Improving 3D Active Visual Tracking

João Barreto, Paulo Peixoto, Jorge Batista, and Helder Araujo

Institute of Systems and Robotics, Dept. Electrical Engineering, University of Coimbra, 3030 Coimbra, Portugal,
jpbar,peixoto,batista,helder@isr.uc.pt,
WWW home page: http://www.isr.uc.pt/

Abstract. Tracking in 3D with an active vision system depends on the performance of both motor control and vision algorithms. Tracking is performed based on different visual behaviors, namely smooth pursuit and vergence control. A major issue in a system performing tracking is its robustness to partial occlusion of the target as well as its robustness to sudden changes of target trajectory. Another important issue is the reconstruction of the 3D trajectory of the target. These issues can only be dealt with if the performance of the algorithms is evaluated. The evaluation of such performances enable the identification of the limits and weaknesses in the system behavior. In this paper we describe the results of the analysis of a binocular tracking system. To perform the evaluation a control framework was used both for the vision algorithms and for the servo-mechanical system. Due to the geometry changes in an active vision system, the problem of defining and generating system reference inputs has specific features. In this paper we analyze this problem, proposing and justifying a methodology for the definition and generation of such reference inputs. As a result several algorithms were improved and the global performance of the system was also enhanced. This paper proposes a methodology for such an analysis (and resulting enhancements) based on techniques from control theory.

1 Introduction

Tracking of moving 3D targets using vision can be performed either with passive or active systems. Active systems facilitate tracking and the reconstruction of 3D trajectories if specific geometric configurations of the system are used [1, 2]. In the case of active systems robust 3D tracking depends on issues related both to vision processing and control[3, 4]. Robustness of a specific visual behavior is a function of the performance of vision and control algorithms as well as the overall architecture[5]. The evaluation of the global performance of both vision and control aspects should be done within a common framework. 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 [6].Improvements in performance as well as the identification of less robust elements in the system strongly benefit from a common approach[7].

Many aspects related to visual servoing and tracking have been studied and several systems demonstrated [8, 9]. One of these aspects is the issue of system dynamics. The study of system dynamics is essential to enable performance optimization [10, 11]. Other aspects are related to stability and the system latencies [12, 13]. In [13] 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 [14, 15]. In [14] 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 [15] Hong models the dynamics of a two-axis camera gimbal and also proves that a model reference adaptive controller is Lyapunov stable. In [16] 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 [17, 18] the control performance of the Yorick head platform is also presented. They pay careful attention to the problem of dealing with image processing inherent delays and in particular with variable delays. Problems associated with overcoming system latencies are also discussed in [19, 20]. Optimality in visual servoing was studied by Rivlin in [21]. Recently, in the GRASP laboratory, the performance of an active vision system has also been studied [22].

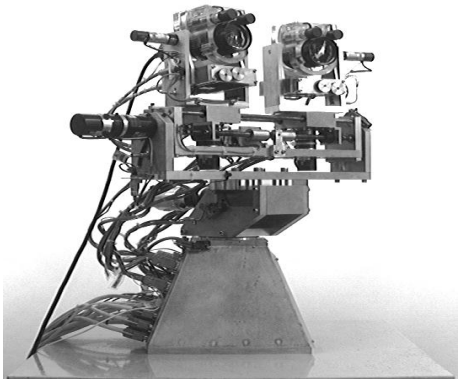

Fig. 1. The MDOF binocular system

2 Control of the MDOF Binocular Tracking System

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

2.1 Monocular Smooth Pursuit

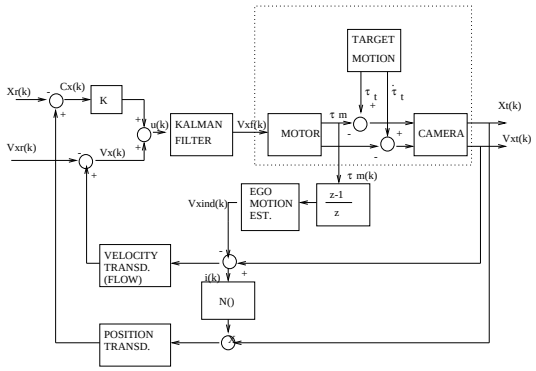

Fig. 2. Monocular smooth pursuit block diagram. The dotted box encloses the analog components of the structure. Block $N(i(k))$ represents a non-linear function. $V_{xf}(k)$ is the command sent to the motor, obtained by filtering $u(k)$, the sum of the estimated velocity with the position error multiplied by a gain $K.V_{xind}(k)$ is the velocity induced in image by camera motion

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 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.

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

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) .

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 2 where Δv_{xf} is the horizontal retinal motion disparity and f the focal length.[25]

$$\frac{\partial \beta}{\partial t} = \frac{\Delta v_{xf}}{2f} \quad (2)$$

A schematic for vergence control is depicted in Fig.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;

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

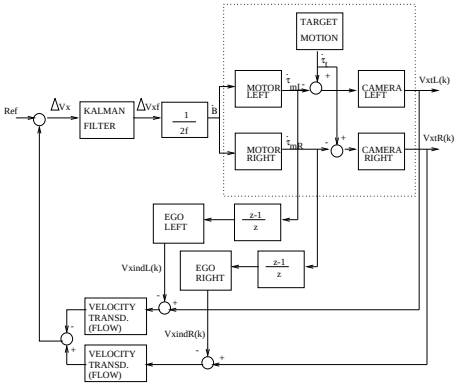

Fig. 3. Vergence block diagram. Egomotion is estimated for each camera. After that target velocities in both left and right images are computed using differential flow. Estimated horizontal disparity (Δv_{xf}) is obtained by filtering the difference of measured velocities in both images

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[26]. 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.

4 Perturbation Signals. 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.

4.1 The Monocular Tracking 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:

$$\theta(t) = \begin{cases} Const \Longleftarrow t > 0 \\ 0 \Longleftarrow t = 0 \end{cases} \tag{3}$$

$$\theta(t) = \omega.t \tag{4}$$

$$\theta(t) = \frac{\gamma}{2}.t^2 \tag{5}$$

$$\theta(t) = A\sin(\omega.t) \tag{6}$$

Equations 3, 4, 5 and 6 describe a step, a ramp, a parabola and a sinusoid for the pan motor. For instance, if the target moves according to equation 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 3 to 6.

Assume now a linear actuator and camera moving along the X axis. Cartesian equations 7 to 10 are the equivalent to spherical equations 3 to 6. In all cases the depth z_i is made constant.

$$x_i(t) = \begin{cases} Const \Longleftarrow t > 0 \\ 0 \Longleftarrow t = 0 \end{cases} \tag{7}$$

$$x_i(t) = v.t \tag{8}$$

$$x_i(t) = \frac{a}{2}.t^2 \tag{9}$$

$$x_i(t) = A\sin(v.t) \tag{10}$$

Reference Test Signals Defined in Image 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).

Assume the camera is static. Target motion described by equations 7 to 10 generates the standard control test signals in image. This result is still true if camera moves along a linear path.

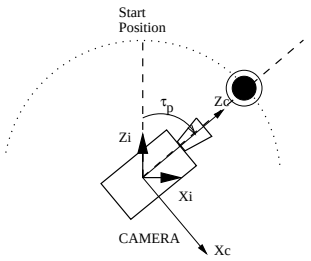

Fig. 4. Monocular tracking. $\alpha_p(t)$ is motor angular position and $\theta(t)$ the target angular position

$$\omega_i = f.\frac{d\theta}{dt}.\frac{1}{\cos^2(\theta - \alpha_p)} \tag{11}$$

$$\gamma_i.t = f.\frac{d\theta}{dt}.\frac{1}{\cos^2(\theta - \alpha_p)} \tag{12}$$

$$A\omega_i \cos(\omega_i.t) = f.\frac{d\theta}{dt}.\frac{1}{\cos^2(\theta - \alpha_p)} \tag{13}$$

However MDOF system eye cameras perform rotations. For this situation the reference trajectories that generate a perturbation in ramp, parabola and sinusoid are derived by solving the differential equations 11, 12 and 13 in order to $\theta(t)$ (ω_i, γ_i and A are the desired induced velocity, acceleration and amplitude in image plane).[27] The difficulty is that the reference trajectories ($\theta(t)$) will depend on the system reaction to the perturbation ($\alpha_p(t)$). Thus to induce a constant velocity in image during operation, target angular velocity must be computed at each frame time instant in function of the the tracking error.

Consider the case of perfect tracking. The tracking error will be null and $\alpha_p(t) = \theta(t)$. With this assumption the solutions of differential equations 11 to 13 are given by equations 4 to 6 (making $\omega = \frac{\omega_i}{f}$ and $\gamma = \frac{\gamma_i}{f}$). These are the reference trajectories that we use to characterize the system. While is true that, for instance, trajectory of eq.4 (the ramp) only induces a constant velocity in image if tracking error is null (small velocity variation will occur otherwise), the test signal becomes independent of the system reaction and the generated perturbation allows the evaluation of system ability to recover from tracking errors.

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. 5 are presented.

$$2fb.\frac{d\rho}{dt} + v.\rho^2 = -v.b^2 \tag{14}$$

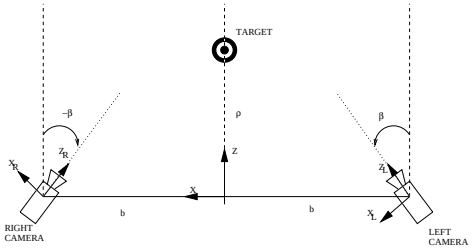

Fig. 5. Top view of binocular system. The distance between the cameras is $2b$ and symmetric vergence is assumed. $\rho(t)$ is the target Z coordinate.

$$a = -\frac{2fb}{\rho^2 + b^2}.\frac{d^2\rho}{dt^2} + \rho.\frac{4fb}{(\rho^2 + b^2)^2}.(\frac{d\rho}{dt})^2 \tag{15}$$

$$2fb.\frac{d\rho}{dt} + Aw\cos(wt).\rho^2 = -Aw\cos(wt).b^2 \tag{16}$$

Assume perfect tracking. 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 14. For a parabola (constant acceleration disparity a) equation 15 must be solved. In the case of a sinusoidal stimulus, the relevant target motion equation $\rho(t)$ can be computed by solving equation 16.[27] Test signals obtained solving diferential equations 14 and 16 are depicted in

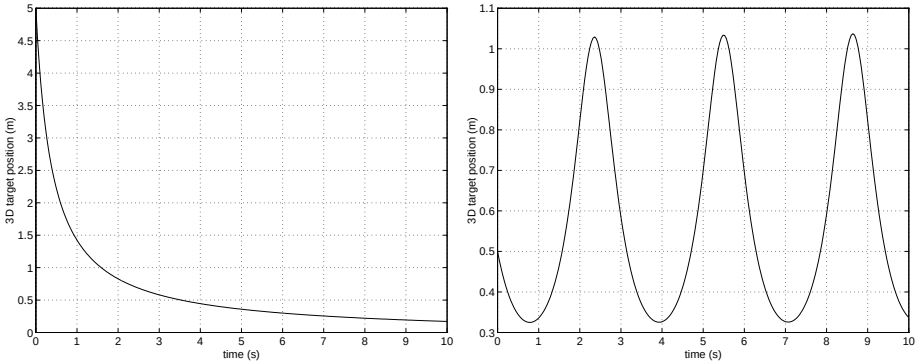

Fig. 6. Left: Ramp perturbation. Target motion to generate a constant disparity of 1 pixel/frame ($\rho(0) = 5$(m)).Right: Sinusoidal Perturbation. Target motion that generates a sinusoidal velocity disparity in images($A = 2$(pixel), $\omega = 2$(rad/s) and $\rho(0) = 1$(m))

Fig.6. Notice that to induce a constant velocity disparity in images the 3D target velocity increases with depth. This is due to the perspective projection.

5 System Response to Motion

In this section we analyze the system ability to compensate for perturbations due to target motion. As demonstrated spherical/circular target motion must be used to generate the standard control test signals. Pan and tilt control algorithms are identical except for some parameter values. Because that we only consider the pan axis.

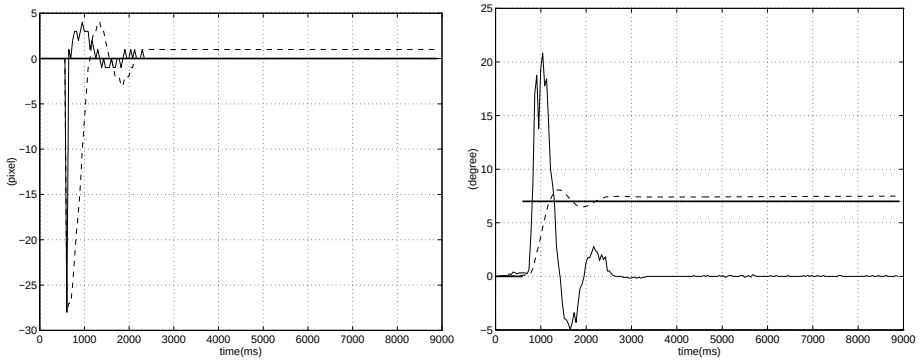

Fig. 7. Left: Regulation performance. Target position (- -) and velocity (-) in image. Right: Servo-mechanical performance. Target angular position (.), motor position (- -) and velocity (-)

Step Response A step in position is applied to the system. Fig. 7 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 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 camera motion. In that case target velocity ($V_{xt}(k)$) is equal to 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 7 shows the ability of the system to cancel the perturbation. Note that only the first peak velocity is due to real target motion.

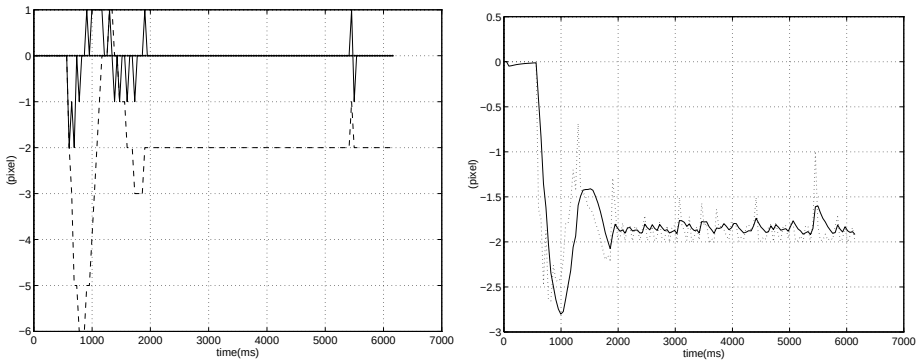

Fig. 8. Left:Regulation performance.Target position (- -) and velocity (-) in the image. Right: Kalman filtering. Kalman input $u(k)$ (.) and output $V_{xf}(k)$(-)

Ramp Response Fig.8 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

$$e(k) = X_t(k) - C_x(k) \tag{17}$$

will be directly proportional to the speed of motion.

The algorithm for velocity estimation using optical flow only performs well for small velocities (up to 2 pixels/frame). For higher speeds of motion 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 ve-

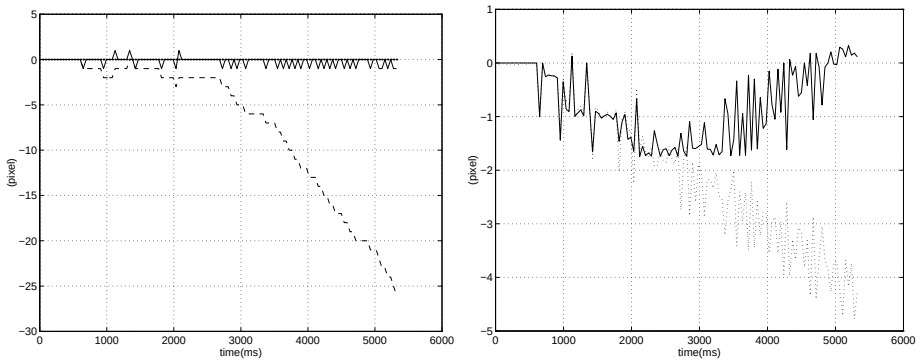

Fig. 9. Left: Regulation performance. Target position (- -) and velocity (-) on image). Right: Velocity estimation. Target velocity (.) and flow (-)

locity 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 center of the image and the error in position increases.

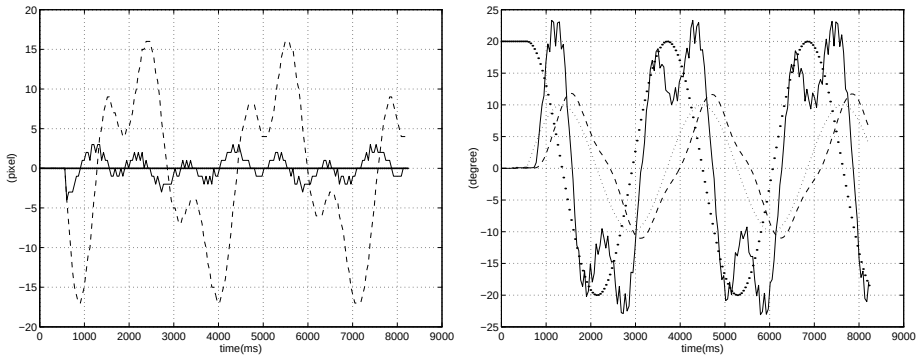

Fig. 10. Left: Regulation Performance–Target position(- -) and velocity (-) in the image. Right: Servo-mechanical performance in position. Motor position (- -) and velocity (-). Target position (:) and velocity (.)

Sinusoidal Response System reaction to a sinusoidal perturbation of angular velocity $2rad/s$ is studied. Fig. 10 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. 10.

6 Motor Performance and Its Implication in Global System Behavior

During system response analysis non-linear behaviors were observed. Despite that, linear approximations can be considered for certain ranges of operation. Therefore we have estimated the transfer functions of some of the sub-systems depicted in Fig. 2 using system identification techniques. [26].

$$M(z) = 0.09z^{-3}.\frac{1 + 0.38z^{-1}}{(1 - z^{-1})(1 - 0.61z^{-1} + 0.11z^{-2})} \tag{18}$$

Equation 18 gives the obtained motor transfer function. $M(z)$ relates the computed velocity command ($V_{xf}(k)$) in $pixel/sec$, with motor angular position in degrees. The pole in $z = 1$ is due to the integration needed for velocity-position conversion. A pure delay of 3 frames (120ms) was observed. There is a consid-

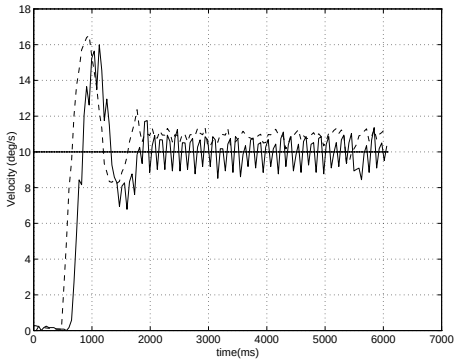

Fig. 11. Motor input and output for a ramp perturbation of 1.5pixel/frame (10deg/s). Velocity command sent to DCX board controler (–) and motor velocity measured by reading the encoders (-). Sampling period of 40ms (each frame time instant)

erable inertia from motor input to output (see Fig.11). Such a delay certainly interferes with global system performance. In the MDOF robot head actuation is done using DC motors with harmonic drive controlled by Precision Microcontrol DCX boards. The implemented control loop is depicted in Fig.12. 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

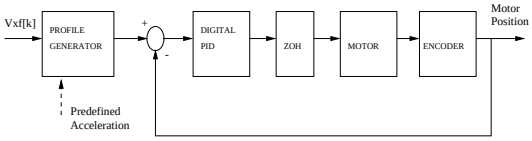

Fig. 12. Motor control loop. A PID is used to control motor position. The sampling frequency in the closed-loop is 1KHz. A profile generator allows to control the motor in velocity

commands sent by the user process. Acceleration and deacceleration 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.

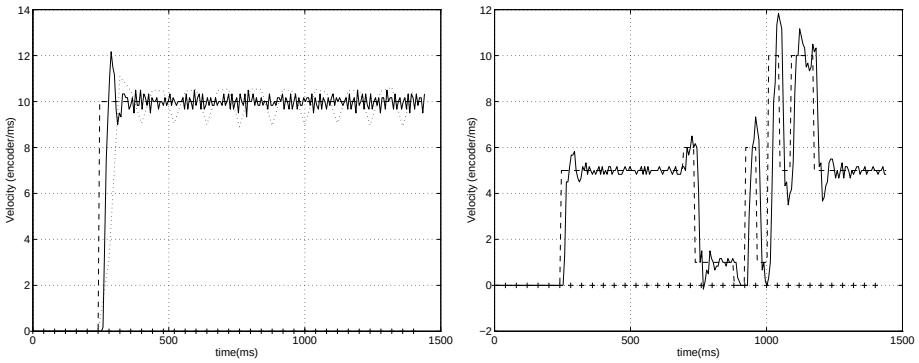

Fig. 13. Left: Step response for the velocity controlled motor. The velocity command (–) and the motor velocity output for a sampling interval of 6ms (-) and 40ms (:). Right: Motor response to sudden changes in velocity. The velocity command (–) and the motor velocity response measured for a sampling interval of 6ms (-). In both figures, the dashes along zero axis mark the frame time instants (40ms)

$$V_\alpha[kT] = \frac{1}{T}\int_{(k-1)T}^{kT} v_\alpha(t).dt \tag{19}$$

As shown in Fig.2, at each frame time instant (40ms), the velocity command $V_{xf}[k]$ is sent to motor and camera position is read at the encoders. These readings are used to estimate motor velocity. Assuming that $v_\alpha(t)$ is the continuous time motor velocity, the measured value $V_\alpha[kT]$ will be given by equation 19, where T is the sampling period. Therefore, at each sampling instant, we are estimating the average velocity along the period instead of the velocity at that

instant. Fig.13 shows the same step response measured for two different sampling rates. we can therefore conclude that $T = 40ms$ is too large to correctly estimate instantaneous velocity. We can also conclude that the delay of 3 frames as well as the ripple observed in Fig.11 results from different sampling rates. As a matter of fact such delay as well as ripple do not occur.

Notice that the delay observed in transfer function $M(z)$ is correct (velocity input and position output). It means that the motor takes about 3 frame time instants to get to the same position that it would reach in 1 frame time instant for an ideal velocity response. from the standpoint of position regulation this value is important. It interferes with the tracking steady state error. However, to achieve high performance visual tracking the system must perform velocity regulation. The rise time of motor velocity response is the crucial parameter to achieve responsive behaviors.

The DCX board turns a position controlled axis in a velocity controlled axis using an additional integrator (the profile generator). The PID of the inner position loop must be "tight" in order to minimize the position error and guarantee small velocity rise times. Fig.13 exhibits the motor response for 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. So, during operation, abrupt changes in velocity commands must be avoided to maximize motor performance. A decrease in processing time

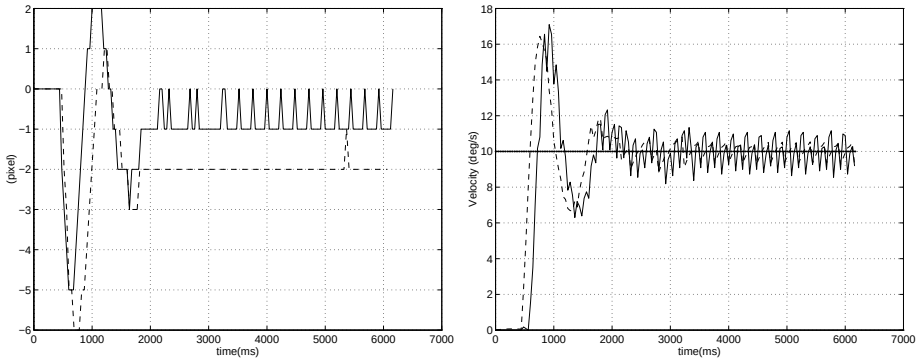

Fig. 14. Response for a ramp perturbation of 1.5pixel/frame (10deg/s).Left: Regulation performance. Processing time of 38ms (–) and 8ms(-).Right: Velocity command sent to DCX board controller (–) and motor velocity measured by reading the encoders (-). In both figures the sampling interval is 40ms

from 38ms to 9ms was achieved by improving the used hardware (processor upgrade). The effects in global performance can be observed in Fig.14. 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

instant, 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 target velocity estimated in the previous iteration.

7 Improvements in the Visual Processing

The characterization of the active vision system allows the identification of several aspects that limit the global performance. In this section improvements in visual processing are discussed as a way to overcome some of the problems.

7.1 Target Position Estimation in Image

The input velocity sent to the motor is obtained by filtering the sum of the estimated target velocity with the estimated target position multiplied by a gain K(equation 20). This is a simple control law that will probably be changed in future developments. However, the position component is always fundamental to keep the position regulation error small and to reduce the effects of occasional velocity misprediction.

$$u(k) = V_x(k) + K.C_x(k) \tag{20}$$

$$C_x[k] = C_x[k-1] + V_{xind}[k] \tag{21}$$

Some problems in position estimation, that interfere with global system perfor-

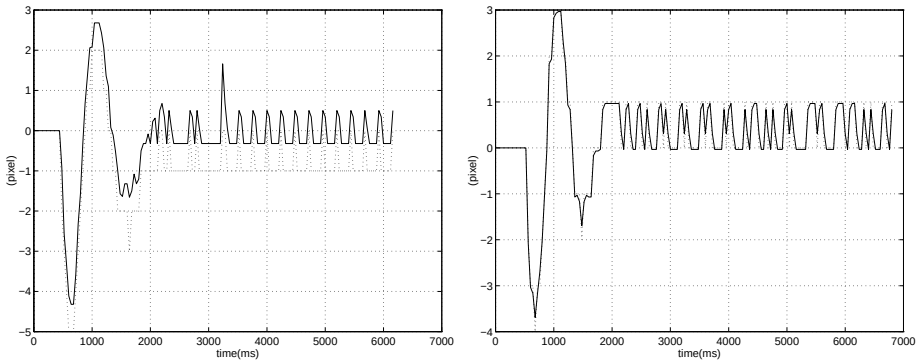

Fig. 15. Response for a ramp perturbation of 1.5pixel/frame (10deg/s). Left: Position estimation using the original method. Target position (:) and target position estimation(-). Right: Position estimation using the improved method. Target position (:) and target position estimation (-)

mance, were detected. The center of motion is estimated only when the target

induces motion in image. When no target motion is detected (after egomotion compensation) it can be assumed that the target did not move. Thus the new position estimate should be equal to the previous estimate compensated for the induced velocity due to camera motion(equation 21). Another problem is that the center of motion is computed instead of the target position. The position 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 last grabbed frame that have non-zero brightness partial derivatives with respect to X and Y, the average location will be near the target position. The improvements in position estimation can be observed in Fig.15. The improvements on

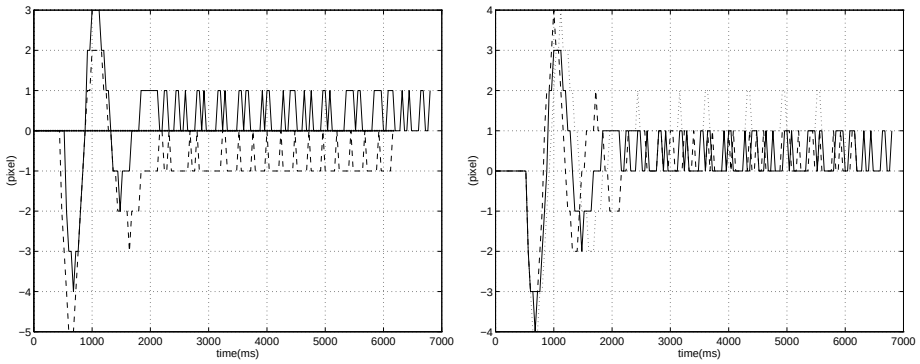

Fig. 16. Response for a ramp perturbation of 1.5pixel/frame (10deg/s). Left: Regulation performance. Original (–) and improved (-) position estimation method. Right: Regulation performance. Improved position estimation method for a K=0.2 (:), 0.3 (-) and 0.4 (–)

global system performance can be observed in Fig.16. The selected value for the gain K was 0.3. This value decreases the time of reaction and the position error without leading to oscillatory behaviors.

7.2 Target Velocity Estimation in Image

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

$$I_x.u + I_y.v + I_t = 0 \tag{22}$$

The flow constraint 22 is true for a continuous brightness function. However our brightness 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 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 for small 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 tried two methods to increase the range of target velocities in image that the systems is able to estimate.

Method 1 Consider that the image is grabbed by the system with half resolution. Computation of flow with a $2x2$ mask would allow the estimation of velocities up to 4 pixels/frame. Notice that an estimated velocity of 2 pixels/frame corresponds to a velocity of 4 pixels/frame in the original image. Thus, by lowering image resolution, the system is able to compute higher target displacements using the same flow algorithm. Lower resolution frames can be obtained by subsampling original images after a low-pass filtering.

This method starts by building a pyramid of images with different resolutions. For now only two levels are considered: the lower with a $64x64$ image, and the higher with a $32x32$ resolution. Flow is simultaneously computed in both levels using the same 2x2 mask. Theoretically, velocities below the 2 pixel/frame are well estimated in the low pyramid level (V_{low}). Higher displacements (between 2 to 4 pixels/frame) are better evaluated at the higher pyramid level (V_{high}). At each frame time instant the algorithm must decide which estimated velocity (V_{low} or V_{high}) is nearest to real target velocity in image.

$$\sum_{i=1}^{N} (I_x^i.u + I_y^i.v + I_t^i)^2 = 0 \tag{23}$$

Consider N data points where brightness gradient is evaluated. The velocity (u, v) is computed as the vector that minimizes the quadratic error of 23. Most of times the "fitting" is not perfect and each data point has a residue associated with it. The mean residue of the N points can be used as a measurement of the estimation process performance. Our algorithm chooses the velocity estimation with the smaller mean residue.

Method 2 As in method 1 a similar two-level pyramid is computed. The flow is computed at the high level using a $2x2$ mask. The result of this operation (V_{high}) controls the size of the mask that is used to estimate target velocity in the 64x64 level (V_{low}). The mask can have the size of 2,3 or 4 pixels depending on the value of V_{high} at each time instant. Notice that in this approach the final velocity is always given by V_{low}. The decision is not about which level has the best velocity estimation, but about changing the mask size for flow computation in the low level frame.

The law that controls 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. For a two level pyramid two threshold values are needed. The threshold values are determined experimentally.

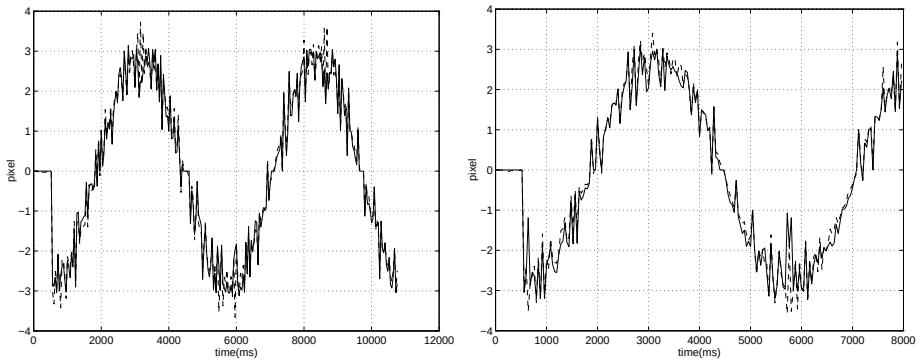

Fig. 17. Response for a sinusoidal perturbation. Right: Velocity estimation using method 1. Left: Velocity estimation using method 2. The target velocity in image (–) and the estimated value(-). Both methods perform a correct estimation of velocity

Experimental Results The original implementation is unable to estimate velocities above 2pixel/frame (see Fig18). With the new methods the system becomes able to estimate velocities up to 4pixel/frame.

The improvements in system performance can be observed in Fig18. In both methods the range of estimated velocities can be increased by using more levels in the pyramid. In method 2 the choice of the threshold values is critical for a good performance. Method 2 has the advantage of decoupling the velocity estimation in X and Y. For instance, consider that target velocity in the image is very high in X direction (horizontal) and small in Y direction (vertical). With method 1 it is not possible to have a good velocity estimate in both directions. If V_{high} is chosen then vertical velocity estimation will be affected by a great error; if V_{low} is chosen the horizontal velocity will be underestimated. Method 2 deals with this case by computing the flow in the $64x64$ image with a rectangular mask $2x4$.

8 Summary and Conclusions

In this paper we address the problem of improving the performance of tracking performed by a binocular active vision system. In order to enable the evaluation

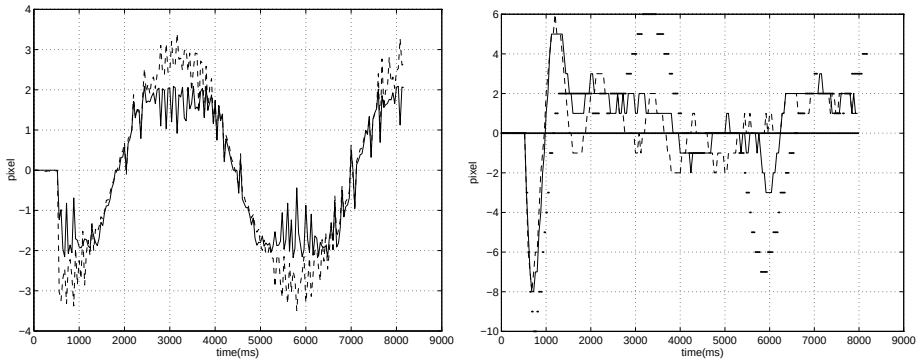

Fig. 18. Response for a sinusoidal perturbation. Left:Velocity estimation using the original method. The target velocity in image (–) and the estimated value(-). The systems only estimates velocities up to 2pixel/frame. Right: Regulation Performance. The target position in image for the original method (.), method 1(–) and method 2(-)

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.

References

1. Ruzena Bajcsy. Active perception vs. passive perception. *Third Workshop on Computer Vision: Representatin and Control*, pages 55–59, October 1985.
2. Y. Aloimonos, I. Weiss, and A. Bandyopadhay. Active vision. *International Journal of Computer Vision*, 1(4):333–356, January 1988.
3. H. I. Christensen, J. Horstmann, and T. Rasmussen. A control theoretic approach to active vision. *Asian Conference on Computer Vision*, pages 201–210, December 1995.
4. J. L. Crowley, J. M. Bedrune, M. Bekker, and M. Schneider. Integration and control of reactive visula processes. *Third European Conference in Computer Vision*, 2:47–58, May 1994.
5. J. O. Eklundh, K. Pahlavan, and T. Uhlin. The kth head-eye system. In *Vision as a Process*, pages 237–259, 1995.

6. T. Vieville. A few steps towards 3d active vision. *Springer-Verlag*, 1997.
7. A. Bernardino and J. Santos-Victor. Sensor geometry for dynamic vergence: Characterization and performance analysis. *Workshop on Performance Characterization of Vision Algorithms*, pages 55–59, 1996.
8. G. Hager and S. Hutchinson. Special section on vision-based control of robot manipulators. *IEEE Trans. on Robot. and Automat.*, 12(5), October 1996.
9. R. Horaud and F. Chaumette, editors. *Workshop on New Trends in Image-Based Robot Servoing*, September 1997.
10. E. Dickmanns. Vehicles capable of dynamic vision. *in Proc. of the 15th International Conference on Artificail Intelligence*, August 1997.
11. 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.
12. P. I. Corke and M. C. Good. Dynamic effects in visual closed-loop systems. *IEEE Trans. on Robotics and Automation*, 12(5):671–683, October 1996.
13. P. I. Corke. *Visual Control of Robots: High-Peformance Visual Servoing*. Mechatronics. John Wiley, 1996.
14. R. Kelly. Robust asymptotically stable visual servoing of planar robots. *IEEE Trans. on Robot. and Automat*, 12(5):697–713, October 1996.
15. W. Hong. Robotic catching and manipulation using active vision. Master's thesis, MIT, September 1995.
16. A. Rizzi and D. E. Koditschek. An active visual estimator for dexterous manipulation. *IEEE Trans. on Robot. and Automat*, 12(5):697–713, October 1996.
17. P. Sharkey, D. Murray, S. Vandevelde, I. Reid, and P. Mclauchlan. A modular head/eye platform for real-time reactive vision. *Mechatronics*, 3(4):517–535, 1993.
18. P. Sharkey and D. Murray. Delays versus performance of visually guided systems. *IEE Proc.–Control Theory Appl.*, 143(5):436–447, September 1996.
19. C. Brown. Gaze controls with interactions and delays. *IEEE Trans. on Systems, Man and Cybern.*, 20(2):518–527, 1990.
20. D. Coombs and C. Brown. Real-time binocular smooth pursuit. *International Journal of Computer Vision*, 11(2):147–164, October 1993.
21. H. Rotstein and E. Rivlin. Optimal servoing for active foveated vision. *in Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 177–182, June 1996.
22. Ulf M. Cahn von Seelen. Performance evaluation of an active vision system. Master's thesis, University of Pennsylvania, Philadelphia, USA, May 1997.
23. B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robot. and Automat.*, 8(3):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.*, 9(2):152–165, 1993.
25. J. Batista, P. Peixoto, and H. Araujo. Real-time active visual surveillance by integrating peripheral motion detection with foveated tracking. In *Proc. of the IEEE Workshop on Visual Surveillance*, pages 18–25, 1998.
26. J. Barreto, P. Peixoto, J. batista, and H. Araujo. Evaluation of the robustness of visual behaviors through performance characterization. *in Proc. of the IEEE Workshop on Robust Vision for Vision-Based Control of Motion*, 1998.
27. J. Barreto, P. Peixoto, J. Batista, and H. Araujo. Performance characterization of visula behaviors in an active vision system. *In 6th International Symposium on Intelligent Robotic Systems*, pages 309–318, July 1998.