To appear in the journal REAL TIME IMAGING, 1997

Real Time Tracking of Moving Objects with an Active Camera

K. Daniilidis, Ch. Krauss, M. Hansen, G. Sommer
Computer Science Institute, Christian-Albrechts University Kiel
Preusserstr. 1-9, 24105 Kiel, Germany
email:kd@informatik.uni-kiel.de

Abstract

This article is concerned with the design and implementation of a system for real time monocular tracking of a moving object using the two degrees of freedom of a camera platform. Figure-ground segregation is based on motion without making any a priori assumptions about the object form. Using only the first spatiotemporal image derivatives subtraction of the normal optical flow induced by camera motion yields the object image motion. Closed-loop control is achieved by combining a stationary Kalman estimator with an optimal Linear Quadratic Regulator. The implementation on a pipeline architecture enables a servo rate of 25 Hz. We study the effects of time-recursive filtering and fixed-point arithmetic in image processing and we test the performance of the control algorithm on controlled motion of objects.

1 Introduction

Traditional computer vision methodology regarded the visual system as a passive observer whose goal is the recovery of a complete description of the world. This approach led to systems unable to interact in a fast and stable way with a dynamically changing environment. Several variations of a new paradigm appearing under the names *active, attentive, purposive, behavior-based, animate, qualitative* vision were introduced in the last decade in order to overcome the efficiency and stability caveats of conventional computer vision systems. Common principle of the new theories is the behavior dependent selectivity in the way that visual data are acquired and processed. To cite one of the first definitions [1]: "Active Sensing can be stated as a problem of controlling strategies applied to the data acquisition process which will depend on the current state of the data interpretation and the goal or the task of the process".

Selection involves the ability to control the mechanical and optical degrees of freedom during image acquisition. Already in the early steps of active vision it was proven that controlling the degrees of freedom simplifies many reconstruction problems [2]. Selection encompasses the processing of the retinal stimuli at varying resolution what we call space variant sensing [3]. This means the ability to process only critical regions in detail while the rest of the field of view is coarsely analyzed. Last and most important, selection means the choice of the signal representation appropriate for a specific task to be accomplished taking also into account the physiology of the observer [4, Introduction]. Brown [5] resumes that a "a selective system should depending on the task decide which information to gather, which operators to use at which resolution, and where to apply them".

The subject of this paper is the accomplishment of one of the fundamental capabilities of an active visual system, that of pursuing a moving object. Since the moving object is detected at the beginning our system encompasses also the capability of saccadic eye movements. Here the only cue for the "where to look next" problem is motion. It is the first step towards a repertoire of oculomotor behaviors which will run in parallel. These involve fixating a stationary point or stabilizing the entire field of view if the observer is moving as well as binocular vergence movements. We will first describe the usefulness of pursuing a moving object.

The most evident reason for object pursuing is the limited field of view available by CCD cameras. The two degrees of freedom of panning and tilting enable keeping a moving object of interest in view for a longer time interval. Even if we had a sensor with 180 degrees field of view it would not be computationally possible to process every part of the field of view in the same detail. We would be enforced to apply foveal sensing, hence we should move the camera in order to keep the object inside the fovea. As already was proved in [6] and [7] tracking facilitates the estimation of the heading direction by reducing the number of unknowns restricting the position of the focus of expansion. It allows the use of an object-centered coordinate system and the simpler model of scaled orthographic projection. Object pursuing is necessary in cooperation with vergence control to keep the disparity inside an interval facilitating, thus, binocular fusion and a relative depth map.

As almost every visual system is engaged in a behavior of an animal or a robot that involves action vision becomes coupled with feedback control in order to enable a closed-loop between perception and action. Such a cycle is also the task of pursuing a moving object with an active camera described here. The most crucial matter is the accomplishment of this task in real time given the limited resources of our architecture. Under

these conditions, Marr's conception of an implementation step succeeding the algorithmic stage becomes obsolete. Here, the choice of the low-level signal processing depends on the given pipeline-architecture: we use two-dimensional non-separable FIR kernels for spatial filtering because our pipeline machine includes such a dedicated module but we apply recursive filtering in time. Normal flow can be computed inside the pipeline image processor, therefore it is the basis of our motion detection algorithm. This does not mean that we apply ad hoc techniques. We believe that real time design should be based on the detailed performance study of algorithms satisfying the real time constraints. Hardware components become faster so that mathematically sound image processing methods can replace the Sobel operator for spatial derivatives or the time differences for temporal ones.

The contribution of the work presented here can be summarized as following:

- A system that can detect and track moving objects independent of form and motion in 25 Hz.
- A study for the choice of the individual algorithms - which we do not claim to have invented – regarding
  - fixed-point arithmetic accuracy
  - space and time complexity of the filters given a specific architecture
  - and performance of the closed-loop control algorithm.
- Experiments with several object forms and motions.

Concerning biological findings eye movements of primates are classified in saccades, smooth pursuit, optokinetic reflex, vestibulo-ocular reflex, and vergence movements [8]. Optokinetic and vestibular reflexes try to stabilize the entire field of view in order to eliminate motion blur. Saccades are fast ballistic movements which direct gaze to a new focus of attention whereas smooth pursuit are slow closed-loop movements that keep an object fixated. Fixation enables the analysis of objects in the high-resolution foveal region. Vergence movements minimize the stereo disparity facilitating thus binocular fusion. Tracking of objects consists of both smooth pursuit movements that move the eye at the same velocity as the target and corrective saccades that shift a lost target again into the fovea. In this sense, our system accomplishes tracking with corrective saccades which, however, are smoothed by the closed-loop control.

Potential applications for the presented system are in the field of surveillance in indoor or outdoor scenes. The advantages are not only in the motion detection but mainly in the capability of keeping an intruder inside the field of view. Another application is in automatic video recording and video teleconferencing. The camera automatically tracks the acting or speaking person so that it always remains in the center of the field of view. In manufacturing or recycling environments, an active camera can track objects on the conveyor-belt so that they are recognized and grasped without stopping the belt.

New directions are opened if such an active camera platform is mounted on an autonomous vehicle. As already mentioned in the introduction fixation on an object has computational advantages in navigational tasks. Keeping objects of interest in the center reduces the complexity of processing the dynamic imagery by allowing fine-scale analysis in the center and a coarse resolution level for the periphery. Shifting and holding gaze facilitates also scene exploration and the building of an environmental map.

We start the paper with a description of the related approaches in the next section. In section 3 we describe the kinematics of the binocular head and in section 4 we give the solution to the object detection problem. In section 5 we study the spatiotemporal filtering and in section 6 the estimation and control. We finish with the architecture (section 7) and the presentation of the experimental results (section 8).

2 Related Work

As pursuing is one of the basic capabilities of an active vision system most of the research groups possessing a camera platform have reported results. We divide the approaches in two groups. The first group consists of algorithms that use only motion cues for gaze shifting and holding and this is the group our systems belongs to. Computational basis of this approach group is the difference between measured optical flow and the optical flow induced by camera motion.

The Oxford surveillance system [9, 10] uses data from the motor encoders to compute and subtract the camera motion induced flow. It runs in 25 Hz with processing latency of about 110 ms. Camera behavior is modeled as either saccadic or pursuit motion. Saccadic motion is based on the detection of motion in the coarse scale periphery. Pursuit motion is based only on the optical flow of the foveal region. This is also the difference to our system which can also smoothly pursue but with repeated motion detection. A finite state automaton controls the switching between the two reactions.

The KTH-Stockholm system [11] computes the ego motion of the camera by fitting an affine flow model in the entire image. It is the only approach claiming pursuit in presence of arbitrary observer motion and not only pure rotation as assumed by the rest of the algorithms. However, this global affinity assumption is valid only if the object occupies a minor fraction of the field of view which is not a realistic assumption. Furthermore, the real time (25 Hz) implementation assumes a constant flow model over the entire image. Such a constant flow model is approximately realistic only if the observer's translation is much smaller than the rotation. We experimentally show in the last sequence in Sec. 8 that if flow components induced by slow forward translation are so negligible in comparison to the tracking rotation then they have no effect on the detection task using our approach, either. However, an advantage of the global fitting is that it deliberates the motion detection from the encoder readings.

Elimination of the flow due to known camera rotation is also applied by Murray and Basu [12]. The background motion is compensated by shifting the images. Then large image differences are combined with high image gradients to give a binary image. This binary image is processed with morphological operators and its centroid is extracted. No real time implementation results are reported.

The Bochum system [13] is able to pursue moving objects with a control rate of 2-3 Hz. The full optical flow is computed and then segmented to detect regions of coherent motion signaling an object. The known camera rotation is subtracted only in order to compute the object velocity. Tracking is carried out by a sequence of saccadic and smooth gaze shifts.

Neither of the above approaches involves a study of the appropriate real-time image processing techniques or the control performance. The second group of approaches in object pursuit is based on other cues and a priori knowledge about the object form. Coombs and Brown [14] demonstrated binocular smooth pursuing on objects with vertical

edges with a control rate of 7.5 Hz. Vergence movements are computed using zero-disparity filtering. The authors studied thoroughly the latency problem and the behavior of the α-β-γ-filter. Du and Brady [15] use temporal correlation to track an object that has been detected during the camera was stationary. They achieved a sample rate of 25 Hz with 45 ms latency. Dias et al. [16] present a mobile robot that follows other moving objects which are tracked at approximately human walking rate. Only horizontally moving objects are detected based on very high image differences without ego-motion subtraction. There are further many systems that use very simple image processing to detect and track well-defined targets like white blobs [17, 18] putting emphasis on the control aspect of the problem.

The problem of moving object detection by a moving observer has been intensively studied using passive cameras. However, without the need of a reactive behavior real-time constraints were not considered. The approaches involve global affine flow models [19], temporal coherency models [20], frequency domain methods [21], and variational methods [22], to mention only few of them.

The estimation and control part of our work is related to the approaches dealing with visual servoing. Like our work these approaches apply a regulation criterion in order to control the joints of the robot by means of visual sensor measurements. Most of them put the main emphasis on the controller design and they use a model of the objects to be tracked. Furthermore, many of them apply a more complex regulation criterion like the minimization of both relative position and orientation with respect to an object. The application in this case is grasping a moving object instead of keeping it in the center of the field of view. The most similar control scheme to ours is the first method of Hashimoto and Kimura [23] who also apply optimal LQ control and neglect the robot dynamics. Their second method in [23] considers the robot dynamics and applies input-output feedback linearization. Similar to the latter method is the visual servoing approach by Espiau et al. [24] who introduced the concept of a task function. A task function gives the optimality criterion and expresses the error between actual and dependent on the task desired visual measurements. Feddema et al. [25] concentrate on the selection of geometric features in the image and their impact on the properties of the Jacobian transforming joint angle changes to feature shifts. The error in the image space is transformed to the joint space where the regulation is performed by six PD controllers one for each joint angle. Papanikolopoulos et al. [26] use the optical flow in the center of the image to track an object. Four different control methods (LQG, pole assignment with DARMA and ARMAX models, and PI) are compared with special emphasis on the disturbance treatment. Allen et al. [27] use a stationary stereo camera system and employ object detection by thresholding the normal flow magnitude. A position prediction is based on the curvature of the trajectory and the velocity of the object. Hager et al. [28] use also stationary cameras but they exploit both the image of the end-effector and the object image. A PI controller is applied on the joint angle error obtained by means of the inverse Jacobian of the mapping from angles to stereo measurements.

3 Head kinematics

The binocular camera mount [1] has four mechanical degrees of freedom: the pan angle χ of the neck, the tilt angle ϕ, and two vergence angle θ_l und θ_r for left and right, respectively (Fig. 1). The stereo basis is denoted by b.

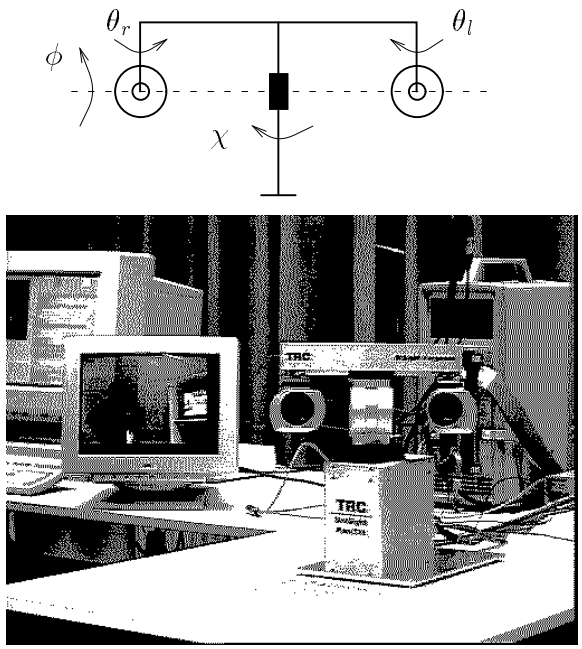

Figure 1. *The four degrees of freedom of the camera platform (top) and how it looks like (bottom).*

We denote by $\boldsymbol{P_w}$ the 4×1 vector of homogeneous coordinates with respect to the *world* coordinate system having origin at the intersection of the pan and the tilt axes. Let $\boldsymbol{P}_{l/r}$ be the vectors with respect to the left and right *effector* coordinate systems located at the intersection of the tilt and the vergence axes. The transformation between world and effector reads

$$\boldsymbol{P_w} = T_\chi T_\phi T_{\theta_{l/r}} \boldsymbol{P}_{l/r}, \tag{1}$$

with

$$T_\chi = \begin{pmatrix} \cos\chi & 0 & -\sin\chi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\chi & 0 & \cos\chi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad T_\phi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

and

$$T_{\theta_{l/r}} = \begin{pmatrix} \cos\theta_{l/r} & 0 & \pm\sin\theta_{l/r} & \mp b/2 \\ 0 & 1 & 0 & 0 \\ \mp\sin\theta_{l/r} & 0 & \cos\theta_{l/r} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Regarding monocular tracking we need only the tilt and the vergence angle of a camera, therefore we omit the subscript in $\theta_{l/r}$. Furthermore, we assume that the effector

[1] Consisting of the TRC BiSight Vergence Head and the TRC UniSight Pan/Tilt Base

coordinate system coincides with the *camera* coordinate system having origin at the optical center. We introduce a *reference* coordinate system with origin at the intersection of the tilt and the vergence axis. The orientation of the reference coordinate system is identical to the resting pose $\phi = 0$ and $\theta = 0$. As monocular visual information gives only the direction of viewing rays we introduce a plane $Z = 1$ whose points are in 1:1 mapping with the rays and are denoted by $\boldsymbol{p} = (x, y, 1)$. The transformation of the viewing ray between reference and camera coordinate system reads

$$\lambda \boldsymbol{p_r} = R_\phi R_\theta \boldsymbol{p_c} \tag{2}$$

with $\boldsymbol{p_c}$ the coordinates after rotations R_ϕ and R_θ about the x and y axis, respectively. The mapping is a projective collineation in P^2. As opposed to translation a pure rotation of the camera induces a projective transformation independent of the depths of the projected points. If a translation existed - like in the mapping between left and right camera - then a point is mapped to a line - the well-known epipolar line - and the corresponding position on this line depends on the depth. After elimination of λ in the above equation we obtain

$$\begin{aligned} x_r &= \frac{x_c \cos\theta + \sin\theta}{-x_c \cos\phi \sin\theta + y_c \sin\phi + \cos\phi \cos\theta} \\ y_r &= \frac{x_c \sin\phi \sin\theta + y_c \cos\phi - \sin\phi \cos\theta}{-x_c \cos\phi \sin\theta + y_c \sin\phi + \cos\phi \cos\theta}. \end{aligned} \tag{3}$$

These equations fully describe the forward kinematics problem.

The inverse kinematics problem is given a camera point $(x_c, y_c, 1)$ to find the appropriate angles so that the optical axis $(0, 0, 1)$ after the rotation is aligned with this point. From (3) we obtain the ray in the reference coordinate system and applying again (3) with $(x_c, y_c) = (0, 0)$ yields

$$\tan\phi = -y_r \qquad \tan\theta = \frac{x_r}{\sqrt{1 + y_r^2}}. \tag{4}$$

We proceed with the computation of the instantaneous angular velocity $\boldsymbol{\omega}$ of the camera coordinate system necessary later for the optical flow representation. Let $R(t) = R_{\phi(t)} R_{\theta(t)}$ be the time varying rotation of the camera coordinate system and Ω the skew-symmetric tensor of the angular velocity. Then we have $\dot{R}(t) = R(t)\Omega$ and the angular velocity with respect to the moving coordinate system reads

$$\boldsymbol{\omega} = \begin{pmatrix} \dot{\phi}\cos\theta & \dot{\theta} & \dot{\phi}\sin\theta \end{pmatrix}^T. \tag{5}$$

To complete the geometric description we need the transformation from pixel coordinates (x_i, y_i) in the image to viewing rays in the camera coordinate system. This is an affine transformation given by

$$x_i = \alpha_x x_c + x_0 \qquad y_i = \alpha_y y_c + y_0.$$

The scaling factors α_x, α_y depend on the focal length, the cell size on the CCD-chip, and the sampling rate of the A/D converter. The principal point (x_0, y_0) is the intersection of the optical axis with the image plane. For the computation of this transformation -called intrinsic calibration- we applied conventional [29] as well as active techniques similar to [30, 31].

4 Pursuing a moving object

Pursuing is accomplished by a series of correcting saccades to the positions of the detected object which yield a trajectory as smooth as possible due to our control scheme and the under-cascaded axis-control of the mount. A moving object in the image is defined as the locus of points with high image gradient whose image motion is substantially different from the camera induced image motion. We exploit the fact that the camera induced optical flow is pure rotational

$$\boldsymbol{u_c} = \begin{pmatrix} x_c y_c & -(1+x_c^2) & y_c \\ (1+y_c^2) & -x_c y_c & -x_c \end{pmatrix} \boldsymbol{\omega} \tag{6}$$

where $\boldsymbol{\omega}$ can be computed from (5) using the angle readings of the motion encoder. If $\boldsymbol{u} = (u, v)$ is the observed optical flow then $\boldsymbol{u} - \boldsymbol{u_c}$ is the optical flow induced only from object motion. We assume the Brightness Change Constraint Equation

$$g_x u + g_y v + g_t = 0$$

with g_x, g_y and g_t the spatiotemporal derivatives of the grayvalue function. From this equation we can compute only the normal flow - the projection of optical flow in the direction of the image gradient (g_x, g_y). The difference between the normal flow u_{c_n} induced by camera motion and the observed normal flow u_n

$$u_{c_n} - u_n = \frac{g_x u_c + g_y v_c}{\sqrt{g_x^2 + g_y^2}} + \frac{g_t}{\sqrt{g_x^2 + g_y^2}}$$

is the normal flow induced by the object motion. It turns out that we can test the existence of object image motion without the computation of optical flow. The sufficient conditions are that the object motion has a component parallel to the image gradient and the image gradient is sufficiently large. We can thus avoid the computation of full optical flow which would require the solution of at least a linear system for every pixel. Three thresholds are applied: the first for the difference between observed and camera normal flow, the second for the magnitude of the image gradient, and the third for the area of the points satisfying the first two conditions. The object position is given as the centroid of the detected area.

5 Real time spatiotemporal filtering

Special effort was given to the choice of filters suitable for the used pipeline-processor [2] so that the frequency domain specifications are satisfied without violating the real time requirements. Whereas up to 8×8 FIR-kernels can be convolved with the image with processing rate of 20 MHz the temporal filtering must be carried out by delaying the images in the visual memory. We chose IIR filtering for the computation of the temporal derivatives since its computation requires less memory than temporal FIR filtering for the same effective time lag.

The temporal lowpass filter chosen is the discrete version of the exponential [32]

$$E(t) = \begin{cases} \tau e^{-t\tau} & t \geq 0 \\ 0 & t < 0. \end{cases}$$

[2] Datacube MaxVideo 200 board

If $E_n(t)$ is the n-th order exponential filter ($n \geq 2$) its derivative reads

$$\frac{dE_n(t)}{dt} = \tau(E_{n-1}(t) - E_n(t)).$$

After applying the bilinear mapping $s = 2(1 - z^{-1})/(1 + z^{-1})$ to the Laplace transform $\tau/(s + \tau)$ of the exponential filter from the s-plane to the z-plane we obtain the transfer function of the discrete lowpass filter

$$H(z) = q\frac{1 + z^{-1}}{1 + rz^{-1}}, \qquad q = \frac{\tau}{\tau + 2} \quad r = \frac{\tau - 2}{\tau + 2}.$$

If $H(z)^n$ is the n-th order low pass filter its derivative is equal to the difference $\tau(H(z)^{n-1} - H(z)^n)$ of two lowpass filters of subsequent order. The recursive implementation for the second order filter reads

$$\begin{aligned} h_1(k) + rh_1(k-1) &= q(g(k) + g(k-1)) \\ h_2(k) + rh_2(k-1) &= q(h_1(k) + h_1(k-1)) \\ g_t(k) &= \tau(h_1(k) - h_2(k)), \end{aligned}$$

where $g(k)$ is the input image, $h_1(k)$ and $h_2(k)$ are the lowpass responses of first and second order, respectively, and $g_t(k)$ is the derivative response. We note, that the lowpass response is used to smooth temporally the spatial derivatives.

The spatial FIR-kernels are binomial approximations to the first derivatives of the Gaussian function [33]. The spatial convolutions are carried out in fixed-point 32 bit arithmetic with the result stored in 8 bit word length. The inverse of the magnitude of the spatial gradient needed for the computation of normal flow is computed using a LUT table. Fixed-point arithmetic affects primarily the IIR filtering since the binomial coefficient can be represented by the quotient of powers of two. We use the Diverging Tree sequence [34] as a testbed for our accuracy investigations. The ground truth optical flow field is known and we test the filtering effects on the computation of the optical flow field. We use a conventional method [35] that assumes local constancy of the optical flow field. In Fig. 2 we show the 20th image of the sequence as well as the optical flow field based on the spatiotemporal derivatives computed with fixed-point arithmetic. In Fig. 2 (bottom) we compare the average relative error between fixed-point and floating-point filtering as a function of the flow vector length which increases with the distance from the focus of expansion. In the central area of ± 30 pixels the relative errors vary from 200% down to 10%. After this distance we note a constant bias in the fixed-point case of 2.5% error relative to the floating-point case. The fixed-point effects are severe only for lengths between 0.2 and 0.4 pixels.

We proceed with studying the differences between temporal FIR and IIR filtering in order to justify the choice of the recursive IIR filter described above. The delay of the temporal FIR first gaussian derivative (and its binomial approximation) is equal to half of the kernel size. The delay for the second order exponential filter is between the mode $1/\tau$ and the mean $2/\tau$. We show in Fig. 3 (top) the continuous impulse responses for a Gaussian derivative with standard deviation $\sigma = 1$ and the second order exponential filter with $\tau = 1$. The zero-crossings of both filters coincide but the IIR filter is highly asymmetric. For these settings we show the spectra of the two filters in the middle of Fig. 3

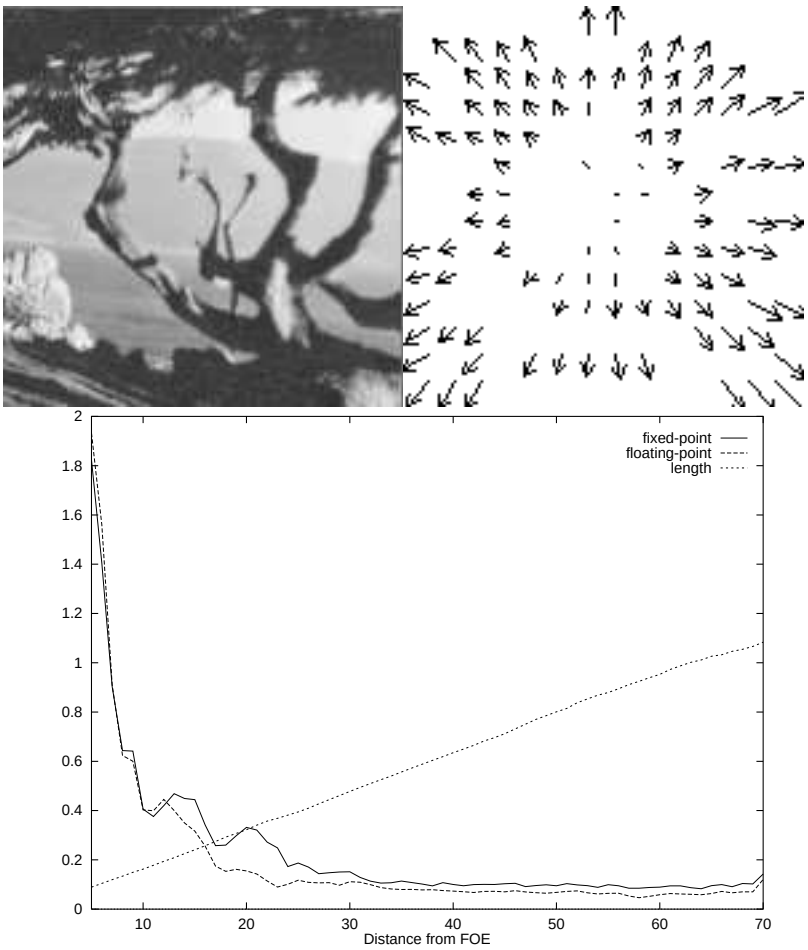

Figure 2. *The 20th image of the Diverging Tree sequence (above left), the optical flow field computed with the fixed point implementation of the FIR and IIR filters (above right), and the relative error in the estimation of optical flow of fixed- vs. floating-point arithmetic. The relative error as well as the flow vector length are plotted as functions of the distance from the focus of expansion, here the center of the image (below).*

as well as the goodness of differentiation in Fig. 3 (bottom). The latter is obtained by dividing the frequency response of the derivative filters with the frequency response of the involved low-pass filters: a low-pass binomial mask in the FIR case and the exponential in the IIR case. We observe that FIR outperforms IIR for frequencies in the transition band and both are similar for low frequencies.

We compare the behavior of both filters in the computation of optical flow in the same sequence as above. We tested several settings for the parameters of both filters. The average relative errors for about the same densities[3] of computed vectors are shown in Table 1. The IIR filters were computed with a delay of one frame. The best results are obtained for an FIR kernel of length 7 and for a recursive IIR with $\tau = 1.0$.

We applied the same tests in one more sequence with known ground truth, the Yosemite sequence. The results (Tab. 2) are worse in this sequence - but comparable

[3]Density is the ratio of image positions where the flow computation satisfies a confidence measure divided by the image area

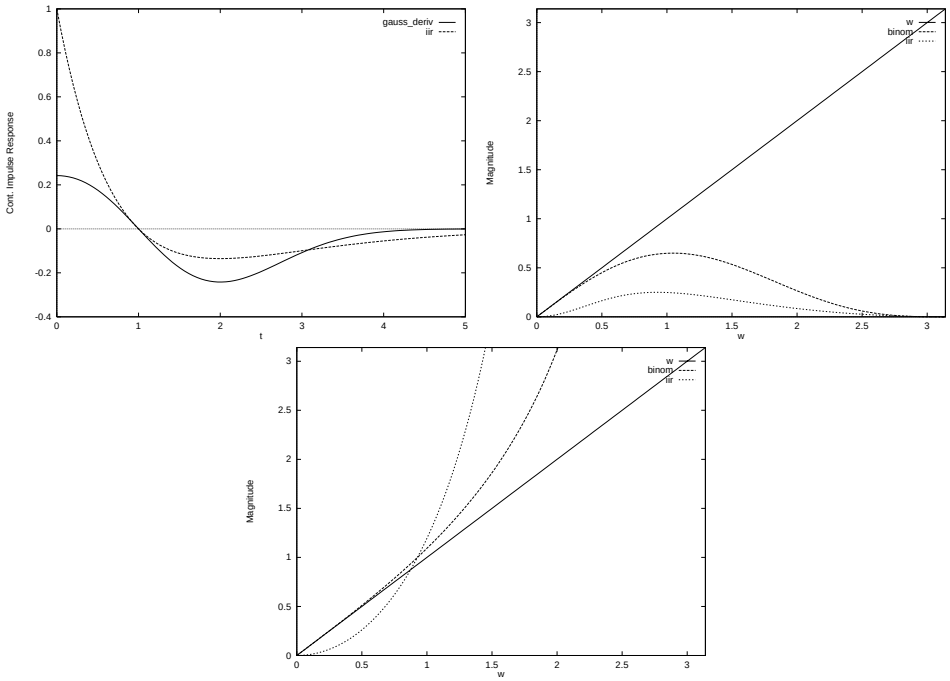

Figure 3. *Continuous impulse response comparison of the shifted first derivative of a Gaussian ($\sigma = 1$) and the IIR second order derivative filter ($\tau = 1$) (above). In the middle we show the frequency responses of the 5-points binomial approximation of the first Gaussian derivative and the IIR second order derivative filter ($\tau = 1$). Below we show the pure differentiation effects, i.e. the same spectra divided by the frequency responses of the low-pass prefilters.*

to the reported in the survey [34] - and qualitatively the same as in the Diverging Tree sequence with the exception of the FIR filter which shows the best accuracy with a kernel length of 5.

filter	aver. rel. error (%)	vector density (%)
IIR ($\tau = 0.5$)	10.55	52.33
IIR ($\tau = 1.0$)	9.88	52.29
IIR ($\tau = 1.25$)	10.26	52.21
IIR ($\tau = 2.0$)	11.96	52.84
FIR (3p)	11.62	52.83
FIR (5p)	10.01	52.23
FIR (7p)	9.89	52.21

Table 1 *The average relative error in the Diverging Tree sequence for different τ's and kernel lengths.*

Considering the used architecture (MaxVideo200) a temporal FIR filter needs as many image memories as the kernel length N. The computational cost is N multiplications and

filter	aver. rel. error (%)	vector density (%)
IIR ($\tau = 0.75$)	28.47	50.74
IIR ($\tau = 1.0$)	19.96	50.62
IIR ($\tau = 1.25$)	20.04	50.60
IIR ($\tau = 2.0$)	22.09	50.28
FIR (3p)	25.21	50.56
FIR (5p)	19.61	50.38
FIR (7p)	22.42	50.86

Table 2 *The average relative error in the Yosemite sequence for different τ's and kernel lengths.*

$N-1$ additions and the delay $(N-1)/2$ frames. Our second order IIR filter implementation uses four image memories with the complexity of two multiplications and three additions. The delay for $\tau = 1$ is between one and two frames. Taking into account the almost negligible difference in the flow computation performance the IIR filter guarantees the same motion behavior with much lower space and time complexity.

6 Estimation and Control

The control goal of pursuing is to hold the gaze as close as possible to the projection of a moving object. Actuator input signals are the pan angle ϕ and the vergence angle θ. Since the angles can be uniquely obtained from the position (x_r, y_r) through (4) we use the reference coordinates (x_r, y_r) as input vector. The intersection of the optical axis with the plane $Z = 1$ of the reference coordinate system is denoted by $\boldsymbol{c}$. Output measurements are the position of the object in the reference coordinate system denoted by $\boldsymbol{o}$ obtained from the centroid in the image and (3). Let $\boldsymbol{v}$ and $\boldsymbol{a}$ be the velocity and acceleration of the object and $\Delta\boldsymbol{u}(k)$ the incremental correction in the camera position. The state is described by the vector

$$\boldsymbol{s} = \begin{pmatrix} \boldsymbol{c}^T & \boldsymbol{o}^T & \boldsymbol{v}^T & \boldsymbol{a}^T \end{pmatrix}^T .$$

A motion model of constant acceleration yields the plant

$$\boldsymbol{s}(k+1) = \Phi\boldsymbol{s}(k) + \Gamma\Delta\boldsymbol{u}(k) \quad \text{with} \quad \Phi = \begin{pmatrix} I_2 & O_2 & O_2 & O_2 \\ O_2 & I_2 & \Delta t I_2 & \Delta t^2/2 I_2 \\ O_2 & O_2 & I_2 & \Delta t I_2 \\ O_2 & O_2 & O_2 & I_2 \end{pmatrix}$$

and

$$\Gamma = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

where I_2 and O_2 are 2×2 identity and null matrix, respectively. Assuming a linear control function $\Delta\boldsymbol{u}(k) = -K\hat{\boldsymbol{s}}(k)$ with $\hat{\boldsymbol{s}}$ an estimate of the state we make use of the separation principle stating that optimal control can be obtained by combining the optimum deterministic control with the optimal stochastic observer [36].

The minimization of the difference $\|\boldsymbol{o} - \boldsymbol{c}\|$ between object and camera position in the reference coordinate system can be modeled as a Linear Quadratic Regulator problem

with the minimizing cost function $\sum_{k=0}^{N} \boldsymbol{s}^T(k) Q \boldsymbol{s}(k)$ where Q is a symmetric matrix

$$Q = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

In steady state modus a constant control gain K is assumed resulting in an algebraic Ricatti equation with the simple solution

$$K = \begin{pmatrix} 1 & -1 & -\Delta t & -\Delta t^2/2 \end{pmatrix}. \tag{7}$$

The meaning of the solution is that input camera position should be equal the predicted position of the object. One of the crucial problems in vision based closed loop control is how to tackle the delays introduced by a processing time longer than a cycle time. We emphasize here that the delay in our system is an estimator delay. The normal flow detected after frame k concerns the instantaneous velocity at frame $k-1$ due to the mode of the IIR temporal filter. At time $k-1$ the encoder is also asked to give the angle values of the motors. To the delay amount of one frame we must add the processing time so that we have the complete latency between motion event and onset of steered motion. The prediction (7) enables a compensation for the delayed estimation by appropriate settings for Δt in the gain equation.

Concerning optimal estimation we also assume steady state modus obtaining a stationary Kalman Filter with constant gains. The special case of a second order plant yields the well known α-β-γ-Filter [37] with update equation

$$\hat{\boldsymbol{s}}^+(k+1) = \hat{\boldsymbol{s}}^+(k) + \begin{pmatrix} \alpha & \beta/\Delta t & \gamma/\Delta t^2 \end{pmatrix}^T (\boldsymbol{m}(k+1) - \boldsymbol{m}^-(k+1)),$$

where $\boldsymbol{s}^+$ is the state after updating and $\boldsymbol{m}^-$ is the predicted measurement. The gain coefficients α, β and γ are functions of the target maneuvering index λ. This maneuvering index is equal to the ratio of plant noise covariance and measurement noise covariance. The lower is the maneuvering index the higher is our confidence in the motion model resulting to a smoother trajectory. The higher is the maneuvering index the higher is the reliability of our measurement resulting to a close tracking of the measurements which may be very jaggy. This behavior will be experimentally illustrated in the following.

In this experimental study we excluded the image processing effects by moving an easily recognizable light-spot. We controlled the motion of the light-spot by mounting it into the gripper of a robotic manipulator. The control frame rate is equal to the video frame rate (30 Hz). The world trajectory of the light-spot is a circle with radius equal 20cm on a plane perpendicular to the optical axis in resting position. The center of the circle was 145cm in front of and 80cm below the head.

We varied the angular velocity of the light-spot and for every velocity we observed the tracking behavior for different maneuvering indices. We first tested the tracking error for the high velocity of 1 target revolution per 823 ms (1.2 Hz, Fig. 4). The maneuvering index λ was set equal to 1. The motors reached an angular velocity of about 45 deg/s in both tilt and vergence angles. In order to decrease time complexity we first tested the possible application of first order motion model with an $\alpha\beta$-filter. We applied both filters for a target velocity of 0.52 Hz (Fig. 5). The behavior of the first order filter is

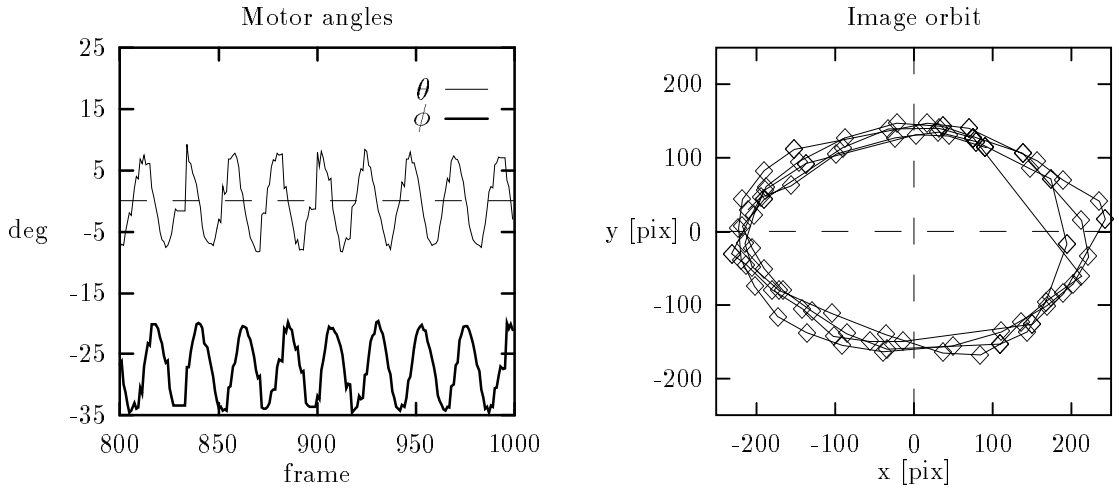

Figure 4. *The tilt ϕ and vergence θ angles (left) and the image orbit of the target (right) with the large error due to the high velocity (1.2 Hz) of the target.*

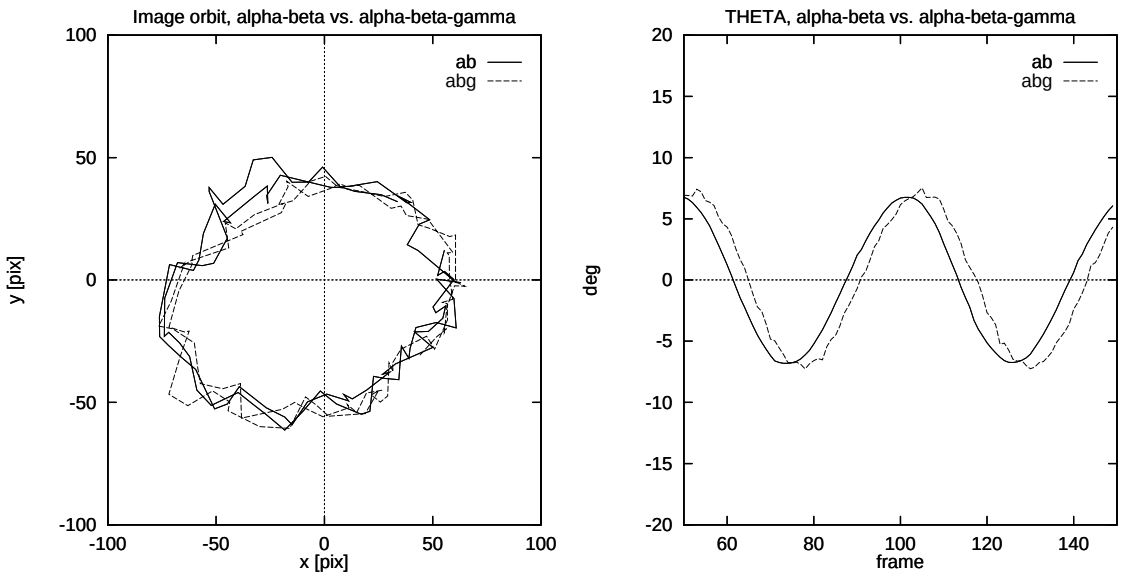

Figure 5. *Image orbit (left) and vergence angle vs. time (right) for the $\alpha\beta$- and the α-β-γ-filter.*

satisfactory with the additional advantage that it is not as jaggy as the α-β-γ-filter. We applied, therefore, in all following tests the $\alpha\beta$ filter.

We, then, tested the controller for two different maneuvering index values $\lambda = 0.1, 1$ and four different velocities of the target starting from 0.17 Hz up to 0.70 Hz (Fig. 6). The pixel error increases with the velocity of the target. It is higher for the low maneuvering index as expected but with smoother image orbit.

Then we let the maneuvering index vary by keeping constant the velocity (Fig. 7). The decreasing smoothness with increasing λ can be observed in the image orbit as well as in the trace of the vergence angle along time.

To resume, we do not expect a pixel error better than ± 10 pixels for the highest maneuvering index if we assume that the object motion trajectory is as smooth as a circle.

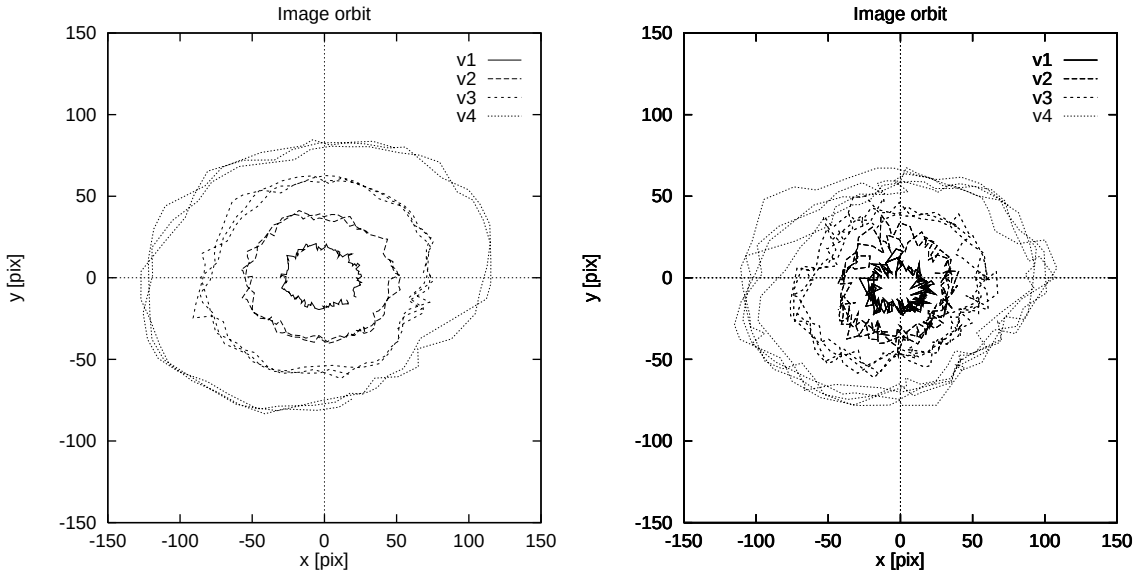

Figure 6. *Image orbit of the target for four different velocities* $v_{1..4}$*=(0.17 Hz, 0.35 Hz, 0.52 Hz, and 0.70 Hz) for* $\lambda = 0.1$ *(left) and* $\lambda = 1.0$ *(right).*

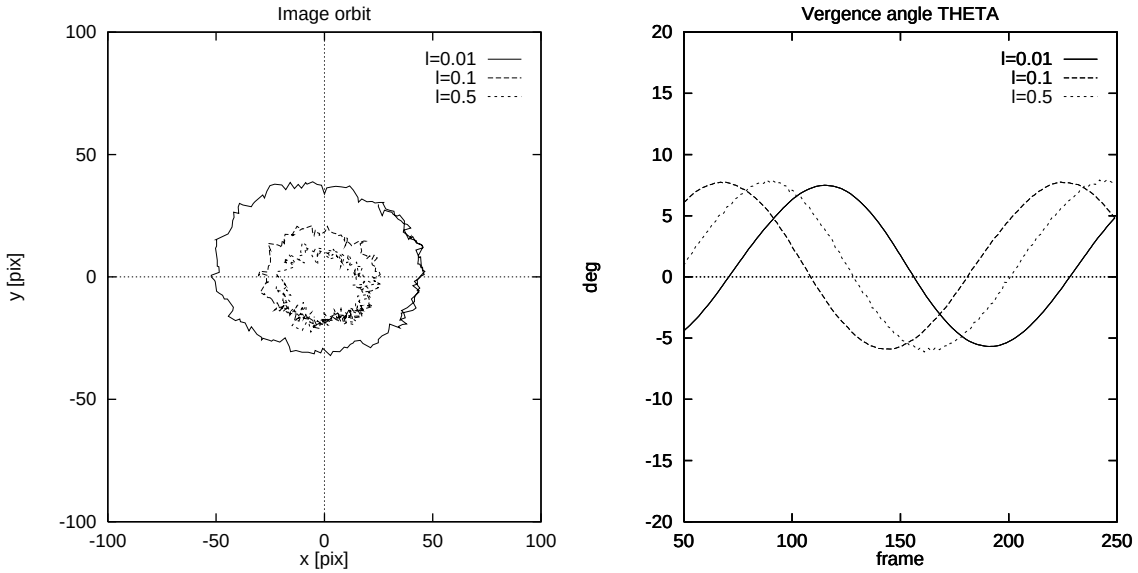

Figure 7. *Image orbit of the target for three values of* $\lambda = 0.01, 0.1, 0.5$ *(left) and the vergence angle as a function of time (right).*

As we will observe in the experiments with usual moving objects instead of light-spots the trajectory of the detected moving area is so irregular that only a high maneuvering index can lead to smaller tracking errors.

7 Integration and system architecture

The image processing and control modules above were implemented on an architecture consisting of several commercial components (Fig. 8).

We summarize here all processing steps of the loop:

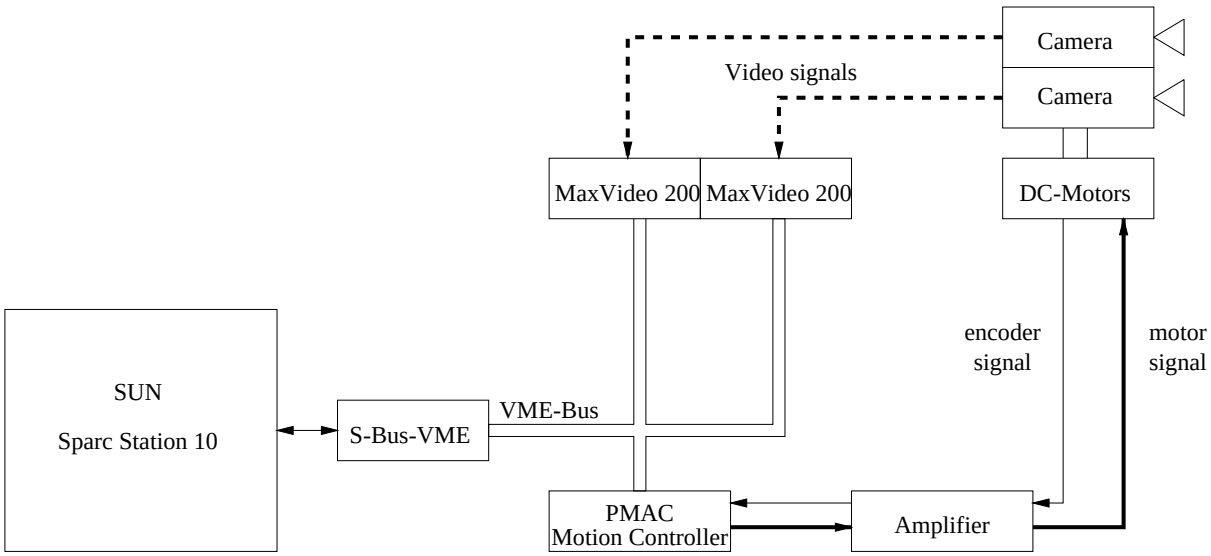

Figure 8. *Hardware architecture of the closed-loop.*

1. The current tilt and vergence angle values are read out from the encoders.
2. The video signal is transmitted from the camera[4] to the MaxVideo 200 board where it is digitized, lowpass filtered and subsampled to a resolution of 128x128. The real time operating system[5] on the SparcStation enables the firing of the image acquisition exactly after the angle reading in the last step.
3. The spatial derivatives are computed by convolving with 7x7 binomial masks.
4. The spatial derivatives are lowpass filtered with an IIR filter. The temporal derivatives are computed with an IIR filter and then spatially smoothed with a 7x7 binomial kernel.
5. The normal flow difference is computed using the LUT table of the inverse of the gradient magnitude.
6. The difference image and the gradient magnitude image are thresholded and combined with a logical AND. On the resulting binary image $b(x, y)$ are computed the sums $\sum xb(x, y)$ and $\sum yb(x, y)$ as well as the area. The resulting vectors are transmitted to the SparcStation.
7. The centroid of the detected area is computed and then transformed to the reference coordinate system using the intrinsic parameters and the angle readings.
8. The state is updated with the α-β-γ-filter.
9. The state is predicted considering the time delay and the input camera position is obtained in the reference coordinate system.
10. The desired camera position is transformed to the tilt and vergence angles.

[4]We use Sony XC-77RR with a frame rate of 30 Hz.
[5]Solaris 2.4

11. The angles are transmitted to the motion controller.

12. The motion controller runs its own axis control with rate 2kHz, computes the intrapoint trajectory, and sends the analog control signals to the amplifier.

By means of the `setitimer(ITIMER_REAL,..)` function of the Solaris 2.4 operating system we guarantee a cycle time of 40ms. This cycle time consists of 37ms image processing (steps 2-6 performed on the MaxVideo 200 board) and 3ms control (steps 7-10 performed in the SparcStation). The motion controller guarantees a motion execution time of 40ms. Considering the effective delay of the temporal derivatives calculations of one frame we obtain an effective latency of 80 ms between event and onset of motion. The motion duration is equal to the processing cycle time so that the camera reaches the desired position 120 ms after the event detected. The prediction for the control signal is computed with respect to this lag.

8 Experiments

We show here the performance of the active tracking system in four different object motions. The images in the figures are chosen out of 20 frames saved "on the fly" during a time of 8s. The images are overlaid with those points on the images where both the normal flow difference and the gradient magnitudes exceed two thresholds which are the same for all four experiments. The centroid of the detected motion area is marked with a cross. We show the tracking error by drawing the trajectory of the centroid in the image as well as the control values for the tilt and the vergence angle, ϕ and θ for the entire time interval of 8s.

In all the experiments the motion tracking error is much higher than the light-spot tracking error. This was expected since the object is modeled in the image by its centroid. Although the target might move smoothly the orbit of the centroid depends on the distribution of the detected points in the motion area. Therefore, it is corrupted with an error of very high measurement variance. Allowing a high maneuvering index which enables close tracking would result in a extreme jaggy motion of the camera. The estimator would forget the motion model and yield an orbit as irregular as the centroid motion. Therefore, we decrease the maneuvering index to 0.01 and obtain as expected a much higher pixel error. Only a post processing of the binary images could improve the position of the detected centroid.

In the first experiment (Fig. 9) the system is tracking a Tetrapak moving from right to left. The small size of the target enables a relatively small pixel error (the target is always observed left from the center). Because the centroid variation is only in the vertical direction - due to the rod holding the target - the tilt angle changes irregularly. The average angular velocity is 8.5 deg/s.

In the second experiment (Fig. 10) we moved a rotating target from right to left and then again to right, first downwards and then upwards. The achieved angular velocity is 10 deg/s. Due to the rotation of the target the normal flow due to object motion is higher yielding, thus, many points above the set threshold. We should emphasize here that algorithms like [11] based on a global ego-rotation fitting would fail since the object covers a considerable part of the field of view.

The same fact characterizes the third experiment (Fig. 11). A box attached in the gripper of a manipulator is moving in a circular trajectory with 0.35 Hz. Here, the target

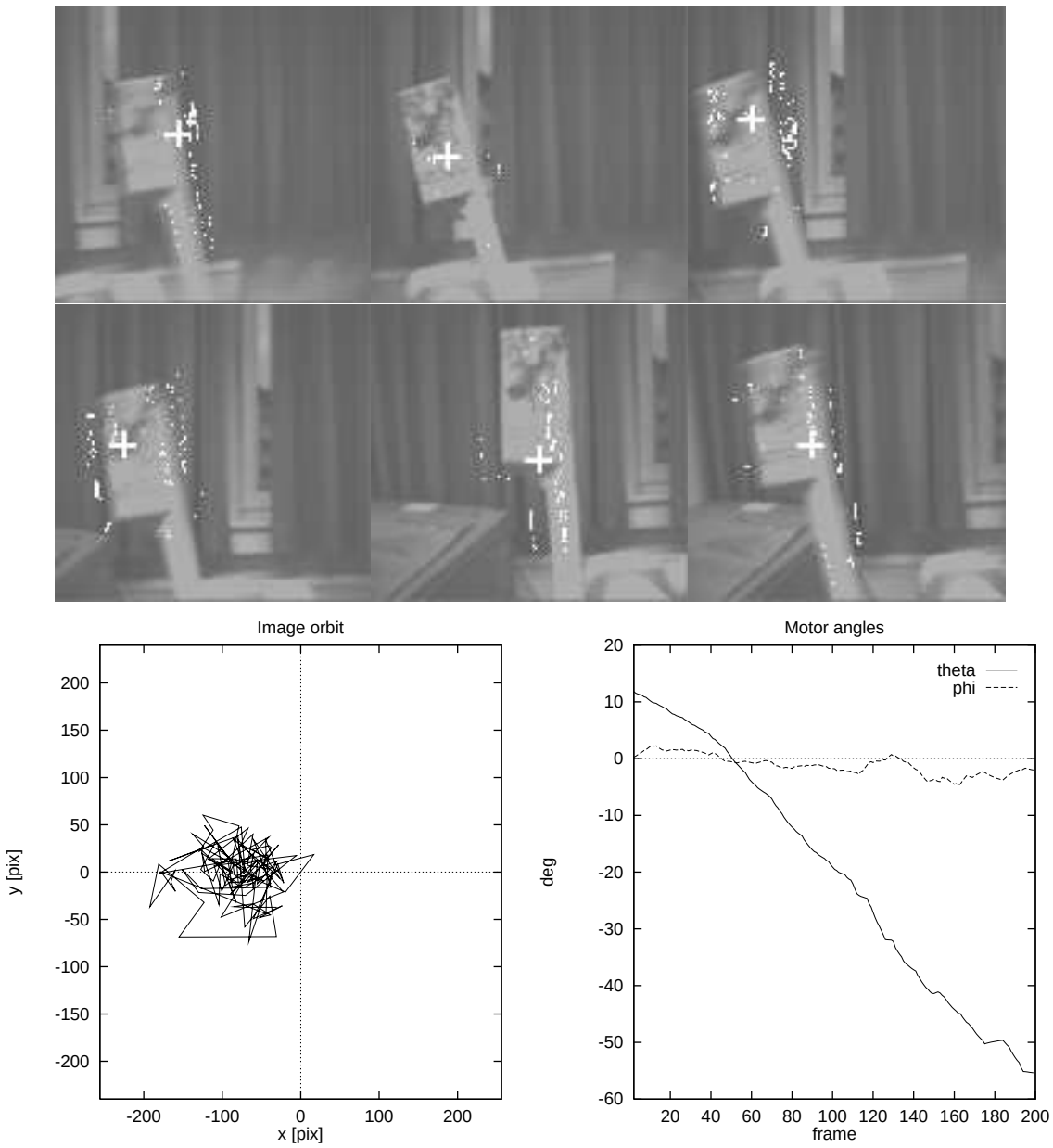

Figure 9. *Six frames recorded while the camera is pursuing a Tetrapak moving from right to left. The pixel error (bottom left) shows that the camera remains behind the target and the vergence change (bottom right) shows the turning of the camera from right to left with an average angular velocity of 8.5 deg/s.*

is not distinctly defined because all joints after the elbow give rise to image motion. The centroid is continuously jumping in the image. However, the system was able to keep the object in an area of ± 130 pix or ± 10 deg visual angle.

In the last experiment, we asked the system to try again a target attached on the manipulator (Fig. 12). However, we moved forward the vehicle where the head is mounted on. This situation is not modeled by our ego motion assumed as pure rotation. With a forward translation of 10cm/s nothing changed in the average pixel error. The approaching of the camera is evident in the image as well as in the angle plots: Positive shift in the

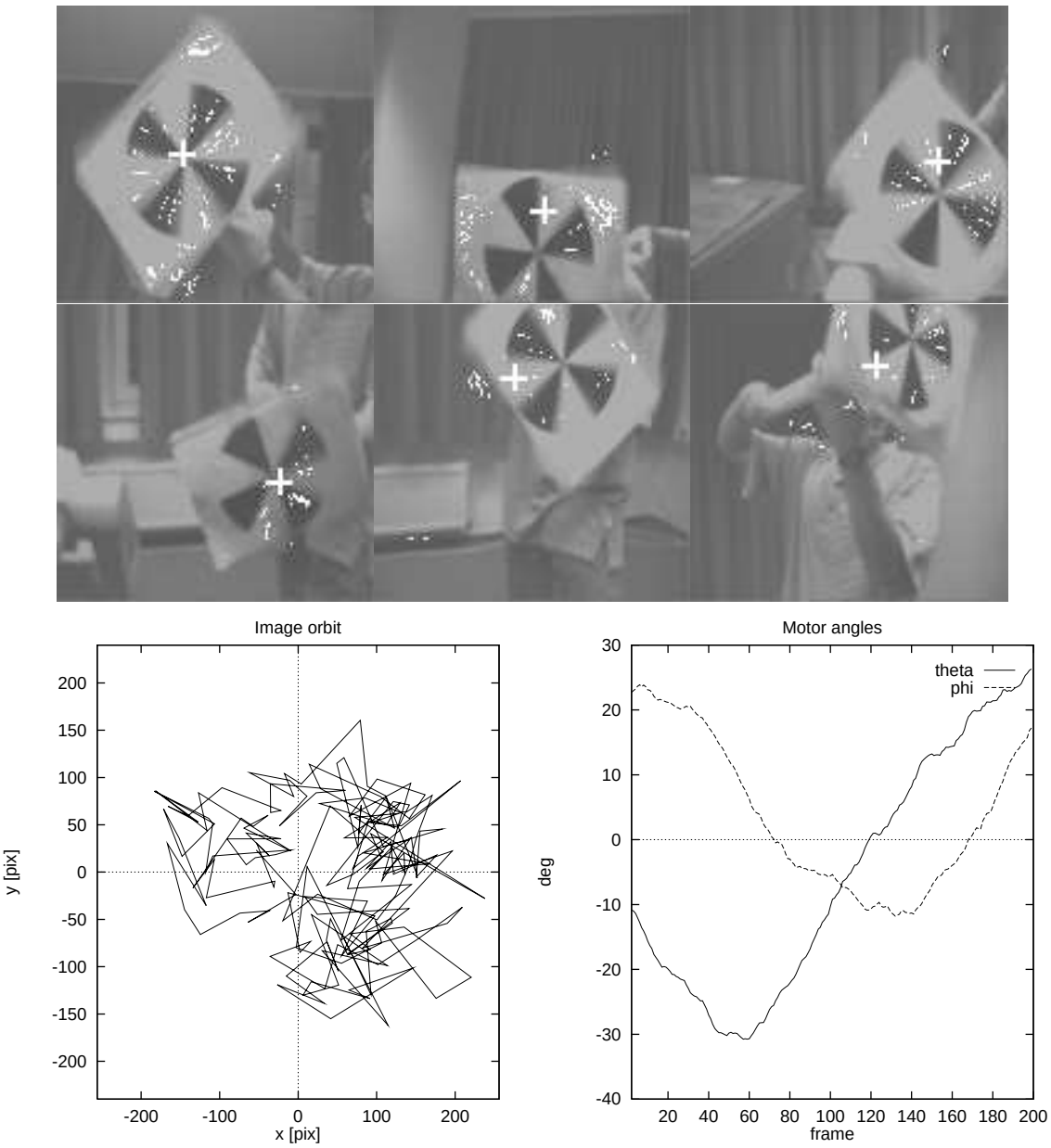

Figure 10. *Six frames recorded while the camera is pursuing a rotating target moving from right to left and then again to right, first downwards and then upwards. The average angular velocities for both the vergence and the tilt are 10 deg/s.*

vergence mean (indicating approaching the left side of the target) and negative shift in the tilt mean (showing the viewing downwards). The reason of this surprisingly good behavior is in the components of the optical flow. As soon as the camera rotates the rotational component is much larger than the translational one so that the effects on the normal flow difference are negligible.

9 Conclusion

We presented a system that is able to detect and pursue moving objects without knowledge of their form or motion. The performance of the system with control rate of 25 Hz, a

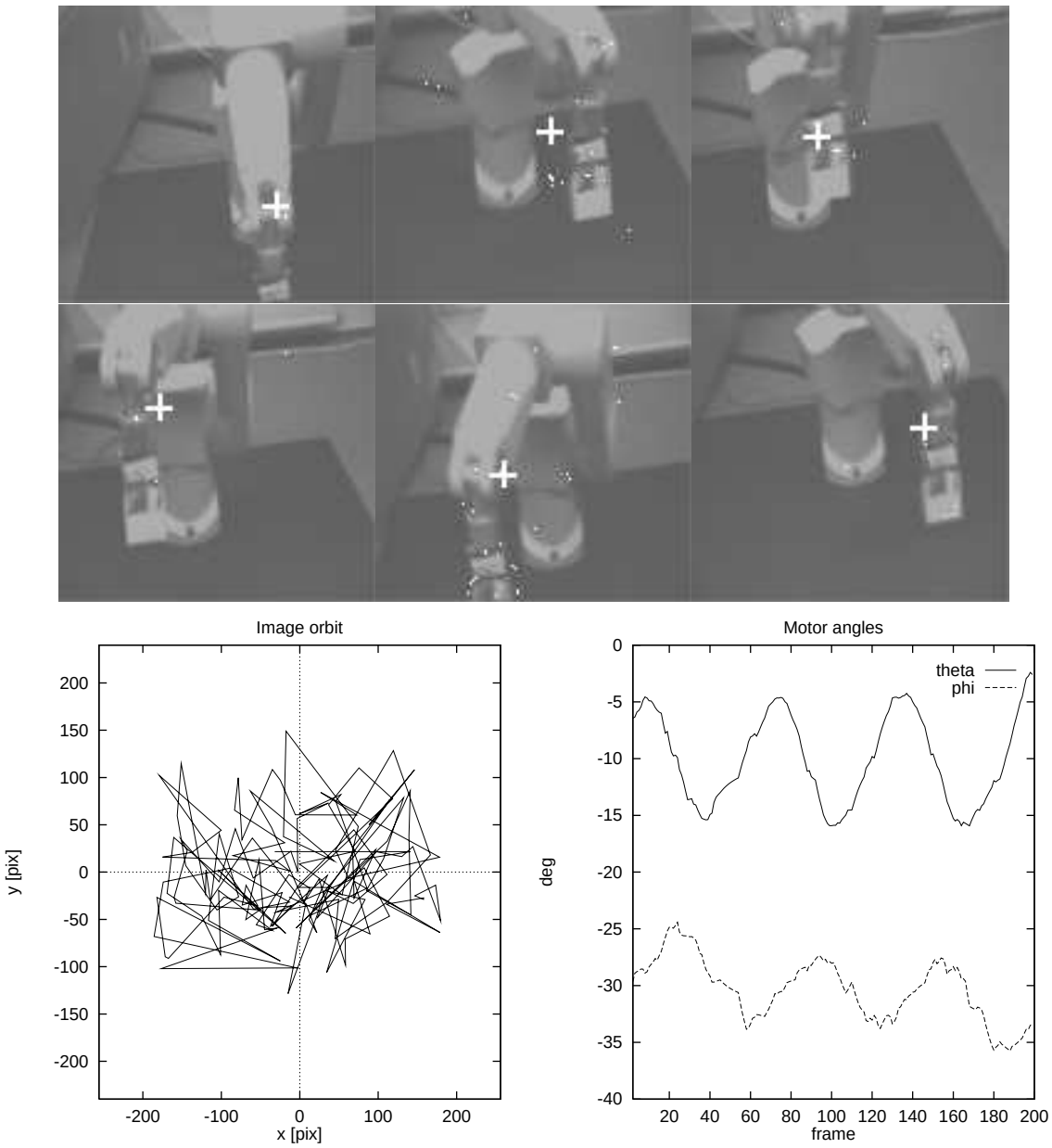

Figure 11. *The camera is pursuing a target attached in the gripper of a manipulator. The target is moving on a circle with frequency 0.35 Hz. The angular velocity is 8 deg/s for vergence and 5 deg/s for tilt.*

latency of 80 ms, and achieved average angular velocities of about 10 deg/s is competitive with respect to the state of the art. The system needs the minimal number of tuning parameters: a threshold for normal flow difference, a threshold for the image gradient, a minimal image area over the mentioned thresholds, and the maneuvering index.

We have shown that in order to achieve real-time reactive behavior we must apply the appropriate image processing and control techniques. The main contribution of this paper is not only in the achieved high performance of the system. Our work is different from other presentations in the study of the individual components with respect to the given hardware, time constraints, and desired tracking behavior. We experimentally studied

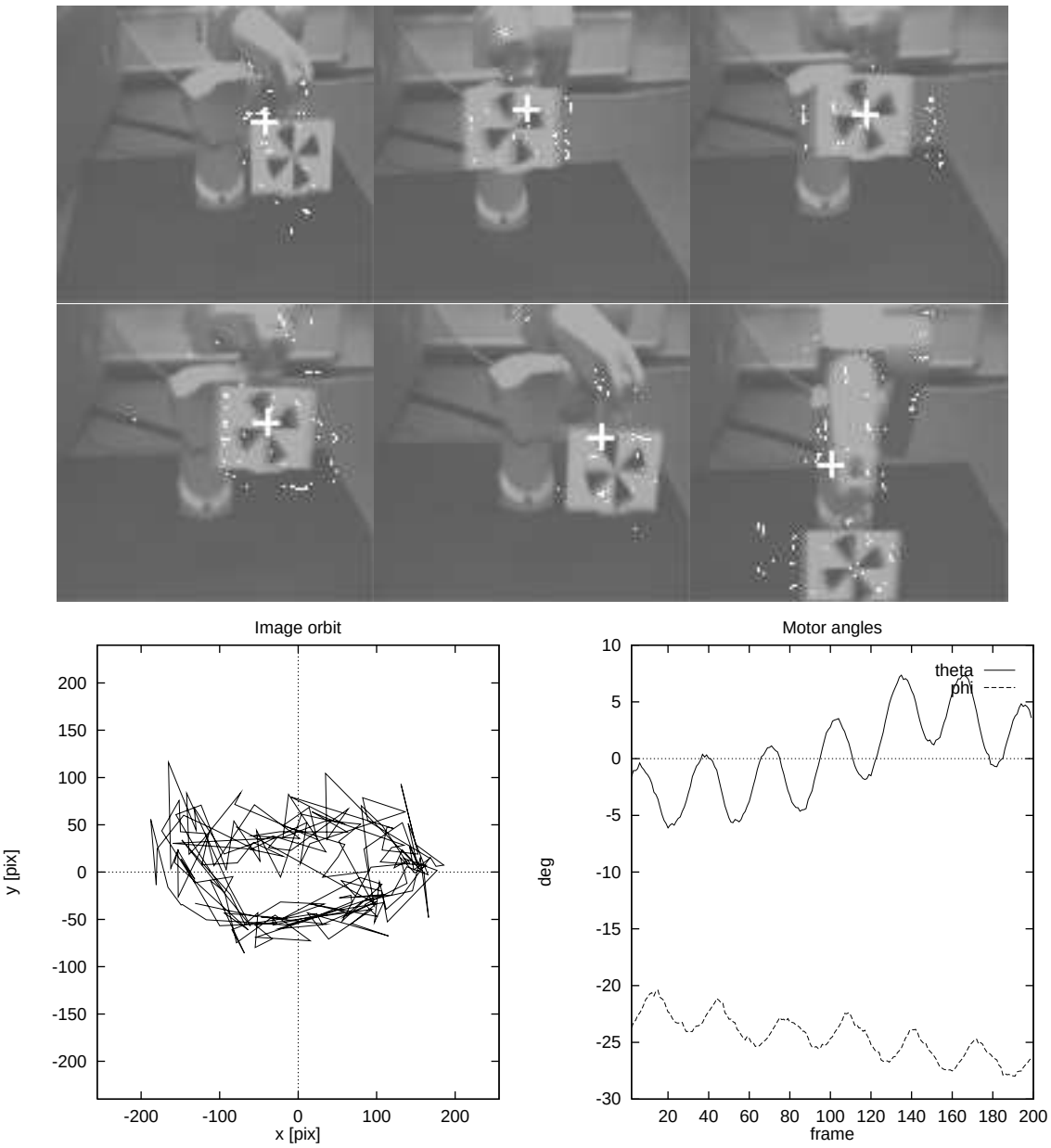

Figure 12. *The camera is pursuing a target mounted on the gripper of a manipulator while the camera is itself translating forwards. The target is moving on a circle with frequency 0.70 Hz. The translation of the camera is shown in the shift of the angle oscillation center. As the camera is approaching on the left side of the manipulator it must turn more to the right (positive shift in vergence) and more downwards (negative shift in tilt).*

the responses of the image processing filters if fixed-point arithmetic is used. We studied the trade-off between space-time complexity and response accuracy concerning the choice of FIR or IIR filtering. We dwelled on the control and estimation problem by testing the behavior of the applied estimator with different parameters. Last but not least, we presented experimental results of the integrated system in four different scenarios with varying form and motion of the object.

The system will be enhanced with foveal pursuing based on the full optical flow values in a small central region. A top-down decision process is necessary for shifting attention in case of multiple moving objects. The presented work is just the first step of a long procedure. The goal is the building of a behavior-based active vision system. The next reactive oculomotor behaviors in plan are the vergence control and the optokinetic stabilization.

Acknowledgements: We highly appreciate the contributions of Henrik Schmidt in programming the camera platform, of Jörg Ernst in the intrinsic calibration, and of Gerd Diesner in Datacube programming. We gratefully acknowledge discussions with Ulf Cahn von Seelen from GRASP Lab.

References

[1] R. Bajcsy. Active Perception. *Proceedings of the IEEE*, 76:996–1005, 1988.

[2] Y. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active Vision. *International Journal of Computer Vision*, 1:333–356, 1988.

[3] M. Tistarelli and G. Sandini. Dynamic aspects in active vision. *CVGIP: Image Understanding*, 56:108–129, 1992.

[4] Y. Aloimonos, editor. *Active Perception.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1993.

[5] C.M. Brown. Issues in selective perception. In *Proc. Int. Conf. on Pattern Recognition*, pp. 21–30. The Hague, The Netherlands, Aug. 30-Sep. 3, 1992.

[6] A. Bandopadhay and D.H. Ballard. Egomotion perception using visual tracking. *Computational Intelligence*, 7:39–47, 1990.

[7] C. Fermüller and Y. Aloimonos. Tracking facilitates 3-D motion estimation. *Biological Cybernetics*, 67:259–268, 1992.

[8] R.H.S. Carpenter. *Movements of the Eyes.* Pion Press, London, 1988.

[9] D.W. Murray, P.L. McLauchlan, I.D. Reid, and P.M. Sharkey. Reactions to peripheral image motion using a head/eye platform. In *Proc. Int. Conf. on Computer Vision*, pp. 403–411, Berlin, Germany, May 11-14, 1993.

[10] K.J. Bradshaw, P.F. McLauchlan, I.D. Reid, and D.W. Murray. Saccade and pursuit on an active head-eye platform. *Image and Vision Computing*, 12:155–163., 1994.

[11] P. Nordlund and T. Uhlin. Closing the loop: Pursuing a moving object by a moving observer. In *V. Hlavac et al. (Ed.), Proc. Int. Conf. Computer Analysis of Images and Patterns CAIP, Prag, Springer LNCS vol. 970*, pp. 400–407, 1995.

[12] D. Murray and A. Basu. Motion tracking with an active camera. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16:449–459, 1994.

[13] S. Tölg. Gaze control for an active camera system by modeling human pursuit eye movements. In *Proc. SPIE Vol. 1825 on Intelligent Robots and Computer Vision*, pp. 585–598, 1992.

[14] D. Coombs and C. Brown. Real-time binocular smooth pursuit. *International Journal of Computer Vision*, 11:147–164, 1993.

[15] F. Du and M. Brady. A four degree-of-freedom robot head for active vision. *International Journal of Pattern Recognition and Artificial Intelligence*, 8:1439–1470, 1994.

[16] J. Dias, C. Paredes, I. Fonseca, H. Araujo, J. Batista, and A. de Almeida. Simulating Pursuit with Machines. In *Proc. IEEE Int. Conf. on Robotics and Automation.* Nagoya, Japan, May 21-27, 1995.

[17] J.C. Fiala, R. Lumia, K.J. Roberts, and A.J. Wavering. TRICLOPS: A tool for studying active vision. *International Journal of Computer Vision*, 12:231–250, 1994.

[18] N. Ferrier and J. Clark. The Harvard binocular head. *International Journal of Pattern Recognition and Artificial Intelligence*, 7:9–31, 1993.

[19] P.J. Burt, J.R. Bergen, R. Hingorani, R. Kolczynski, W.A. Lee, A. Leung, J. Lubin, and H. Shvaytzer. Object tracking with a moving camera. In *WVM*, pp. 2–12, WVM89, 1989.

[20] M. Irani, B. Rousso, and S.Peleg. Detecting and tracking multiple moving objects using temporal integration. In *ECCV2*, pp. 282–287. ECCV92, 1992.

[21] M. Shizawa and K. Mase. Principle of superposition: A common computational framework for analysis of multiple motion. In *Proc. IEEE Workshop on Visual Motion*, pp. 164–172. Princeton, NJ, Oct. 7-9, 1991.

[22] P. Nesi. Variational approach to optical flow estimation managing discontinuities. *Image and Vision Computing*, 11:419–439, 1993.

[23] K. Hashimoto and H. Kimura. LQ optimal and nonlinear approaches to visual servoing. In K. Hashimoto, editor, *Visual Servoing*, pp. 165–198. World Scientific, Singapure, 1993.

[24] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. Robotics and Automation*, RA-8:313–326, 1992.

[25] J.T. Feddema, C.S.G. Lee, and O.R. Mitchell. Model-based visual feedback control for a hand-eye coordinated robotic system. *IEEE Computer*, 25(8):21–33, 1992.

[26] N.P. Papanikolopoulos, P.K. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Trans. Robotics and Automation*, pp. 14–35, 1993.

[27] P.K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. Robotics and Automation*, 9:152–165, 1993.

[28] G.D. Hager, Chang W.-C, and A.S. Morse. Robot hand-eye coordination based on stereo vision. *IEEE Control Systems Magazine*, pp. 30–39, Feb. 1995.

[29] O. Faugeras. *Three-dimensional Computer Vision.* MIT-Press, Cambridge, MA, 1993.

[30] T. Vieville. Auto-calibration of visual sensor parameters on a robotic head. *Image and Vision Computing*, 12:227–237., 1994.

[31] M. Li. Camera calibration of a head-eye system for active vision. In *Proc. Third European Conference on Computer Vision*, pp. 543–554, Stockholm, Sweden, May 2-6, J.O. Eklundh (Ed.), Springer LNCS 800, 1994.

[32] D.J. Fleet and K. Langley. Recursive filters for optical flow. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17:61–67, 1995.

[33] M. Hashimoto and J. Sklansky. Multiple-order derivatives for detecting local image characteristics. *Computer Vision, Graphics, and Image Processing*, 39:28–55, 1987.

[34] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–78, 1994.

[35] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understandig Workshop*, pp. 121–130, 1981.

[36] G.F. Franklin, J.D. Powell, and M.L. Workman. *Digital Control of Dynamic Systems.* Addison–Wesley, 1992.

[37] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association.* Academic Press, New York, NY, 1988.