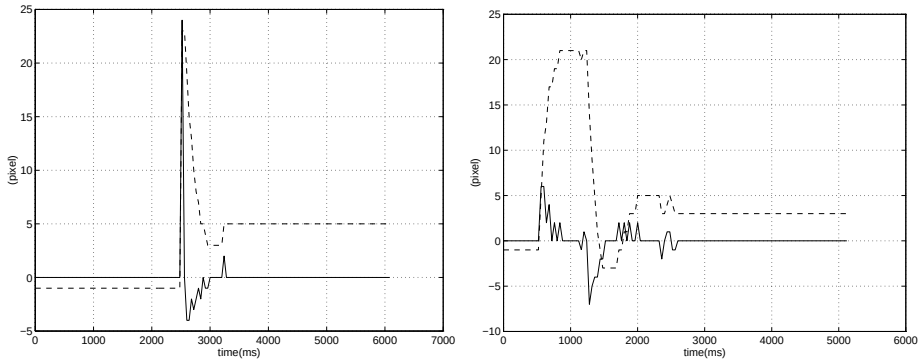

Figure 1.12: Left: Vergence regulation performance for a step perturbation. Right: Vergence regulation performance for a ramp perturbation.Target position(- -) and velocity(-) disparity in the image.

estimate is computed as the average location of the set of points with non-zero optical flow in two consecutive frames. If this set is restricted to the points of the most recently acquired frame that have non-zero spatial derivatives, the average location will be near the target position. The improvements in position estimation can be observed in Fig.1.13.

1.6.2 Target Velocity Estimation in the Image

To estimate the target velocity in the image, the brightness gradient is calculated in all pixels of the acquired frame. Considering the flow constraint equation and assuming that all points in the image move with the same velocity, the velocity vector (u, v) is estimated using a least squares method.

$$I_x.u + I_y.v + I_t = 0 \tag{1.31}$$

The flow constraint equation 1.31 is valid for a continuous brightness function (under the assumption of brightness constancy). However the actual function $I(x, y, t)$ is discrete in time and space. Aliasing problems in partial derivatives computation can compromise a correct velocity estimation. When the target image moves very slowly high spatial resolution is needed in order to correctly compute the derivatives Ix and Iy and to estimate the velocity. On the other hand, if the the target image moves fast, there are high frequencies in time and I_t must be computed with shorter sampling periods. However the sampling frequency is limited to $25Hz$. One solution to estimate high target velocities is to decrease the spatial resolution. The drawback of this approach is that high frequencies are lost, and small target movements will no longer be detected. We tested several methods to increase the range of target velocities in the image that the system is able to estimate. The method that gave the best results is next presented.

The method starts by building a pyramid of images with different resolutions. Two levels are considered: the lower with a $64x64$ image, and the higher with a $32x32$ resolution. The flow is computed at the high level using a $2x2$ mask. The

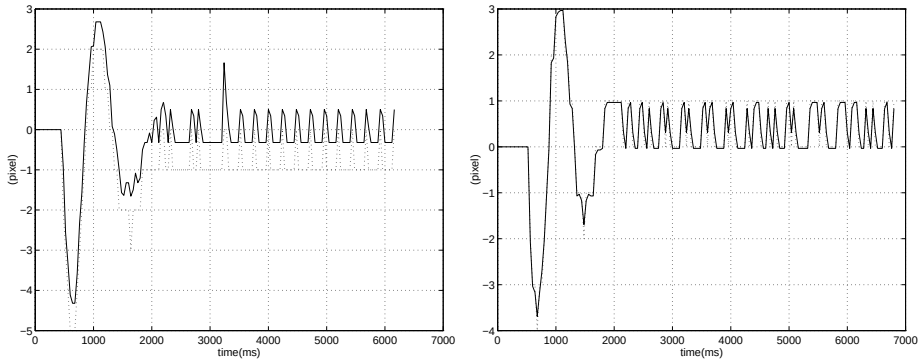

Figure 1.13: Response for a ramp perturbation of 1.5pixel/frame (10deg/s). Left: Position estimation using the original method. Target position (:) and target position estimation(-). Right: Position estimation using the improved method. Target position (:) and target position estimation (-)

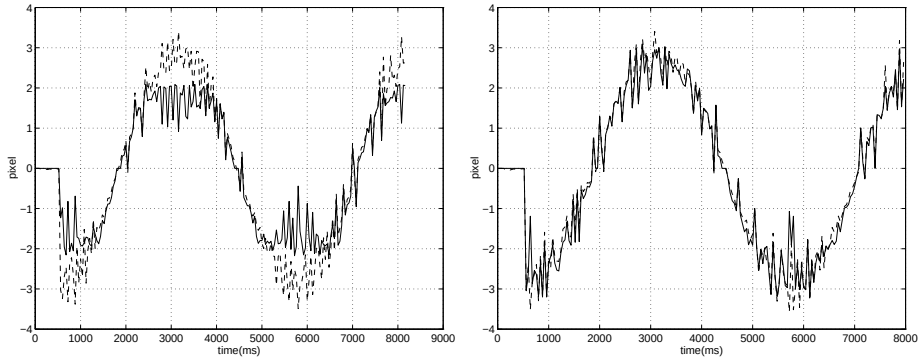

Figure 1.14: Response to a sinusoidal perturbation. Left: Velocity estimation using the original method. Right: Velocity estimation using the new method with a two-level pyramid . The target velocity in the image (–) and the estimated value(-). Both methods perform a correct estimation of velocity

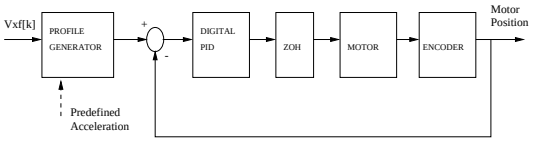

Figure 1.15: Motor control loop. A PID is used to control motor position. The sampling frequency in the closed-loop is 1KHz. A profile generator allows to control the motor in velocity

value of the computed flow (V_{high}) is used to define the size of the mask that is employed to estimate target velocity at the 64x64 level (V_{low}). The mask can have a size of 2,3 or 4 pixels depending on the value of V_{high} at each time instant. Notice that the final velocity is always given by V_{low}. The rule that controls the mask size is based on intervals between predefined threshold values. To each interval corresponds a certain mask size that it is chosen if the value of V_{high} belongs to that interval. The threshold values are determined experimentally. The range of velocity estimation can be increased by using more levels in the pyramid.

1.7 Motor Performance and Global System Behavior

The latency of the actuators is an important issue to achieve high speed tracking. In the MDOF robot head actuation is done using DC motors with harmonic drives controlled by Precision Microcontrol DCX boards. The implemented control loop is depicted in Fig.1.15. Motor position is controlled using a classic closed-loop configuration with a digital PID controller running at 1KHz. For velocity control the reference inputs (in position) are computed by a profile generator. This device integrates the velocity commands sent by the user process. Acceleration and deceleration values can be configured to assure more or less smoothness in velocity changes. Due to the fact that each board controls up to six axis, the user process can only read the encoders and send commands for 6ms time intervals.

The PID of the inner position loop must be "tight" in order to minimize the position error and guarantee small velocity rise times. Fig.1.16 exhibits the motor response to successive velocity commands. The rise time is about 1 frame time instant. The overshoot is not constant (non-linear behavior) and the global performance decreases for abrupt changes in input. Therefore, during operation, abrupt changes in velocity commands must be avoided to maximize motor performance.

A decrease in processing time from 38ms to 8ms was achieved by upgrading the processor. The effects in global performance can be observed in Fig.1.16. In the first implementation, the frame was captured and the actuating command was sent just before the following frame grabbing. Considering a rise time of 1 frame time interval, the motor only reached the velocity reference 80ms after the capture of the corresponding frame. By decreasing the image processing time the reaction delay is reduced to almost half the value and the system becomes more responsive. When the second frame is grabbed, the camera is approximately moving with the

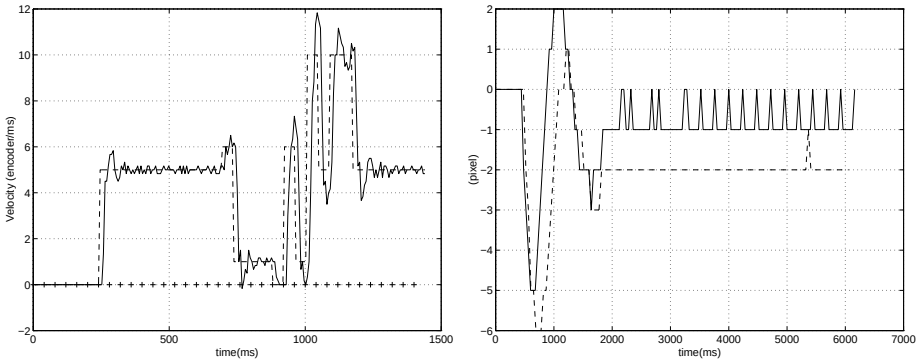

Figure 1.16: Left: Motor response to sudden changes in velocity. The velocity command (–) and the motor velocity response measured for a sampling interval of 6ms (-). The dashes along zero axis mark the frame time instants (40ms). Right: Regulation performance for a ramp perturbation of 1.5pixel/frame (10deg/s). Processing time of 38ms (–) and 8ms(-).

target velocity estimated in the previous iteration.

1.8 Improvements in Global Performance. Experimental Results

The performance and robustness of the pan/tilt smooth pursuit improved by decreasing the computation time and enhacing the visual processing (position and velocity estimation)(see Fig.1.17). Fig.1.18 shows that vergence control performance improved as well.

1.9 Summary and Conclusions

In this paper we address the problem of improving the performance and robustness of tracking performed by a binocular active vision system. In order to enable the evaluation of the robustness of both vision and control algorithms in a common framework, we decided to use a methodology inspired by control techniques. The different subsystems were characterized by their responses to test inputs. Due to the specific features of an active vision system several questions related to the definition of system reference inputs had to be addressed. As a result we propose and justify a methodology for the definition and generation of such reference inputs.

System identification of some modules of the system, including the visual processing routines (which required their linearization), was also done. The results enabled us to identify elements that should be improved. Specifically, in this paper, we described the improvements in the visual processing algorithms. These improvements enable the system to track targets in a much larger range of depths.

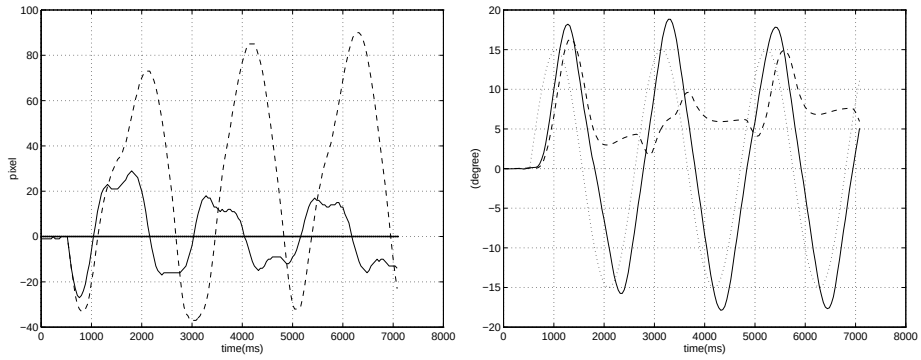

Figure 1.17: Pan/Tilt Control System. Response to a sinusoidal perturbation. Left: Regulation performance. Target position in the image for the original (- -) and improved implementation(-). Right: Servo-mechanical performance. Target angular position(.), motor position in the original (- -) and improved (-) algorithm.

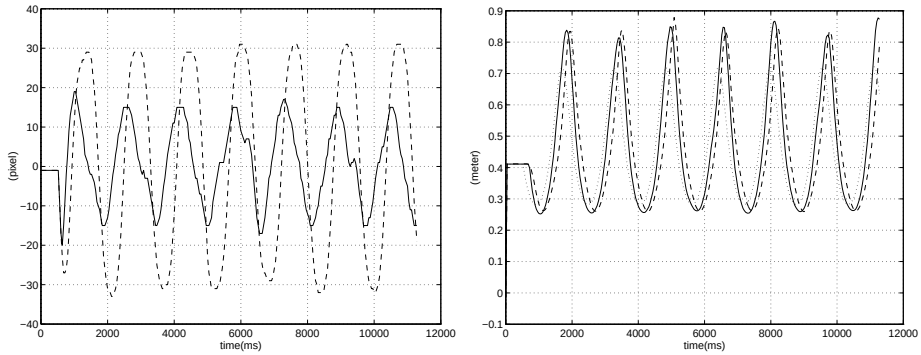

Figure 1.18: Vergence Control System. Response to a sinusoidal perturbation. Left: Regulation performance. Target position disparity in image for the original (- -) and improved implementation(-). Right: Servo-mechanical performance. Target depth position(.), vergence depth position in the original (- -) and improved (-) algorithm.

Bibliography

[1] J. L. Crowley and editors H. I. Christensen, *Vision as a process,* Springer-Verlag, 1995.

[2] J. L. Crowley, J. M. Bedrune, M. Bekker, and M. Schneider, "Integration and control of reactive visual processes," *Third European Conference in Computer Vision*, Vol. 2, pp. 47-58, May 1994.

[3] H. I. Christensen, J. Horstmann, and T. Rasmussen, "A control theoretic approach to active vision," *Asian Conference on Computer Vision*, pp. 201-210, December 1995.

[4] Y. Aloimonos, I. Weiss, and A. Bandyopadhay, "Active vision", *International Journal of Computer Vision*, Vol. 1(4), pp. 333-356, January 1988.

[5] G. Hager and S. Hutchinson, "Special section on vision-based control of robot manipulators," *IEEE Trans. on Robot. and Automat.*, Vol. 12(5), October 1996.

[6] R. Horaud and F. Chaumette, editors, *Workshop on New Trends in Image-Based Robot Servoing*, September 1997.

[7] E. Dickmanns, "Vehicles capable of dynamic vision," *in Proc. of the 15th International Conference on Artificail Intelligence*, August 1997.

[8] E. Dickmanns, "An approach to robust dynamic vision," *in Proc. of the IEEE Workshop on Robust Vision for Vision-Based Control of Motion*, May 1998.

[9] P. I. Corke and M. C. Good, "Dynamic effects in visual closed-loop systems," *IEEE Trans. on Robotics and Automation*, Vol. 12(5), pp. 671-683, October 1996.

[10] P. I. Corke, "Visual control of robot manipulators-a review," In K. Hashimoto, editor, *Visual Servoing*, pp. 1-31. World Scientific, New York, 1993.

[11] P. I. Corke, "Visual Control of Robots: High-Peformance Visual Servoing," *Mechatronics,* John Wiley, 1996.

[12] R. Kelly, "Robust asymptotically stable visual servoing of planar robots," *IEEE Trans. on Robot. and Automat*, Vol. 12(5), pp. 697-713, October 1996.

[13] W. Hong, *Robotic catching and manipulation using active vision,* Master's thesis, MIT, September 1995.

[14] A. Rizzi and D. E. Koditschek, "An active visual estimator for dexterous manipulation," *IEEE Trans. on Robot. and Automat*, Vol. 12(5), pp. 697-713, October 1996.

[15] P. Sharkey, D. Murray, S. Vandevelde, I. Reid, and P. Mclauchlan, "A modular head/eye platform for real-time reactive vision," *Mechatronics*, Vol. 3(4), pp. 517-535, 1993.

[16] P. Sharkey and D. Murray, "Delays versus performance of visually guided systems," *IEE Proc.–Control Theory Appl.*, Vol. 143(5), pp. 436-447, September 1996.

[17] C. Brown, "Gaze controls with interactions and delays," *IEEE Trans. on Systems, Man and Cybern.*, Vol. 20(2), pp. 518-527, 1990.

[18] D. Coombs and C. Brown, "Real-time binocular smooth pursuit," *International Journal of Computer Vision*, Vol. 11(2), pp. 147-164, October 1993.

[19] J. Dias, C. Paredes, I. Fonseca, H. Araujo, J. Batista, and A. Almeida, "Simulating pursuit with machines: Experiments with robots and artificial vision," *IEEE Trans. on Robot. and Automat.*, Vol. 14(1), pp. 1-18, 1998.

[20] H. Araujo J. Batista, J. Dias and A. Almeida, "The isr multi-degrees-of-freedom active vision robot head: design and calibration," *M2VIP'95–Second International Conference on Mechatronics and Machine Vision in Practice*, Hong–Kong, September 1995.

[21] J. Batista, P. Peixoto, and H. Araújo, "Real-time visual behaviors with a binocular active vision system," *ICRA97–IEEE Int. Conf. on Robotics and Automation*, New Mexico, USA, April 1997.

[22] J. Batista, P. Peixoto, and H. Araújo, "Real-time vergence and binocular gaze control," *IROS97–IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Grenoble, France, September 1997.

[23] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. on Robot. and Automat.*, Vol. 8(3), pp. 313-326, June 1992.

[24] P. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Trans. on Robot. and Automat.*, Vol. 9(2), pp. 152-165, 1993.

[25] J. Batista, P. Peixoto, and H. Araujo, " Real-time active visual surveillance by integrating peripheral motion detection with foveated tracking," *Proc. of the IEEE Workshop on Visual Surveillance*, pp 18-25, 1998.