Control of Visual Attention in Mobile Robots *

James J. Clark
Nicola J. Ferrier

Division of Applied Sciences
Harvard University

Abstract

A vision system for use in a mobile robot system requires attentive control. Attentive control refers to the process by which the direction of gaze of the visual sensors are determined, along with the determination of what processing is required to be applied to the sensed images based on the goals of the robot and the tasks it is performing.

This paper describes the implementation of a motion control system which allows the attentive control of a binocular vision system for use in a mobile robot. Attentive inputs to the system specify the type of visual feedback that the oculo-motor control system will use. The MDL language developed by Brockett [8] is used to communicate between the attentive planner and the motion controller.

1 Introduction: Attentive Vision for Mobile Robots

In complex robotic activities such as those encountered in mobile robot navigation, different actions must be performed at different times. Thus we require a control system that allows for the changing of parameters of the control system in order to allow the carrying out of the various desired operations. A vision system for use in such complex robotic activities will be required to change the field of view in order to best accommodate the task at hand. The environment of a mobile robot changes as the robot moves. The goals of the robot will typically give it an idea of what it should be looking for. The robot will want to direct its attention to objects pertinent to its task. For example, in navigating a hallway, a robot will want to detect the orientation of the hallway, identify obstructions, possibly identify doorways, and generally ignore extraneous objects, such as pictures, which are not important to the task at hand. Burt[7] describes *active sensing* (or "smart" sensing) as the selective, task oriented gathering of information. One focusses the "attention" of the visual system on a portion of the scene that is important to the task at hand. As the demands of the robotic task evolve this focus of attention may shift. Bajcsy [2,3] extends the concept of active perception to include the presence of feedback. In this extension, information obtained through the visual process, both high and low level, is used to control the data acquisition process.

The extraction of the salient information and redirection of computational resources to those areas will improve the efficiency of a visual system. Changing of the focus of attention can refer to changes in the spatial region in the scene upon which our visual system is concentrating (or "foveating" at a high resolution). Much of the information in the field of view is not needed to perform many tasks. Visual tasks require the movement of the eyes to closely examine areas of interest for the particular task, paying little attention to the rest of the scene which is viewed peripherally. Since we want to devote computational power only to regions containing salient information, these regions must be identified and then foveated. As the environment changes we want attention to move so that we are constantly acquiring the regions with salient information in the fovea. Foveation can be accomplished by mechanically adjusting the direction of view of the image sensor or it can be accomplished by moving a processing window about in an internal representation of the image (e.g. see the multiresolution foveator of Burt [7]). This mode of attention forms the basis for Ullman's visual routine paradigm [18], in which sequences of elementary image analysis operations are performed to obtain properties of, and relations between objects, in a scene. Focus of attention may also refer to the selection of a given set of image processing operations that are to be used to extract information from the scene. For example, a given visual task may require that corners of objects be detected, while another visual task may require that the colour of objects be determined. In each of these two cases different features would be attended to.

We can define *attentive visual control* as the process by which the desired direction of gaze of the visual sensors are determined and attained, along with the determination of what processing is required to be applied to the sensed images based on the goals of the robot and the tasks it is performing. Attentive control is important for mobile robots as typical mobile robotic activities are task and goal based and the robot will be driven by these tasks.

Attentive vision control can be divided into two subproblems: 1) deciding where in the visual scene to attend on and 2) deciding which motions are required to redirect the visual sensors toward that location. The second problem, one of oculomotor control, will be briefly discussed; a fuller treatment of that subject can be found in [8]. The first problem depends on the task at hand and the visual environment. The location to attend to varies with the task and will shift as the task proceeds. The desired location should contain the information in the scene which is *salient* to the task.

This paper presents a model of attention whereby the visual feedback pathways which control the direction of camera gaze based on visual input, have adjustable gains. The most salient feature, with respect to this weighting of the visual feedback paths, is found and centered on the field of view (via movement of the vision sensors). Our model of attention can be put into the modal control paradigm proposed by Brockett[5], which uses a device independent modal description of motion. Modal control is adaptive and is thus useful for mobile robots. In the modal control paradigm, attentive behaviour can be specified device independently, in a manner which can be used in unpredictable environments. The modal description uses selective visual feedback, based on a high level description of the sequence of foci of attention or "modes" of activity, to control the position and velocity of the visual sensors (cameras) in a binocular image acquisition system. The sequences of foci of attention would depend on the particular task the robot is performing.

***This research was supported in part by the Harvard-MIT-Brown Center for Intelligent Control Systems (US Army Research Office under grant DAAl03-86-K-0171). and by the University of Maryland/Harvard System Research Center (National Science Foundation under grant CDR-85-00108)**

We have built a system incorporating this model of attention in the modal paradigm. The system allows the specification of attentive behaviour of an oculomotor system at a high level, independent of the specific actuators used in the system. Experiments performed with this system will be presented.

2 The MDL Motion Description Language

This section briefly reviews the motion description language introduced by Brockett[5]. This language allows the motion of a mechanical system to be specified to a desired level of precision in a device-independent manner. adaptable to unpredictable environmental conditions.

One can describe the control of a mechanical system through a differential equation relating the effect of control inputs to the state of the system. Based on this, Brockett [5] proposed an MDL (for a Motion Description Language) device which would accept the open loop controls u and the feedback processing functions k and produce the correct actuator signals which would force the state vector $x(t)$ (typically position of motor shafts) to be a solution of the equation:

$$\dot{x} = f(x(t)) + G(x(t))(u(t) + k(y(t)) \quad (1)$$

where $y(t) = h(x(t))$ is a vector of measurements of the state. In order to perform different actions at different times, we will want a control system that allows for the changing of the user definable parameters of the control system. As we have seen, the important user definable parameters are the setpoints u, and the feedback selection functions k. The MDL device of Brockett consumes (u, k, T) triples which specify the adaptive nature of the control. Each (u, k, T) triple describes the type of control law that is to be used over the epoch T. The time T can be specified *a priori* or can be non-deterministic, such as provided by a stopping or transition rule. Thus, in order to produce a given complex motion, one would supply a string of (u, k, T) triples to the MDL controller. Brockett [5] refers to these strings as *modes*. One could store a number of modes, each of which corresponds to a certain complex motion, in a table where they would be available for accessing when required. These modes could be hardwired, or they could be learned through some optimization process (training and practice). A control system using motions defined as modes, that are input to an MDL controller can be called a modal control system. Note that the modes are described at a high level, and hence the modal definition of a complex motion is "device independent". Only the MDL interpreter, which converts the (u, k, T) strings into actuator signals, need be designed for each mechanical system. Such an interpreter has been built for a four degree of freedom planar gripper[11,4].

The k function can be thought of as a generalized compliance. For example consider the case where x is a position of some kind, and y is a sensed force such as in the hybrid position and force control used in some robotic manipulators [16]. Then k converts forces to changes in position. Thus the system acts as a spring with compliance (1/stiffness) k. If force components are sensed in different directions and different positional degrees of freedom are controlled by these sensed forces, then k is a matrix which relates the effect of a force in a given direction to a change in position in some other direction. This system is then a generalized spring system. If k is diagonal, then its elements determine the compliance of the system in different directions. Such a system could be more stiff in one direction than in another. In general, one may not have force sensors, or use position control. In such a case the k's will not represent compliances, but will still relate the effect of the individual sensory inputs on the control of the system. This is an important point as it shows that, by controlling the k's one can select different types of feedback mechanisms. This example shows how one can select between two types of feedback using the k functions. In section 4 we will extend this idea to the control of visual attention, wherein we change the values of the k's that select for different visual sensing operations in order to attend on a given scene element. In other words by adjusting the feedback weights applied to various feature detector outputs we force our oculomotor system to *comply* with different features in the environment in much the same way that hybrid force control allows a manipulator to comply with a bumpy surface.

In the rest of the paper we describe an MDL based implementation of an attentive vision system. This system will control the motion of a pair of cameras in such a way as to facilitate the execution of varying robotic tasks. The system that we are proposing is a dual level system. The first, or inner, level performs automatic camera movements based on set points and mode controls supplied by the outer level. (The inner level is a conventional actuator position/velocity control scheme.) The outer level sends (u, k, T) triples to the inner level based on a set of (u, k, T) triples provided by the user as input to the outer level. In this case the outer level triples represent a series of foci of attention. The outer level k's describe what *visual routines*, or modes, are to be applied to the binocular visual output (the $y(t)$'s) to compute the desired camera positions and to generate the control signals to move the cameras. Changes in attention are implemented by supplying the outer level motion control component with a new (u, k, T) triple. Visual routines which involve many shifts in attention are implemented by sending the controller a mode containing a string of (u, k, T) triples. The implementation of the outer level component is described in detail in section 4.

3 Hardware Implementation of a Mobile Binocular Camera System

To test our theories of control of visual attention in mobile robotic systems we have constructed a mobile binocular camera system. This is similar in spirit, if not in detail to systems constructed at UPENN [13], MIT [15] and Rochester [6]. The oculomotor control system of our binocular vision system is based on models of mammalian oculomotor control systems. This control system is fully described elsewhere [8]. The head, shown in figure 1, is a seven degree of freedom mechanism. Three of these degrees of freedom are associated with the orientation of the cameras, while the other four have to do with the state of the cameras' aperture and lens focus. The three mechanical degrees of freedom are: 1) Pan, which is a rotation of the inter-camera baseline about a vertical axis, 2) Tilt, which is a rotation of the inter-camera baseline about a horizontal axis, and 3) Vergence, which is an antisymmetric rotation of each camera about a vertical axis. With these three degrees of freedom one can theoretically place the intersection of the optical axes of the two cameras (what we will refer to as the fixation point) anywhere in the three dimensional volume about the head. In practice, the volume of accessible fixation points will be restricted due to the limited range of motions of the degrees of freedom.

The physical motion required to adjust the positions of cameras attached to a robot can be obtained in many ways depending on the mechanical structure of the robot. For example, if the robot is mobile and can move with three degrees of freedom (translation in x, and y and rotation about the z axis) in a plane, then the direction of view of a camera, fixed to the robot, can also be controlled with these three degrees of freedom. In general, however, it is more convenient to decouple the attitude of the camera(s) from the attitude of the body of the robot. This allows the camera to look in a given direction independently of the direction in which the robot is pointed. Furthermore, the time constants of a system that positions the camera alone will be, in general, much smaller than that of a system that positions the robot. So, by controlling the camera orientation independently of the orientation of the robot one, obtains an increase in flexibility and speed, over the case in which the camera orientation is rigidly coupled to that of the robot. The system that controls the cameras should, however have inputs from the system that controls the robot body, so that motions of the robot body can be compensated for by the camera system automatically, much like the human vestibular system.

The control scheme used to control the pan, tilt and vergence is based on the model of human oculomotor control described by Robinson [17] which includes three descriptive regimes: saccades, pursuit, and vergence. The first two control the position and velocity of the pan and tilt axes. The third controls the plane of fixed focus by adjusting the relative angle between the two cameras. We have implemented the control scheme that is depicted schematically in figure 2 for the Harvard head. The pan, tilt, and vergence motors are driven by a pulse width modulated MOSFET amplifier. The input to this amplifier is derived from the output of a Dynamation motor controller board [10]. The Dynamation board is indicated in figure 2 by the box taking in the shaft encoder position from the motor and which outputs a drive signal to the motor amplifier. The Dynamation board takes set point inputs over a VME bus connection to a SUN computer. These setpoints can either be position setpoints (in the case of vergence or a saccade) or velocity setpoints (in the case of pursuit). The Dynamation can output to the VME bus (and then on to the SUN computer) an efference copy of the current motor position. This efference copy is delayed, in the SUN computer, by a time equal to the time taken to perform visual feature localization, and added to the current position errors, determined by the visual feature localization process. The Dynamation board does not have a tachometer, so that an velocity efference copy is not available. Thus we generate one by differentiating the position efference copy. The sampling rate of the Dynamation board is very high (more than 1000 samples per second), however, so that this estimate of velocity should be accurate.

The feature detection and localization is performed in a special purpose image processing system, manufactured by Datacube [15]. This system can do image processing operations such as 8x8 convolution, histogramming, and logical neighborhood operations on a 512x512 pixel image at video rates (30 frames per second). Thus the latency per operation is 33 milliseconds. Most feature detection operations require more that one frame time however. In our initial experiments we implemented a feature detector that could detect black blobs or white blobs, in about 3 frame times. Therefore the latency of our feature detector was about 100 milliseconds. The Datacube system, after it detected the presence of a feature, would output the position and velocity of the feature over the VME bus to the SUN workstation. The SUN workstation then computes the quantities $\theta_{R_x} + \theta_{L_x}$, $\dot{\theta}_{R_x} + \dot{\theta}_{L_x}$, $\theta_{R_y} + \theta_{L_y}$, $\dot{\theta}_{R_y} + \dot{\theta}_{L_y}$, and $\theta_{R_x} - \theta_{L_x}$, where θ_{R_x} is the x component of the retinal disparity in the right camera, θ_{R_y} is the y component of the retinal disparity in the right camera, θ_{L_x} is the x component of the retinal disparity in the left camera, θ_{L_y} is the y component of the retinal disparity in the left camera, and $\dot{\theta}$ indicates a retinal velocity. The difference in the left and right x components of the retinal position is added to the delayed position efference copy of the vergence motor. Thus this difference will be driven to zero. The sum of the left and right retinal position errors in both the x and y directions are added to the delayed position efference copies of the pan and tilt motors respectively. This will, during a saccade, drive these sums to zero. Combined with the driving of the difference of the x retinal position errors to zero by the vergence, the result will be that the x and y retinal position errors in both cameras will be driven to zero, as desired. A saccade trigger signal (that opens up the sample/hold) is generated by the feature detection system when the retinal position error is greater than threshold value. During the saccade, visual processing is turned off to prevent saccades being generated while the saccadic motion is being performed.

During pursuit the sum over the two cameras in each of the x and y retinal velocity errors will be driven to zero. If the system has the correct vergence, then the x and y component of the retinal velocity error will be driven to zero in each eye, and not just the sum of the errors in the two eyes.

We have performed simple blob tracking experiments which show that the system operates as desired, in that the vergence and saccadic modes result in fixation of the feature as we move it about in space.

4 Modal Control of Attention

The inner level control loop described in the previous section is controlled by an outer loop which implements attentional shifts which result in movement of camera positions. This section will describe a model of visual attention and present this model in the modal paradigm.

The first stage in our visual attention model acquires the images and extracts visual "primitives" in parallel across the visual field. The results from this stage are a set of feature maps $y_i(x, y, t)$ which indicate the presence or absence of a feature at each location in the image. Simple feature maps may indicate the presence of a specific colour or line orientation for example. Complex feature maps may perform texture and figure-ground segmentation or implement inhibition from neighboring regions to compute which regions are different from their surroundings.

The next stage of the model combines the results from the feature maps, building a saliency map, $S(x, y, t)$. The output from the feature maps are "amplified" with different feedback "gains", $k_i(t)$ for each map y_i and then summed to form the saliency map, $S(x, y, t)$. The value of the map at each location is a numeric indicator of how "salient" is the information at that location. Hence finding the location with the maximum value will give the most salient location with respect to the given feedback gains, $k_i(t)$. As the notation indicates, these gains may vary over time, thus changing the location of the most salient feature. Figure 3 shows this attention model.

Adjusting the gains of a particular feature map will direct attentional resources to occurrences of that feature. A decaying gain function, $k(t)$, will decrease the saliency of a location over time and hence another location will become more salient and attention will change to a new location. Higher cognitive or planning levels can actively select which features to attend to by adjusting $k_i(t)$.

We have chosen to express our model of visual attention in the paradigm of the motion control language (MDL) described earlier. This paradigm allows a description of motion control of the head/eyes based on visual feedback which is "independent" of the underlying hardware or implementation. Using the MDL will allow a mechanism to control attention as a "high level language".

The gains of the inner feedback loop which is concerned with setpoint control of the head positioning motors remains constant, as the load on the head motors remain roughly constant. In principal, one need only determine the position feedback gains k once, such that the step response of the motor to the inner level setpoints is critically damped. These gains are set in the Dynamation controller board, which handles the inner level control loop. The sensory input to the inner level is the motor shaft position, measured with the shaft encoders. The velocity of the motor shafts are not measured directly but are computed from the position measurements through differentiation as described in the previous section. The inner control loop is switched between position control and velocity control by the outer control level. This is done, in effect by sending a (u, k, T) triple in which the k's decide which measurement (position or velocity) will be used to control the motor. The setpoints u that are input to the inner level control loop also come from the outer control loop in these (u, k, T) triples.

The k's in the (u, k, T) motion control system definitions concerned with the outer, visual, feedback loop will change due to changes in the focus of attention. The feedback selection process at this level is much more complicated than the inner level feedback selection in which only direct position or velocity feedback was being selected for. In the outer level, one still selects for position or velocity feedback but, in addition, one must select the feature(s) to be used to detect the scene element whose position or velocity is fed back. This feature selection is performed, in the MDL paradigm, by adjusting the weight we apply to a given feature in the control feedback loop. Note that all features are looked for in parallel, not just those we are attending to.

The outer control level consumes *modes* which allocate attention to specific features and produces different modes for the inner loop. The output modes consist of position and velocity setpoints and a time interval in which to apply these setpoints. The modes consumed by this second level are again of the form (u, k, T) where u is the desired position (always 0 for foveation – to center target on visual field), k is a vector which represents which features to detect (the feedback gains) and T is the time period in which the mode is to be applied.

In the language given earlier, $y(t)$ is the state measurement vector. In this case, $y(x, y, t)$ is a pair of images (left and right "eyes"). Referring to the model given earlier, $k(t) = (k_1(t), k_2(t), ..., k_n(t))$ is a vector containing the "weights" to be applied to the results from the primitive operations (feature maps). With these gains, the saliency map can be computed and the maximum found. The location of the maximum must then undergo a coordinate transform in order to obtain the setpoints in head coordinates. This transformation will depend on the camera parameters and the particular configuration of the "head" and hence can be absorbed in the $G(\cdot)$ term in equation (1).

Figure 4 shows the lowest two stages of the modal control. A mode, (u, k, T), which was generated at a higher level, is "fed" into the intermediate level (denoted M2). Over a time period, $0 \leq t \leq T$ the weights associated with the feature maps will be $k(t) = (k_1(t), k_2(t), ..., k_n(t))$. At each instant of time, t, a location (x, y) will be output as the "most salient feature" of the image. These positions are output to the inner loop (denoted M1) where they generate positional errors used to drive the head motors.

There are advantages in using the MDL description for the control of attention. The same description can be used with simple vision routines or with more complicated algorithms depending on the available hardware. The complexity of the feature maps used will determine what tasks can be performed. A large set of feature maps with maps at many scales detecting a large group of primitives will allow for sophisticated visual processing.

5 Experiments

Two experiments have been performed so far with our system to demonstrate modal control of attention.

The first experiment involved tracking "blobs", or regions having a specific range of intensity values. The features we used were black or white blobs against a neutral background. The task was to locate either the black or white feature and follow it. The objects were placed 0.5 to 2.0 meters from the head. The head was able to fixate on an object to within 2 pixels. The vision system for simple blob detecting tasks could process on the order of 5 frames per second. Taking into account the communication time between the vision system and the head control system, an overall rate of 3 frames per second could be achieved.

The second experiment was designed to demonstrate the attentive control system on a more complex scene. The features used are the 0^{th}, 1^{st} and 2^{nd} moments of each object and the intensity value. The scene is segmented into connected components, the various features are computed and the saliency map is built (as described in previous sections). Stereo correspondence is performed using the peak saliency values. As the task is to find the most salient feature with respect to the feature gains, k_i, the most salient points are the only ones that need to be considered in computing stereo correspondence. Since only a few points will be maximal (with well chosen gains), the correspondence problem is easily solved. With this done, the disparity values are computed and used to drive the head motors as described above. Using a combination of black and white, circular and rectangular objects, the attention system can successfully locate geometric shapes at different orientations and fixate on them. Altering the feature map gains, k_i, alters the direction of gaze to fixate on the object most salient with respect to the new gains. This experiment is much slower than simply distinguishing between a black and a white object. At present, depending on the complexity of the scene, the attentive system may between 1 to 10 seconds to fixate on the most salient object. The vision system is the culprit. This implementation uses a hybrid vision system employing both Suns and the Datacube and is not yet optimized. Future work to incorporate the entire vision system on the Datacube is already underway. Given that the vision system could work arbitrarily fast, the attentive control system is successful at tracking objects of interest.

6 Summary

We have described a control system for a binocular image acquisition mechanism, for use in mobile robotic systems, which allows shifts in focus of attention to be made in a natural, device independent manner. The control method is based on the modal control technique proposed by Brockett [5]. Shifts in focus of attention are accomplished by altering the feedback gains applied to the visual feedback paths in the position and velocity control loops of the binocular camera system. By altering these gains we can perform a feature selection operation, by which the *saliency*, in the sense of Koch and Ullman[12], of a given feature is enhanced, while the saliency of other features are reduced.

The control system that we have described in this system is a two level one. The first, or inner, level performs the direct control over the position and velocity of the motors attached to the cameras. This level is based on models of the human oculomotor control system. The outer level controls the focus of attention, in that it determines what features are going to be used in determining where to look next.

The advantages of using attentive vision control in a mobile robot application are that different actions can be performed depending on the foci of attention used. Attentive control works in an unpredictable environment, and the foci of attention can be changed in order to carry out various operations. Furthermore, there are some tasks which are naturally suited to attentive vision, and for which conventional vision systems find very difficult to perform. An example is object recognition. The ability to obtain multiple views, and multiple views that are intelligently selected, helps enormously in performing model based object recognition. One of the drawbacks of attentive vision has been the requirement that real-time image processing operations are necessary to maintain real-time operation. However, recent advances in image processing hardware, exemplified by Datacube's [9] Maxvideo system, and the Pipe system [14] produced by Aspex, have made it possible for researchers to perform dynamic image processing operations at video rates on sequences of images obtained from video cameras, so there are few practical reasons why vision systems for mobile robots should not use attentive vision techniques. The control system we have described in this paper will extend the abilities of active vision systems[1] in that it provides a method by which attentive behaviour can be conveniently obtained.

7 Acknowledgements

The Harvard Head project has been a collaboration involving many people. The mechanical component of the system was designed and constructed by J. Page, W. Labossier, and M. Cohn, with input from R. Brockett, J. Clark and E. Rak. M. Cohn should be singled out for the job he did in reverse engineering the Canon electronic focus controls. The electronics for the motor drivers and associated systems was put together by J. Page. Software for the motion control systems was written by N. Ferrier, V. Eng, M. Cohn, and P. Newman. Software for the visual processing modules was written by N. Ferrier with some assistance from M. Lee and E. Rak. Ideas and enthusiasm concerning the development of the head and its motion control were supplied in great abundance by R. Brockett.

References

[1] Aloimonos, Y., Weiss, I., Bandyopadhyay, A., "Active vision", Proceedings of the 1st IEEE Conference on Computer Vision, London, 1987, pp. 35-54

[2] Bajcsy, R., "Active perception vs. passive perception", Proceedings 3rd IEEE Workshop on Computer Vision, Bellaire, pp. 55-59, 1985

[3] Bajcsy, R., "Perception with feedback", in the Proceedings of the 1988 Darpa Image Understanding Workshop, pp. 279-288

[4] Bangs, A., "The Motion Interpreter/Control Computer System", Harvard University Undergraduate Thesis, 1988.

[5] Brockett, R.W., "On the computer control of movement", Proceedings of the 1988 IEEE Robotics and Automation Conference, Philadelphia

[6] Brown, C.M., "Progress in image understanding at the University of Rochester", in the Proceedings of the 1988 Darpa Image Understanding Workshop, pp. 73-77

[7] Burt, P., "Algorithms and architectures for smart sensing", in the Proceedings of the 1988 Darpa Image Understanding Workshop, pp. 139-153

[8] Clark, J., and Ferrier, N., "Modal Control of an Attentive Vision System", in the Proceedings of ICCV, Florida, December 1988.

[9] Maxvideo System Documentation, Datacube Inc., Peabody, MA

[10] Dynamation motor controller board documentation, Dynamation Inc., Mountain View, CA

[11] Eng, V., "Design and Implementation of a Motion Description Language", Harvard University Ph.D. Thesis, to appear.

[12] Koch, C., and Ullman, S., "Selecting one among the many: A simple network implementing shifts in visual attention", MIT AI Memo No. 770, January, 1984

[13] Krotkov, E., Fuma, F., and Summers, J, "An agile stereo camera system for flexible image acquisition", IEEE Journal of Robotics and Automation, Vol. 4, No. 1, pp. 108-113, 1988

[14] Pipe system documentation, Aspex Inc.,

[15] Poggio, T., et al, "The MIT vision machine", in the Proceedings of the 1988 Darpa Image Understanding Workshop, pp 177-198

[16] Raibert, M.H., and Craig, J.J., "Hybrid position/force control of manipulators", in **Robot Motion: Planning and Control**, Brady, M. et al editors, MIT Press, 1982

[17] Robinson, D.A., "Why visuomotor systems don't like negative feedback and how they avoid it", in **Vision, Brain and Cooperative Computation**, Arbib, M. and Hanson, A. eds., MIT Press, Cambridge, MA, 1987

[18] Ullman, S., "Visual routines", Cognition, Vol. 18, pp. 97-159, 1984

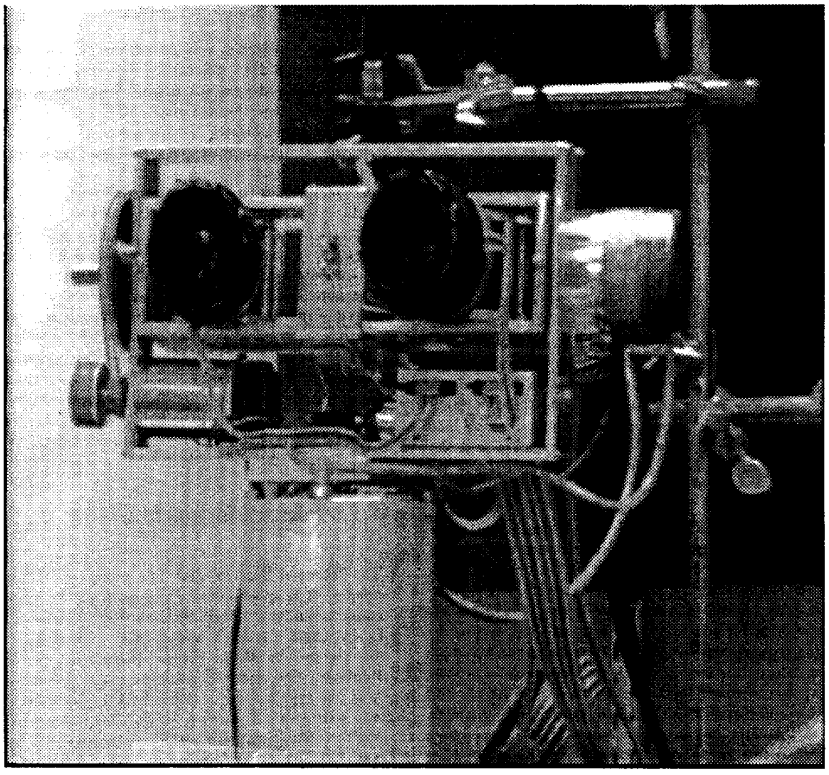

Figure 1: A photograph of the Harvard Head.

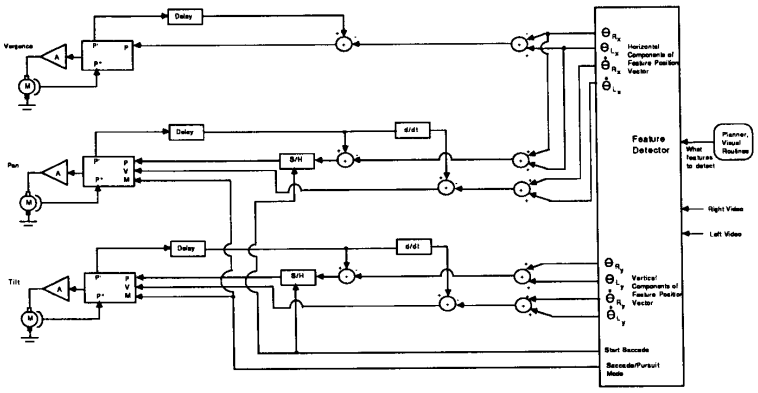

Figure 2: The control system used in the Harvard head

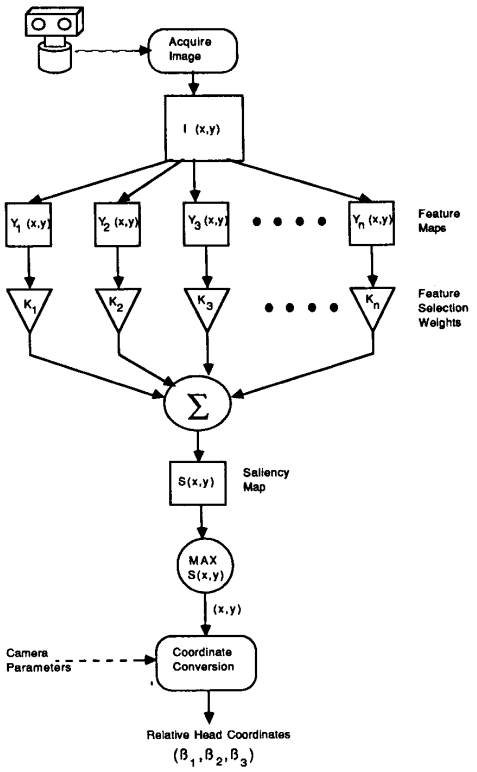

Figure 3: The feedback selection model of attention.

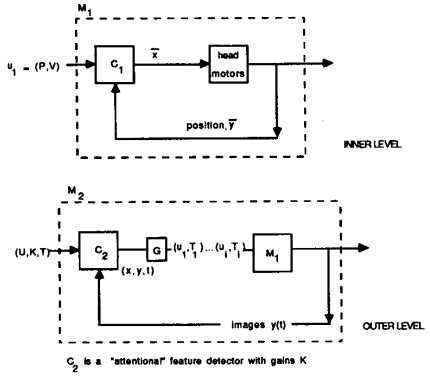

Figure 4: The two levels of the attentive control system.