Real-Time Vergence and Binocular Gaze Control

Jorge Batista Paulo Peixoto Helder Araújo

ISR - Institute of Systems and Robotics
Dep. of Electrical Engineering - University of Coimbra
3000 COIMBRA - PORTUGAL
batista,peixoto,helder@isr.uc.pt

Abstract

In this paper we describe the implementation of real-time binocular gaze control. This implementation is performed by using a complex binocular active vision system. In addition real-time vergence is also implemented. Vergence and gaze control are implemented based on binocular optical flow. Vergence control is implemented based on optical flow disparity, using the horopter as a reference. A kalman filter is used to filter out the signals and compensate for the delays of the system.

1 Introduction

The full recovery of the 3D structure (shape, location and other properties) of the world has proved to be an extremely difficult task. In addition most of the mathematical problems involved in such 3D reconstruction are ill-posed. For the vast majority of vision problems all 3D information is unnecessary. Biological vision systems prove that Nature tailored these systems to the particular needs of each animal.

Instead of trying to find general solutions for the vision modules we can consider the problem of vision in terms of an agent that sees and acts in its environment ([1], [2]). An agent can be defined as a set of intentions (or purposes) which translate into a set of behaviors [3]. The visual system can then be considered as a set of processes working in a cooperative manner to achieve various behaviors ([6], [7]). This is a paradigm known as active/purposive vision. Within this framework we consider that the system is active because it has control over the image acquisition process and acquires images that are relevant for what it intends to do. The control over the image acquisition process enables the introduction of constraints that facilitate the extraction of information about the scene [2]. Therefore our goal when using the active vision system is not the construction of a general purpose description. The system only needs to recover partial information about the scene. The information to be extracted and its representation have to be determined from the tasks the system has to carry out (its purpose). Vision is considered as part of a complex system that interacts with the environment [4]. Since only part of the information contained in the images needs to be extracted, the visual system will operate based on a restricted set of behaviors (sets of perceptions and actions).

By using these principles we have implemented some of the visual behaviors that can be found in the operation of the human visual system. These behaviors prove to be extremely useful in performing pursuit and tracking of moving objects. They simplify the infomation extraction process and facilitate the control aspects of the task. Pursuit of moving objects is important for tasks such as surveillance and for the control of mobile robots.

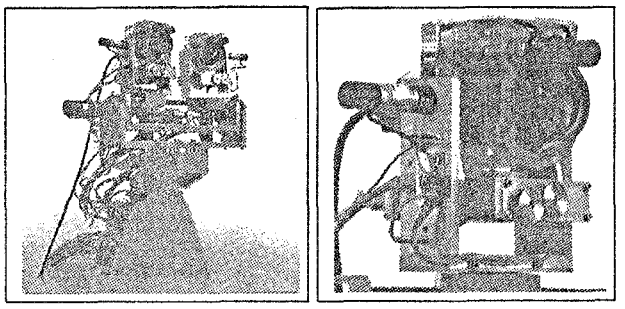

Figure 1: MDOF Active Vision Head and MDOF Eye with the MDOF Motorized Zoom lens

2 Related Work

In order to experiment with visual behaviors and to study active vision issues (inspired by biological implementations and in particular by the human visual system) we decided to build a multi-degrees of freedom (MDOF) robot head [8]. We call it the MDOF active vision head (see fig. 1). The MDOF active vision robot head is controlled by one host computer with a Pentium CPU (166Mhz) and a PCI Matrox Meteor frame-grabber PC board.

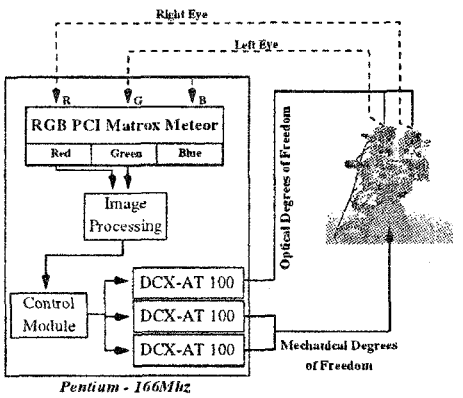

Figure 2: The MDOF system Architecture

Other groups have built heads and demonstrated them in several applications. At the University of Rochester [12, 15] a binocular head was demonstrated and tracking was performed using vergence control by means of zero-disparity filtering. A complex head was also built at

Proc. IROS 97 0-7803-4119-8/97/$10©1997 IEEE

KTH [10] using stepper motors. At the University of Oxford a head was also used [13] to demonstrate the use of image motion to drive saccade and pursuit.

To demonstrate the architecture, we have implemented a *Real-Time Vergence and Binocular Gaze Control.*

2.1 Fixation

Generally, fixation is achieved by a gross fixation followed by some fine adjustments. The fixation process is determined by the information available as follows:

- **Target positions on both of the retinas.** In this situation, the information is sufficient to achieve simultaneous fixation of both cameras.
- **Target position on one retina and the retinal disparity.** This situation is similar to the previous. The only difference is that there is no redundant information for simultaneous fixation. The fixation is verified after the initial fixation process.
- **Target position on one retina.** This may be a very common situation in practice. For the information available, we can only get the view direction of the target in space. Simultaneous fixation by both cameras is impossible. In this situation the best we can do is to plan a head motion such that *the efforts of the system to achieve a perfect fixation in the following vergence is minimized* or *the probability of obtaining a perfect fixation is maximized.*

Another solution to the fixation problem is obtained when the depth of the target is estimated by some other means, such as depth from defocus, or some other precalibration of the camera. Considering this information and the actual state of the head joints, fixation can be achieved.

We performed the fixation process using gross fixation solutions, defined as saccades, followed by a fine adjustments in both eyes in order to achieve vergence. The saccade process starts when one of the cameras detects peripheral motion. Depending on the information available, the fixation process performs different tasks. If the target is detected in both retinas, a neck (pan and tilt) and eyes saccadic movement is started to gaze the head into the target. However, if the moving target is only detected in one of the retinas, a two-stage fixation process is used:

- Before starting any saccadic movement a target search is performed with the eye that does not detected the moving target. For this search procedure two different situations must be considered, depending on the vergence of the eyes:
  1. If the eyes are verged far away the left eye always detect first targets moving from its left as well as the right eye that always detect first targets moving from its right. In this case the search direction is the same as the eye saccadic movement[1] (see fig. 3 a)).
  2. However, if the eyes are verged to a closer distance, the conclusions stated above are not always valid and either the left or right eye can detect moving targets coming from its right or from its left. In this case two different search direction must be considered. If motion is detected in the inner field of view of the head, then the search direction is the opposite direction of the eye saccadic movement. If motion is detected in the outer field of view of the head, then the search direction is the same as the eye saccadic movement (see fig. 3 b)).
- Having the target information on both retinas, redirecting gaze can be regarded as controlling the geometry of the head so that the images of the target will fall into the fovea after the fixation. Because of the rapid movement of the MDOF head joints during saccades, visual processing cannot be used. Therefore, saccades must be accomplished at the maximum speed or in the minimum possible time. Because the MDOF head has redundant degrees of freedom, motion planning is under-constrained. This means that the fixation can be achieved by an infinite number of possible motion trajectories. To guarantee a unique solution, an artificial constraint can be enforced:
  1. The saccade is planned so that the head is at the best state for the next action;
  2. Use only the motion of the lightest joints;
  3. Use all the joint motions simultaneously to achieve minimum time saccades.

The first solution was the one we adopted, moving neck and eyes in order to achieve symmetric vergence and having the cyclopean eye looking forward to the target.

The detection of motion is performed computing the image motion flow. In order to be able to detect the moving

[1]The eye saccadic movement relates to the eye that first detects peripheral motion

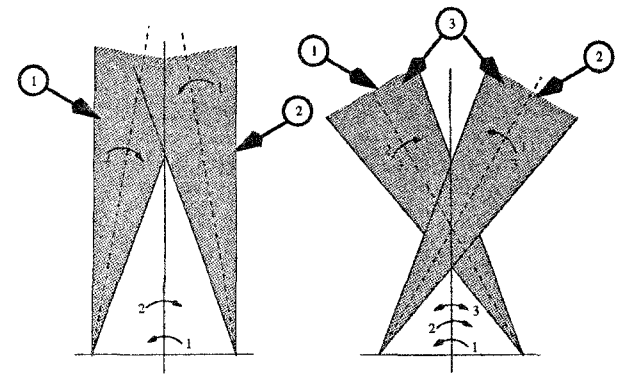

Figure 3: Two different schemes to achieve fixation depending on the vergence of the eyes. a) The search direction is always the same as the eye saccadic movement b) Depending on the eyes vergence, the search direction can be the opposite direction of the eye saccadic movement.

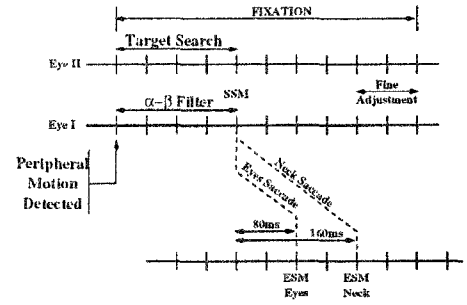

Figure 4: Timing diagram for the fixation process

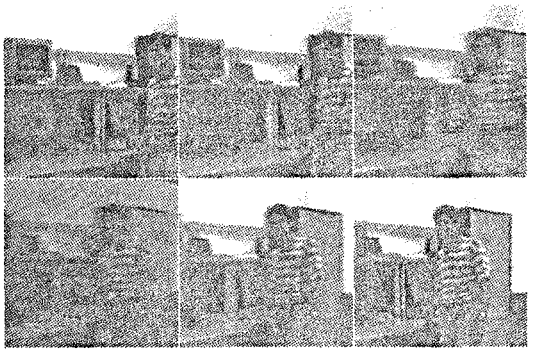

Figure 5: Image sequence obtained during the neck and eye saccade. A latency of 200ms (5 frames) exists for the saccade movement.

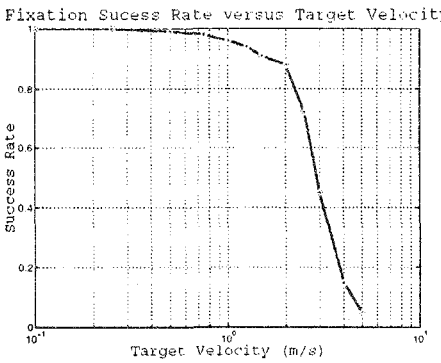

Figure 6: Performance of the fixation process as function of the target velocity.

target during the search target procedure, the image motion induced by the eye egomotion must be subtracted from the overal image motion flow (see subsection 2.2). The center of mass of the moving target is computed and its pixel coordinates converted into rotation angles that the neck or the eye must rotate to foveate on the origin of motion.

Due to the latency of the saccade movement (200ms for neck-saccade and 100ms for eye-saccade) an $\alpha - \beta$ filter is used to predict the image position of the target assuming that the target is moving with constant velocity (see fig. 4).

Saccade motion is performed by means of position control of all degrees of freedom involved. During the saccade period, and since the camera is moving at high velocity no visual feedback information is processed.

In order to achieve perfect gaze of both eyes in the moving target, and since the center of mass is probably not the same in both retinas for non rigid objects, after the saccade a fine fixation adjustment is performed. A grey level cross-correlation tracker is used to achieve perfect fixation of both eyes. With this procedure the image of the target on the retinas is near the center of the fovea (see fig. 5). The performance of the proposed fixation process as function of the target velocity is presented on figure 6. For a target moving at a cyclopean depth of 5 meters, this fixation process perform well for a maximum target velocity of around $2 meters/sec$.

2.2 Smooth Pursuit Using Optical Flow

During this process, the motion of the head must satisfy two basic requirements:

1. stabilize the images of the target on both retinas;
2. maintain fixation on the target.

A prerequisite to use pursuit planning is that the target is not far from the fixation point of the head (the images of the target on the retinas must not be far from the center of the foveal window). Otherwise, a saccade must be started prior to the smooth-pursuit process. This means that saccades have higher priority than pursuit.

After fixating on the target the pursuit process is started by computing the optical flow. During the pursuit process velocity control of the degrees of freedom is used instead of position control as in the case of the saccade.

Assuming that the moving object is inside the fovea after a saccade, the smooth pursuit process starts a Kalman filter estimator, which takes the estimated image motion velocity of the target as an input.

Using a **constant acceleration model**, the system state equation is $X_{k+1} = F_k \cdot X_k + V_k$ where V_k is noise, and

$$F_k = \begin{bmatrix} 1 & \delta_{t_k} & \delta_{t_k}^2 \\ 0 & 1 & \delta_{t_k} \\ 0 & 0 & 1 \end{bmatrix}$$

where $\delta_{t_k} = t_{k+1} - t_k$ and F_k is called the *state transfer matrix*. The state of the plant is $X_k = [x_k, v_k, a_k]^T$ at instante t_k, where x_k, v_k and a_k are position, velocity and acceleration respectively.

The real output of the plant (observation) at sample instant $t + 1$ is $Z_{k+1} = H \cdot X_{k+1} + W_{k+1}$ where $H = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$ is the measurement matrix, and W_{k+1} is the measurement noise at $(k+1)^{th}$ sample instant. V and W are independent and temporally uncorrelated.

Using the state transfer function of the Kalman filter, the current state of the plant can be predicted if the sample interval is not too large with respect to the motion of the target, simply using linear prediction.

$$X_{m|k} = \begin{bmatrix} 1 & \delta_{t_k} & \delta_{t_k}^2 \\ 0 & 1 & \delta_{t_k} \\ 0 & 0 & 1 \end{bmatrix} \cdot X_{k|k}$$

where $\delta_{t_k} = t_m - t_k$ and t_k is the latest sample instant and t_m is current time.

Using this approach, the smooth pursuit controller generates a new prediction of the current image target velocity, and this information is sent to the motion servo controller every 10ms (see fig. 7).

The smooth-pursuit controller assumes that the moving target is always located on the horopter and the cyclopean eye is pointing straight to the target. With this assumption, the motion induced on the retina by the moving target is almost the same on both eyes (see fig. 8). Two kalman filters were used to filter the estimated image motion velocities and the velocity used to control the neck pan and tilt joints was considered as the average value of both velocities (cyclopean eye velocity). Maintaining the target on the horopter is acomplished by the vergence process. To maintain tracking, the desired velocity of each of the neck joints should be $V_{des} = V + K_p \cdot \Delta_i$ where K_p is a gain matrix and Δ_i is the position error of the tracking point. The

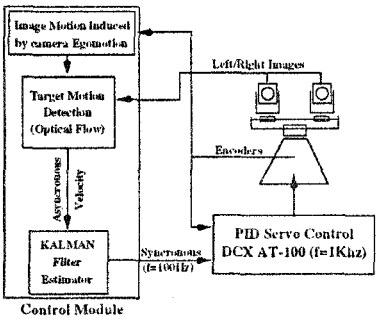

Figure 7: The MDOF High Level Gaze Controller

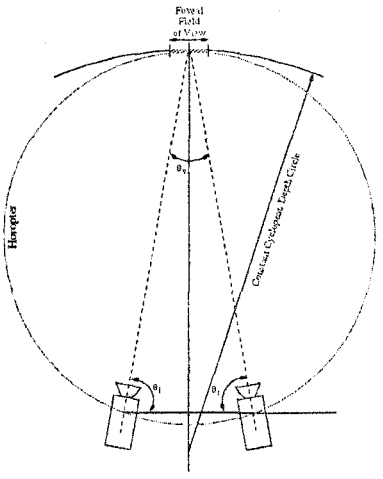

Figure 8: Fixation geometry for vergence and smooth-pursuit

primary function of the first term is to estabilize the images of the target on the retinas. The second term is used to enforce the positional constraint for dynamic fixation.

The Vergence Process

To fixate and verge on a target means to keep the images of the target on the image center (center of the fovea), that is, the positions of the target on the image plane must be ${}^{l/r}P_i = (0,0)$. If the image positions of the target on both eyes are known, the 3D position of the target can be recovered using the inverse kinematics of the head. Considering the constraint that both eyes must have equal vergence angles, and the positions of the target on the image plane must be ${}^{l/r}P_i = (0,0)$, the neck and eyes joints rotation angles can be computed based on ${}^{l/r}P_i = {}^{l/r}T_b \cdot P_b$, where P_b represents the target position in the head base coordinate system, and ${}^{l/r}T_b$ represents the transformation matrix between the head base coordinate system and each one of the retina coordinate systems. Due to the particular geometry of the MDOF robot head, computing the joint rotation angles is not an easy task. To simplify this problem, some assumptions must be made, neglecting some of the translation components of the head kinematics, and considering that the target is not moving too much close of the cyclopean eye. Based on the fact that the baseline distance is relatively small compared to the target distance, and that the target if not far from the center of the fovea, if the target is moving along the horopter its location is almost the same in both retinas (see fig 9). For this particular situation, within the field of view of the foveal area, the horopter and the circle of equal cyclopean depth are almost coincident (see fig. 8), which means that targets moving along the horopter tain a constant cyclopean depth and no vergence control is required.

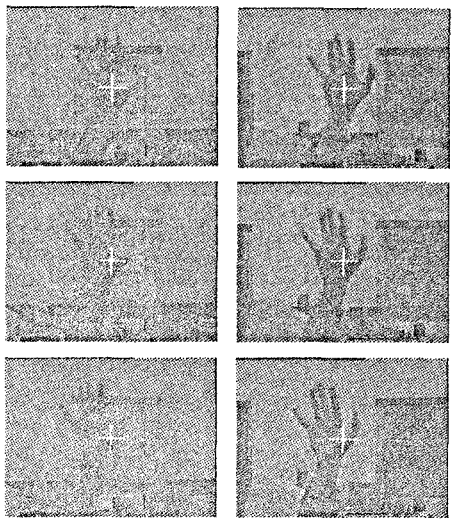

Figure 9: Retinal image disparity for targets moving along the horopter.

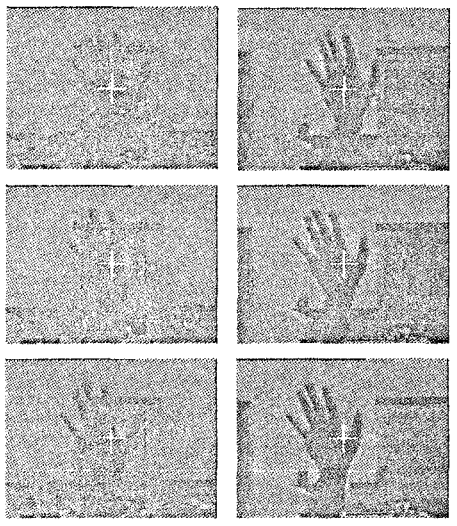

Figure 10: Retinal image disparity for targets moving outside the horopter.

However, if the target is moving outside the horopter, then its location on the retinas is no longer the same (see fig 10). This displacement is used to control the vergence angles and redirect the head gaze. Figures 11 and 12 show the filtered motion vector on both retinas and the retinal disparity, respectively for the case of a target moving along and outside the horopter[2]. The solution adopted to control the vergence of the MDOF robot head is based on two procedures running at different frequencies. The main process is optical flow based and the target motion detected

[2] On both figures, full line and dashed line correspond respectively to left and right retina motion, and dotted line represents the retinal disparity

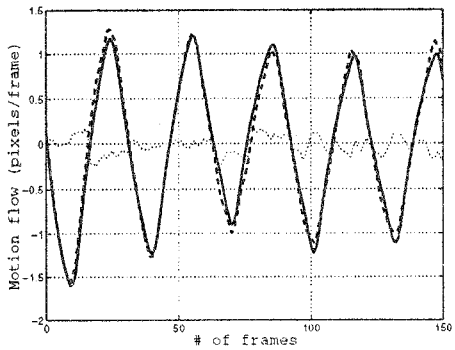

Figure 11: Motion flow and retinal disparity for targets moving along the horopter.

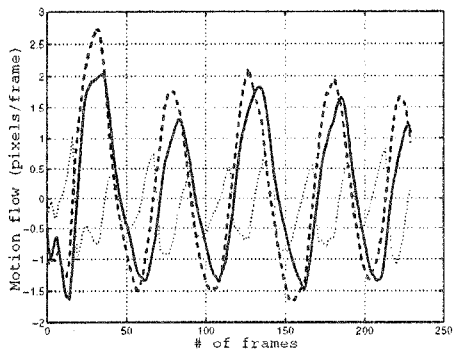

Figure 12: Motion flow and retinal disparity for targets moving outside the horopter.

between both retinas (retinal disparity) is used to control the eyes vergence.

Since this process only garantees that the target center of mass is located in the center of the fovea, we use a grey level cross correlation to adjust vergence and estimate the target depth, at a sample rate 10 times smaller (every 10th frame). The target depth is used to control the auto-focusing of both eyes, taking advantage of the pre-calibration of the focusing depth [9].

Target Motion on the Retinas

Unlike the motion of the target in 3D space, which is independent of the head joint motions, the target motions on the retinas are related to the motion of the head joints. Taking into account that the target motions on the retinas caused by the joint motion are generally faster than that caused by the target motion in space, we can not ignore the effects of joint motions on the target motions on the retinas.

We considered the analysis of motion described by the two-component model proposed in [14]. Two different motions must be considered to exist in the scene: one caused by the motion being undertaken by the head and the second one coming from the object. Since the first one is known (through inverse kinematics of the head), we only have to compute the latter. Let $I(x, y, t)$ be the observed gray scale image at time t. We are assuming that the image is formed by the combination of two distinct image patterns, P(the background motion) and Q (the target motion), having independent motions of p and q:

$$I(x,y,0) = P(x,y) \odot Q(x,y) \qquad I(x,y,t) = P^{tp} \odot Q^{tq} \quad (1)$$

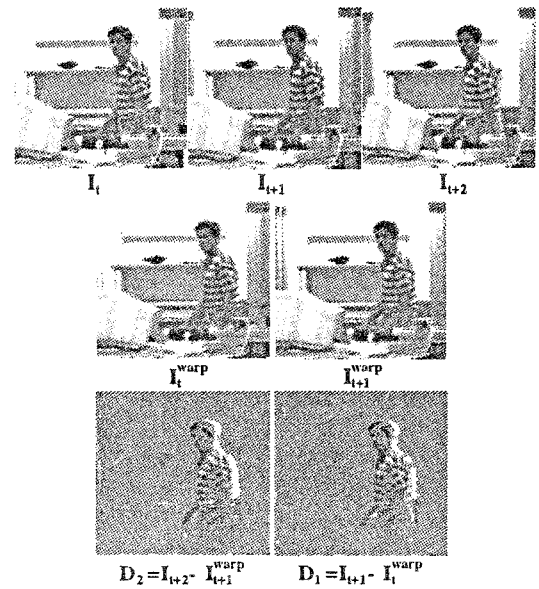

Figure 13: Target motion detection subtracting the image motion induced by camera movement.

where the symbol $\odot$ represents an operator such as addition, multiplication or even a more complex operator to combine the two patterns. In configurations where a foreground object moves in front of a moving background, the combination operator must be more complex, representing the asymmetric relationship between foreground and background. However, since the motion q of the object in the image is in fact $q = p + q'$ where q' is the real image object motion, we can consider that the $\odot$ operator can be approximately additive. Since motion p is known, only q must be determined. The pattern component P moving at velocity p can be removed from the image sequence by shifting each image frame by p and subtracting it from the following frame. The resulting sequence will contain only patterns moving with velocity q (see fig. 13). Let D_1 and D_2 be the first two frames of this difference sequence, obtained from three original frames.

$$\begin{aligned} D_1 &\equiv I(x,y,t+1) - I^p(x,y,t) \\ &= (P^{2p} + Q^{2q}) - (P^{2p} + Q^{q+p}) = Q^{2q} - Q^{q+p} \\ &= (Q^q - Q^p)^q \\ D_2 &\equiv I(x,y,t+2) - I^p(x,y,t+1) \\ &= (P^{3p} + Q^{3q}) - (P^{3p} + Q^{2q+p}) = Q^{3q} - Q^{2q+p} \\ &= (Q^q - Q^p)^{2q} \end{aligned} \quad (2)$$

As we can see the sequence of difference images is undergoing a single motion q. That means that it can be computed using any single-motion estimation technique.

In our case we model image formation by means of the scaled orthographic projection. Even if we model image formation as a perspective projection this is a reasonable assumption since motion will be computed near the origin of the image coordinate system (in a small area around the center x and y are close to zero). We can therefore assume that the optical flow vector is approximately constant throughout all the image, i.e., $u = p_x$ and $v = p_y$.

To compute the optical flow vector we minimize

$$\sum_i (I_{x_i} p_x + I_{y_i} p_y + I_{t_i})^2 = 0 \quad (3)$$

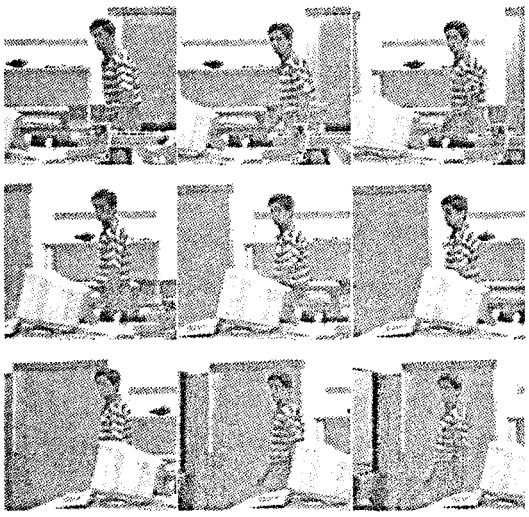

Figure 14: Pursuit sequence. Stack of images obtained during the smooth-pursuit process. The δt between consecutive images is $200ms$ (5frames).

Taking the partial derivatives on p_x and on p_y and making them equal to zero we obtain:

$$(\sum_i I_{x_i}^2)p_x + (\sum_i I_{y_i} I_{x_i})p_y + (\sum_i I_{x_i} I_{t_i}) = 0 \quad (4)$$

$$(\sum_i I_{y_i} I_{x_i})p_x + (\sum_i I_{y_i}^2)p_y + (\sum_i I_{y_i} I_{t_i}) = 0 \quad (5)$$

The flow is computed on a multiresolution structure. Four different resolutions are used: $16*16, 32*32, 64*64$. These are sub-sampled images. A first estimate is obtained at the lowest resolution level ($16*16$), and this estimate is propagated to the next resolution level, where a new estimate is computed and so on. The optical flow computed this way is used to control the angular velocity of the motors. The Fig. 14 shows images of the pursuit sequence (25 frames/sec were processed).

Background Image Motion Estimation

The image motion induced by the camera egomotion can be computed using the following equations

$$v_u = \frac{v_x \cdot f_x}{z} - \frac{v_z \cdot f_x \cdot x}{z^2} \qquad v_v = \frac{v_y \cdot f_y}{z} - \frac{v_z \cdot f_y \cdot y}{z^2} \quad (6)$$

where $(f_x,\ f_y)$ represents the focal length of the lens in pixels and $V = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$ represent the velocity of the point $P = \begin{bmatrix} x & y & z \end{bmatrix}^T$ in the camera coordinate system due to the egomotion of the head.

This velocity V results from the combination of several joints rotation (eye, tilt, swing and pan) and is defined as: $V = V_{eye} + V_{tilt} + V_{swing} + V_{pan}$.

Representing the velocity induced by each joint by $V_{ref} = -V_{Trans} - \Omega \wedge P$ and since each joint only performs rotations ($V_{Trans} = 0$), the velocity induced by each joint can be expressed as $V_{ref} = -\Omega \wedge P$.

Assuming the following angular velocities for each of the rotation joints $\Omega_{eye} = \begin{bmatrix} \Omega_e & \Omega_v & 0 \end{bmatrix}^T$, $\Omega_{tilt} = \begin{bmatrix} \Omega_t & 0 & 0 \end{bmatrix}^T$, $\Omega_{swing} = \begin{bmatrix} 0 & 0 & \Omega_s \end{bmatrix}^T$ and $\Omega_{pan} = \begin{bmatrix} 0 & \Omega_p & 0 \end{bmatrix}^T$ the velocities induced by the rotation of each joint are

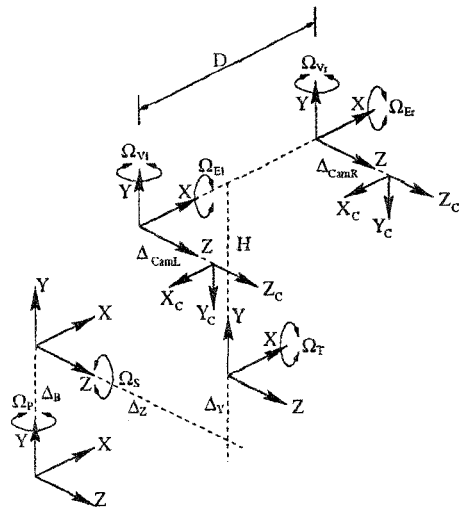

Figure 15: Joints coordinate systems of the MDOF head.

$$V_{eye} = -^{cam}T_{eye} \cdot (-\Omega_{eye} \wedge P_{eye}) \quad (7)$$

$$V_{tilt} = -^{cam}T_{tilt} \cdot (-\Omega_{tilt} \wedge P_{tilt}) \quad (8)$$

$$V_{swing} = -^{cam}T_{swing} \cdot (-\Omega_{swing} \wedge P_{swing}) \quad (9)$$

$$V_{pan} = -^{cam}T_{pan} \cdot (-\Omega_{pan} \wedge P_{pan}) \quad (10)$$

where P_{eye}, P_{tilt}, P_{swing} and P_{pan} represents the coordinates of the point P in each one of the joints coordinate systems (see fig. 15).

2.3 The Global Gaze Controller Strategy

The strategy adopted by the gaze controller to combine saccade, smooth pursuit and vergence to track moving objects using the MDOF robot head was based on a *State Transition System*. This controller defines five different states : *Waiting, Fixation, Pursuit, Vergence and μSaccade*. Each one of these states receives control commands from the previous state, and triggers the next state in a closed loop transition system. The global configuration of this state transition system is show in figure 16.

The distances of the target to the center of the foveal windows on the retinas are well described by the view direction difference between head fixation point and the target in space. Suppose the minimum view angle difference is α_{min} to guarantee that foveal image processing can still yield reliable results. If $\alpha - \alpha_{min} > 0$ then a μsaccade motion planning is activated. Otherwise, smooth-pursuit is used. During the μsaccade motion planning, no visual information is processed, and we use the Kalman filter estimator to predict the position and velocity of the target after a μsaccade.

Due to the architecture of our system we can change the control parameters on the fly so that the system can adapt itself to changes in velocity. This way the system can cope with sudden changes in velocity. We can also switch from velocity control to position control on the fly. The model used to compute the optical flow enables the system to cope with translations that do not have a component along the optical axis. To handle translations towards the

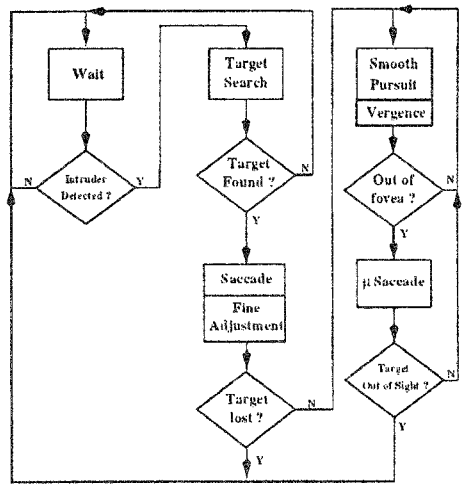

Figure 16: State Transition System.

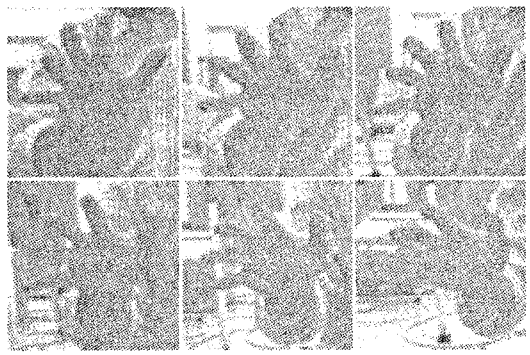

Figure 17: Stack of images obtained during the smooth-pursuit process with vergence control. The δt between consecutive images is $200ms$ (5frames).

head we analyze the vergence motions of both cameras. Even in the case where the target moves directly towards one of the cameras the optical flow induced in the other image originates a vergence motion. Basically taking into account that the object is fixated by both cameras the system is able to cope with the possible type of motions.

3 Conclusions

In this paper we have shown that by using the concept of *purposive behavior* it is possible to implement real-time active vision systems. The concept is essential for the design of the system architecture, if real time operation and robustness are major design goals. Another result of our approach is that the control architecture we have used enabled real-time operation with limited computing. On the other hand the use of parallelism enabled us the continuous processing of the image data as well as the coordination of the several actuation systems that have to work in synchrony. Parallelism is also essential to allow the visual agents to attend to the several events that are happening in the world continuously. The integration, the system architecture, the information processing modules, and the motor control processes were all designed taking into account the tasks and behavior of the systems.

References

[1] Aloimonos, Y.: Purposive and qualitative active vision. *Proc. Image Understanding Workshop* (1990) 816–828.

[2] Aloimonos, Y., Weiss, Y., Bandopadhay, A.: Active Vision. *Intern. J. Comput. Vision* **7** (1988) 333–356

[3] McFarland, D., Bosser, T.: *Intelligent Behavior in Animals and Robots.* MIT Press (1993)

[4] Aloimonos, Y.: What I Have Learned. *CVGIP: Image Understanding* **60** No.1 (1994) 74–85

[5] Ho, Y.: Dynamics of Discrete Event Systems. *Proc. IEEE* **77** (1989) 3–7

[6] Sloman, A.: On Designing a Visual System. *J. Exper. and Theor. Artif. Intell.* **1** (1989) 289–337

[7] Horridge, G.: The evolution of visual processing and the construction of seeing systems. *Proc. Roy. Soc. London B* **230** (1987) 279–292

[8] Batista, J., Dias, J., Araújo, H., Almeida, A., The ISR Multi-Degrees-of-Freedom Active Vision Robot Head: design and calibration. *M2VIP'95–Second International Conference on Mechatronics and Machine Vision in Practice* Hong-Kong, September, (1995)

[9] Batista, J., Peixoto, P. Araújo, H., Real-Time Visual Behaviors with a Binocular Active Vision System. *ICRA97 - IEEE Int. Conf. on Robotics and Automation.* Albuquerque, New Mexico, USA, April, (1997)

[10] Pahlavan, K.: Active Robot Vision and Primary Ocular Processes. *PhD Thesis*, CVAP, KTH, (1993), Sweden.

[11] Dias, J., Paredes, C., Fonseca, I., Batista, J., Araújo, H., Almeida, A.: Simulating Pursuit with Machines: Experiments with Robots and Artificial Vision. *Proc. IEEE Int. Conf. on Rob. and Auto.* 472–477 May 21–27, (1995), Nagoya, Japan

[12] Brown,C.: Gaze controls with interaction and delays. *IEEE Transactions on Systems, Man and Cybernetics* **20**, May, (1990), 518–527.

[13] Murray D., Bradshaw K., MacLauchlan P., Reid I., Sharkey P.: Driving Saccade to Pursuit Using Image Motion. *Intern. Journal of Computer Vision* **16**, No.3, November, (1995), 205–228.

[14] Bergen, J., Burt, P., Hingorani, R., Peleg. S.: Computing Two Motion from Three Frames In *David Sarnoff Research Center* April, (1990).

[15] Brown,C., Coombs D.: Real-Time Binocular Smooth Pursuit *Intern. Journal of Computer Vision* **11**, No.2, October, (1993), 147–165.

[16] Almeida, A., Araújo, H., Nunes, U., Dias, J.: Multi-Sensor Integration for Mobile Robot Navigation. In *Artificial Intelligence in Industrial Decision Making, Control and Automation* Spyros Tzafestas and Henk Verbruggen (Eds.), Kluwer Academic Publishers, (1995).

[17] Burt, P., Bergen, J., Hingorani, R., Kolczynski, R., Lee, W., Leung, A., Lubin, J., Shvaytser, H.: Object tracking with a moving camera. *Proc. IEEE Workshop Visual Motion*, Irvine, (1989).

[18] Willson, R.: Modelling and Calibration of Automated Zoom Lenses. *CMU-RI-TR-94-03*, Carnegie Mellon University, (1994).

[19] Carpenter, R. H. S.: Movements of the Eye. Pion, (1988).