An Active Vision System integrating Fast and Slow Processes

M. Castrillón-Santana[1], C. Guerra-Artal, J. Hernández-Sosa, A. Domínguez-Brito, J. Isern-González,
J. Cabrera-Gámez, F.M. Hernández-Tejera

Departamento de Informática y Sistemas
Universidad de Las Palmas de Gran Canaria
SPAIN

ABSTRACT

This paper describes an Active Vision System whose design assumes a distinction between fast or reactive and slow or background processes. Fast processes need to operate in cycles with critical timeouts that may affect system stability. While slow processes, though necessary, do not compromise system stability if its execution is delayed. Based on this simple taxonomy, a control architecture has been proposed and a prototype implemented that is able to track people in real-time with a robotic head while trying to identify the target. In this system, the tracking module is considered as the reactive part of the system while person identification is considered a background task. This demonstrator has been developed using a new generation DSP (TMS320C80) as a specialized coprocessor to deal with fast processes, and a commercial robotic head with a dedicated DSP-based motor controller. These subsystems are hosted by a standard Pentium-Pro PC running Windows NT where slow processes are executed. The flexibility achieved in the design phase and the preliminary results obtained so far seem to validate the approach followed to integrate time-critical and slow tasks on a heterogeneous hardware platform.

Keywords: active vision, movement detection, tracking, movement estimation, applications of image processing to robotics.

1. INTRODUCTION

Active Vision Systems can be considered as dynamical systems that control camera parameters, motion and processing to simplify, accelerate and perform robust artificial visual perception. These possibilities of adaptation are offered by mechanical devices such as robotic heads and motorized lens. Thus, Active Vision makes use of visual information to integrate an artificial system with the environment[1], receiving sensorial feedback.

Research and Development in Active Vision Systems[2, 3, 4] is a main area of interest in Computer Vision. Mainly by its potential application in different scenarios where real-time performance is needed such as robot navigation, surveillance, visual inspection, among many others. Several systems have been developed during last years using robotic-heads for this purpose. These systems and their adaptation capabilities offer a way to face those problems that were not solved adequately with previous schemas in Artificial Vision.

Real-time Computer Vision applications have a bottleneck in the computation of massive amount of input data with image processing procedures in a reduced and fixed amount of time. This restriction imposes severe constraints on the structure of such a system. There are some known solutions to apply to this problem. First, custom hardware developments have been a common resort to increase the processing capabilities of this kind of systems. These solutions have ranged from the design of specialized circuits and boards to the utilization of transputers or Digital Signal Processors (DSPs) networks, commonly, using VME bus to interconnect the system. However, these systems tend to be very expensive, hard to adapt to new tasks, specially those using custom hardware, and not based on mainstream cost-effective technology[5, 6]. Second, data filtering can be used to reduce the amount of data to process as biological systems which have developed foveated visual systems. Finally, it is normally necessary to adapt the intended processing architecture to the available hardware in order to fully

---

[1] Correspondence to Edif. Dptal. Informática y Matemáticas, Campus de Tafira, 35017- Las Palmas de Gran Canaria,Gran Canaria, SPAIN. Email: mcastrillon@dis.ulpgc.es; www: http://mozart.dis.ulpgc.es

exploit the capabilities of the design. This has normally resulted in complex systems that are hard to adapt to new contexts and expensive to maintain or improve.

This work describes an experimental active vision system capable of performing reactive or fast as well as non-reactive or slow processes. Reactive processes take care of system aspects that are critical for its stability. On the other hand, slow processes are not critical, as their results, while necessary, do not thread the stability of the system. These are high level tasks or capabilities that are based on low level processes. The system manages the interaction among processes with different latencies using a multithreaded software architecture that eases the modification of the process architecture of the system while allowing for specialized hardware subsystems within the system.

The organization of this paper is as follows. Next section describes the main ideas that have oriented the design of the system. Third section describes a prototype that performs real-time tracking of a person while is able to apply a face recognition method to the target asynchronously. The last sections are devoted respectively to present some experimental results and the conclusions.

2. CONCEPTUAL DESIGN

When a biological organism, capable of visual perception, is studied the first and perhaps the most amazing observation is the huge variety of activities in which the visual system may be engaged and the flexibility the system shows to deal with these activities. The visual system is a basic resort to guarantee the integrity of the organism so that it must be reactive upon the reception of visual signal that could evidence some sort of dangerous situation. Meanwhile the living being may be involved in a variety of tasks demanding much more attention, care and possibly processing time. Obviously, both types of “visual processes” are necessary for the survival of the organism.

If this simple observation is considered in the design of an active vision system, the basic problem is how to integrate fast or reactive processes devoted to low level computations with slow or background processes, whose operation is normally based on the results achieved by the low level processes. Attending to simple tasks, it is noticed that the system must be able to react to fast changes or sudden events in the environment so that the operation of certain processes must be guaranteed in a critical amount of time. On the other hand, it is obvious that certain visual tasks do not demand to be processed in a reactive fashion, even though their operation may be based on results issued by low-level tasks. For example when approaching to a person that is greeting us, we will probably begin to track his or her head, meanwhile trying to recognize that person. As the person may be looking sideways or simply may not be close enough, it is critical for recognition to pursue his movements “attentively” until a clear view of the face has been obtained. In this case, it is clear that the recognition process does not need to be as fast as the tracking task and it may operate on an opportunistic manner over front and close enough views of the person’s face.

The system presented in this paper is based on a software architecture that is based on two kinds of processes:

- Time-critical or reactive processes, normally involving some type of low level processing.
- Processes that need a greater latency for its execution and are not critical for the system’s stability so that they may run with lower priorities.

To be useful, a visual system must not only react to its visual environment but it must also provide basic visual capabilities to orient the visual activity of the system, i. e., it must be purposive. The above distinction is not new and in the last years, some proposals have been presented for the design of robotic architectures in an effort to combine two previous proposals[10, 11]: behavioral architectures[12] and centralized architectures[13]. The resulting combined schema for controlling robotic systems can be resumed in a hierarchy of three levels[10, 11, 14, 15, 16]: symbolic/high level/deliberative, reactive and servo level.

According to these proposals Artificial Visual Systems may be built based on certain simple processes that are necessary for higher level tasks. These simple processes are needed in real-time so that the system could adapt itself to the environment. In the following subsections, some considerations are expressed about these different tasks, and a sample process is selected for testing the design in an experimental system based on the hardware exposed in the next section.

Modularity and flexibility[11] have been major design goals of the system. The system should allow incorporating new capabilities, tasks or modules without affecting or compromising previous visual capabilities. To achieve this, each module must have its own incoming and outgoing signal/data and communication channels perfectly defined. Thus, system extensions would affect only to those modules that interact with the new one but would not modify the communication interface. For this purpose, the system control is designed based on a Discrete Events Model[11].

2.1 Fast tasks

A real-time vision system should process at least 25 frames per second. As it was introduced, reactive tasks are under this restriction. An example of this situation could be the movement of an element in a static scenario. This kind of systems[9, 14] offer, as one of their basic capabilities, to be able to perform tracking of the moving element. Tracking procedure needs as input data not only the current image acquired but also an object to follow. Thus, the whole procedure can be described in two steps:

- Movement Detection
- Object Tracking

A tracking definition can be: "*Tracking is a basic process in visual systems whose goal is to keep an object of interest located and fixated, pursuing it when the object is moving in the field of view*"[18]. In addition to this definition, it is also desirable that tracking keeps the object of interest centered in the image. This process has been labeled as basic, it means that this process does not manage situations such as occlusions or very fast movements. Higher level processes must take care of those cases.

A relevant aspect in any real-time tracking visual system is the amount of data the system is able to process in a bounded time. This system has as input data the object to follow and the current image. It can face the problem processing the whole image or filtering input data and processing only a windowed area of the image. This area is commonly known as fovea[8, 19]. The system must process a certain number of frames per second due to the temporal restriction. Depending on the available hardware the valid range for window size can vary its value.

After this brief description of the process, the requirements for the process can be established. There are two main desirable behaviors in a basic tracking module of a visual system:

- First, moving the camera pursuing the object and maintaining it centered in the fovea. The system has to be able to move physically the perception hardware.
- Second, detecting when the object has been lost. It is important to notice as soon as possible when the object has been lost to avoid following erroneous objects.

Up to this point a basic reactive task such as tracking has been described. This task would be adopted for the system prototype, but before it is necessary a brief description of tracking proposals already presented in the literature. Three main techniques have been developed to perform tracking[8]:

- Filter-and-follow methods send the image through a previous filter, which gives as result only highlighted areas in the image where the object of interest is. To do this, it is necessary the object to be distinctive enough to obtain good results. Unfortunately, this is not common in the majority of the objects in the real world[20, 21, 21].
- Local area correlation methods perform a correlation over the image; this method offers a maximum value where the image is more similar to the searched pattern[23, 24].
- Feature correlation methods look for object's features and then try to identify those features in following images[25].

Among these techniques, the second solution has been selected. Before attempting the implementation of this reactive task in the experimental system, a slow process will be described.

2.2 Slow tasks

The ability of pursuing focus of attention points is a necessary capability in biological and artificial visual systems to cope with the complexity of many visual tasks. Tracking information may be used not only for self-localization and navigation, but also to provide other higher level tasks with filtered data. Within the context of this work, stabilized images of the target being pursued are the input data for a set of processes dedicated to the recognition of individuals in an indoor environment. Due to the characteristics of the environment, these processes are treated as non time-critical and operate in an opportunistic manner over the data obtained by the tracking. Here the term opportunistic is to be understood in the sense that the provided images are filtered to discard those that do not correspond to a frontal or close enough view of a face.

People localization and identification/recognition is a very active line of research within the Computer Vision and Pattern Recognition communities and it cannot be considered as a solved problem. To the scope of the prototype system described in this paper several problems need to be addressed. In a first stage, it is necessary to gather evidence that the target being tracked correspond may to a moving person. Then the head should be localized and its face recognized. In this prototype the problem has been somehow simplified as it is assumed that moving targets always correspond with people. Since the individual may appear with a pose unsuitable to attempt to recognize his or her face, it is necessary to discard these views to minimize the possibility of false identifications. Thus, before proceeding the system should determine if the person being tracked shows a back, lateral or frontal view. It should be noticed however, that this prefiltering stage is highly valuable for the overall performance of the system to be able to confirm that the focus of attention being tracked corresponds to a head, even though it is not a frontal view of the face. This type of high level feedback is used by the tracking module to insist in pursuing the active focus of attention or to leave it. Thus people identification is accomplished by means of two sequential steps: Head Location and Face Recognition.

Head location is an important step in the process as it allows avoiding trying to recognize something that indeed is not a face. In [26] a person is located using a color segmentation algorithm, the systems expects a person wearing a solid color shirt, so the person localization is reduced to a search for this color in the image. Then the robot approaches the person until it arrives to a known distance of the subject. Using the supposition of a head located above the shirt, that area in the image is thresholded and a template determines if the view is frontal or not. This method operates always with the same template size, what is acceptable as the robot can control its approach and measure its distance to the person, so that the head maintains a size range.

Once the head is located, and a frontal or almost frontal view of the face is captured, a Face Recognition procedure is run. This procedure requires an off-line training stage that utilizes a set of labeled face images as a training set. After training the method permits the recognition of new faces taken among the same people. In [27] a survey of these techniques resumes different proposals for performing the classification of a viewed face:

- A K-nearest neighbor classifier can be defined in the image space.
- Reducing the problem dimension based on the extraction of some features, i.e using principal components as in the Eigenfaces[28] technique or linearly projecting onto a new representation basis as is done with the Fisherfaces[29] method.
- Using neural networks [26].

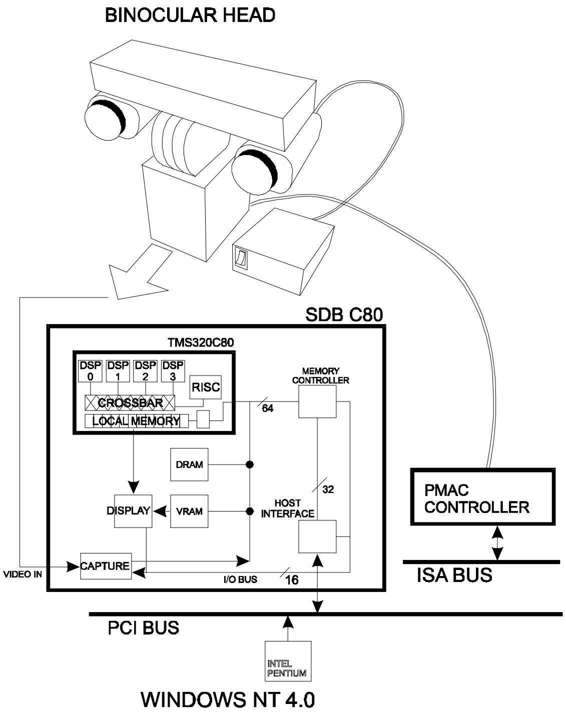
For the purpose of this paper, the Fisherfaces method was selected as it offered some immunity against illumination conditions if compared to the Eigenfaces approach.

It should be noticed that the work has been based on the structure of the system more than in the excellence of each individual method. The modularity of the system allows exchanging a module completely to add a new improved module or even more, a new cooperative module.

3. IMPLEMENTATION

3.1 Hardware

The main feature of this system is the design of an architecture based on the interaction of several hardware elements to compose a full-fledged perception-action system. The active vision system presented has been developed using heterogeneous components as below figure depicts. TMS320C80 DSP for heavy image processing and a stereoscopic robotic-head as active element are the most remarkable hardware components of the system. A PC running Windows NT 4.0 interconnects all these elements.



The perception subsystem is based on a TMS320C80 development board[30] to perform image acquisition and tracking of the target. There is a wide range of hardware solutions to perform tracking from expensive custom VLSI design, recent FPGAs, through those traditionally based on DSPs or transputers, up to economic general-purpose processors. DSPs and transputers has been chosen for most experimental active head-eye systems due to the ratio processing-power/cost offered. The launch of TMS320C80 with its architecture and capabilities for image processing offers a new point of view for this kind of systems. C80 is actually composed by five processors, four parallel processors (PPs), which are actually designed to perform image processing and a master processor (MP) dedicated to management and control tasks.

The action subsystem is a commercial motorized robotic head. It offers four mechanical degrees of freedom: pan, tilt and two vergences, plus other six optical degrees: iris, zoom and focus for both lenses. All of these degrees are controlled via a commercial controller board. This board provides capabilities for reading robot position and commanding new poses or lens configuration. Current position of the robotic head is a critic feature of the system in order to relate robot pose and image processing results. Synchronization is necessary between pose data and images to avoid unexpected behavior of the system. As a reminder, the system should be able to adapt any of the head degrees of freedom to correspond with processed results. This task that can be labeled as action task is basically the translation of results coming from the perception subsystem into movements of the head. Both subsystems, perception and action, need a layer to communicate each other. A PC with a Pentium processor that hosts subsystems carries out the control of the operation cycle. Different problems have to be solved by this processor to coordinate the perception and the action subsystems.

Some hardware features force the distribution of the processes; for example, heavy image processing is assigned to the most powerful hardware available, which is DSP C80. A PC running Windows NT is in charge of sending commands to the robotic head and background tasks, which are not critical in this prototype.

3.2 Modules

The prototype is made up of four main software modules. Each of these modules performs a concrete task and makes use of the communication layer provided by the PC to interconnect each other. In the following, a brief description presents main aspects of their functionality and implementation.

1. Detector:

   The objective of this module is the detection of moving areas and the definition of attention zones. In the first prototype it was supposed only one object is attended, so the system can be detecting or tracking but not both operations simultaneously. Future developments plan to manage a multi focus of attention behavior.

   Movement detection is achieved from optical flow computation using an approach[31] that estimates the optical flow in the center of variable rectangular patches. The resulting optical flow approximation provides a good estimation for detecting “moving areas”. The system selects the largest or fastest moving area to fixate on it. A small window focused on the center of gravity of the selected patch is used to obtain a primary pattern for tracking. The variable size of the patches allows for selecting the resolution of the optical flow returned.

   Once the module has detected a movement, the robotic head centers that area executing a saccadic movement. Then, this module issues a position, an image and velocity estimation to the tracker module.

2. Tracker:

   Once a moving object has been detected, based on current image and the fixed pattern provided by the detector, this module is responsible of:

   - Searching the object on the image.
   - Estimating the target’s trajectory.
   - Commanding in consequence the mechanical elements of the system.

   Considering the implementation of the tracking algorithm more in detail, two major tasks are distinguished: correlation and target's trajectory estimation. The correlation algorithm searches a previously chosen pattern only on a rectangular foveal area[8, 19] of the image. Correlation is a fast and simple technique whose performance can be improved using pattern updating policies in order to deal with variations in the appearance of the target. Basically, the tracker keeps a collection of the most stable views of the target that is updated using an error measure between the different patterns in the collection and the best match area with the active pattern. This technique is described in detail in [36].

   In order to optimize the utilization of the image processing resources present in the C80 architecture it has been designed a parallel correlation algorithm[32] that divides in a special way the rows of the fovea by the number of PPs so that each PP is equally loaded. Using this approach all PPs work with exactly the same amount of data located in their cache memory and without idle times. The partial results issued by each PP are integrated by the MP RISC processor. Correlation results, best match position and correlation index, provide enough information to decide what the new position of the object is or alternatively if the object has been lost. This information is transferred to the PC to command the robotic head.

   A foveal image can be used to adequate the processing load to the computational power of the system. Commonly the fovea is located in the center of the image but in our case a relocatable fovea has been considered. This configuration allows the fovea to be placed anywhere in the image [33]. The relocatable fovea indeed reduces the response time of the pursuing mechanism to the visual processing latencies, instead of the larger mechanical latencies.

   Trajectory estimation is necessary for pursuing an object, this technique allows the system moving in advance, i.e., anticipating object movement. For this purpose, an alpha-beta filter which is an adaptation of a second order Kalman filter[34] is used. This filter makes use of the previous position, velocity and acceleration data to predict the trajectory of the object in short term. Once the new position of the target is computed, the head tries to focus the target on the center of the field of view.

3. Identifier:

   Asynchronously the tracking module makes available to the identifier module a central patch in the last processed image. The tracker module will keep its focus of attention on the upper part of the person, which is assumed his/her head. Thanks to the tracking module, there is no need of segmenting the moving person on the sequence[27].

   The identifier module acts in three steps in a cascade recognition fashion:

   i. First, the head zone is located. Knowledge of the environment has been used to simplify this task assuming that the head area is always darker than the background. Thus a simple threshold can be applied to the image to get those points that are supposed to correspond to the head of the subject. Once the head is located the window containing the face zone is resized[35] to the dimensions used during the learning stage.

   ii. After the head has been located, the Fisherfaces recognition method is applied to discriminate among frontal, back and lateral views of the head. Only frontal views are considered for face recognition. Classification has been based on a learning set composed of positive samples of frontal views and negative ones constituted by lateral and back views of the head, and head counterexamples obtained from the visual environment of the system.

   iii. Face recognition is performed using the Fisherfaces recognition method over the extracted face. In [29] a comparison between two reduced space methods: eigenfaces and fisherfaces is presented. Fisherfaces offers the best performance when high illumination variance is typical in the system, as in this case where person location is not constant, i. e., the person can be in movement so light conditions may vary.

   Whenever the first test returns that current view is a frontal view, a second Fisherfaces classification procedure is applied to determine the identity of the individual. In order to achieve a robust classification a double scheme has been implemented. The first is based on a voting scheme using K nearest neighbor classification (K=3). Secondly, temporal congruency is required in order to avoid positive identification with just one sample. Thus the system must confirm a positive identification for a given time interval.

4. Supervisor:

   This module is responsible of the coordination of the modules. By definition, the system works asynchronously; and due to that, this module has to be in charge of synchronizing all the events and signals generated inside the system to resolve conflicts and define objectives. It should be noticed that not all the modules have the same requirements, so the supervisor module imposes a communication structure for an easy addition of new modules to the system. By one hand, focusing on reactive tasks, the moment of acquisition of an image and the pose of the robotic head have to be put in correspondence. On the other hand, it must control which image fragment is being processed by a background task in order to know who was, and when.

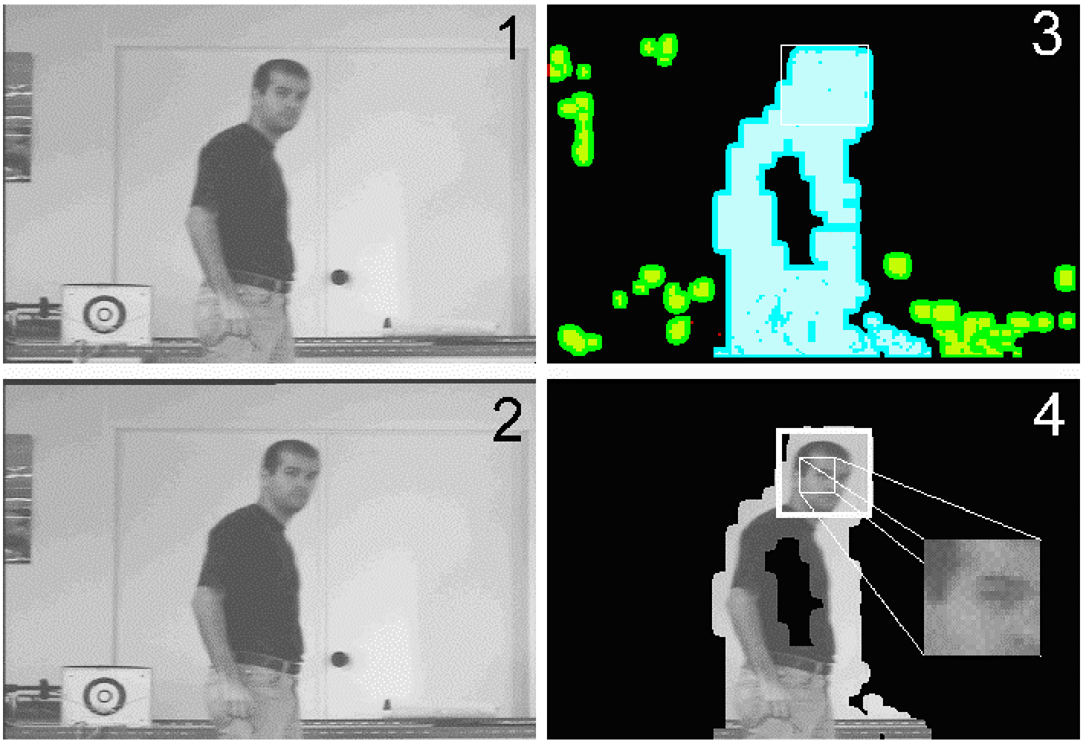
   An architecture that lets to implement an asynchronously complex system composed of different tasks has been developed to cover these supervisor requirements. The proposed software architecture allows mapping the conceptual design of task and signals into a working implementation. The architecture maps tasks to threads and messages or events between tasks to a particular redefinition of *signals*. Each signal transports data and each thread can receive and send signals. The architecture assigns a multiple buffer of input signals to each thread and permits to link each input signal and its data with a specific behavior or process.

4. EXPERIMENTS

The described experimental system has been tested with real world images. In a first stage, the tracking system was tested following people walking at different speeds, once the person was detected after entering in the field of view. Current implementation of optical flow calculus takes 90 milliseconds over 176x120 images on a Pentium Pro 200Mhz. This prototype is able to perform a correlation on an image each 31 milliseconds. This result is obtained splitting the C80 ALU and searching a 24x24 pattern on a 80x80 fovea using gray levels. This response time provides C80 with an extra time up to 40 milliseconds (25 frames per second) to perform other image processing operations.

Once the real-time tracking process showed an expected performance, it was added to the prototype the recognition module. Its integration did not affect the tracking performance as those tasks reside in different hardware. Identification results are not relevant because of the restricted learning set used: a set of 20 views for 4 different people.

The aim of these previous experiments was to prove that the underlying design supported a visual system that offers an asynchronous structure for executing different priorities processes.



Images 1 and 2 show two consecutive frames. On 3 it is shown the detected moving areas obtained from optical flow computed using images 1 and 2. Finally number 4 presents the first fovea selected (big thick square) and the initial pattern (small thin square) selected for correlation.

In this figure some frames of a tracking example are presented. Top left image corresponds to first moment. The system is able to pursue the person even when he is rotating over his vertical axis. The fovea is drawn as a black frame. After a target movement the fovea moves on the image until the mechanical system can update and center

5. CONCLUSIONS

It has been designed, developed and tested a prototype of a vision system for real-time detection, tracking and identification of people. The main aim that has guided the design of this active vision system has been to achieve a first prototype where fast or reactive processes could interplay with slow processes in order to define the visual capabilities of the system. Coordination between both types of processes has been formulated in such a way that does not compromise the reactivity of the system to its environment.

The level of performance achieved has been possible through a suitable combination of capable hardware and a modular software architecture. The singular pieces in the utilized hardware involve a commercial robotic head and an advanced DSP targeted for image processing applications, installed on a standard PC under Windows NT. The software architecture of the system has exploited a modular approach that has proved valuable to attain a flexible and easy-to-modify system. All modules in the system share a common object structure and communicate each other using a scheme based on discrete events.

6. ACKNOWLEDGEMENTS

This work has been partially supported by Spanish CICYT Project TAP95-0288.

7. REFERENCES

1. K. Pahlavan and J.-O. Eklundh, “A head-eye system for active, purposive computer vision”, Tech. Rep. ISRN, KTH/NA/P--90/31--SE, 1990.
2. Aloimonos, J. Weiss, I. and Bandyopadhyay, A. “Active Vision”, *International Journal of Computer Vision*,1(4):333-356, 1987.
3. Bajcsy R, “Active Perception”, *Proceedings of the IEEE*, 76(8):996-1005, 1988.
4. Bajcsy R, “Active Perception vs. Passive Perception”, *Proc. 3rd. IEEE Workshop on Computer Vision, 55-59*, 1985.
5. Kourosh Pahlavan, “Active Robot Vision and Primary Ocular Processes”, *CVAP*, 1993.
6. Ulf M. Cahn von Seelen, Brian C. Madden, “A Modular Architecture For An Active Vision System Using Off-The-Shelf Components” GRASP Laboratory, Department of Computer and Information Science. University of Pennsylvania. 1996.
7. Koch C. And Ullman. S., “Shifts in Selective Visual Attention: Towards the Underlying Neuronal Circuitry”, Human Neurobiology 4: 219-227, 1996.
8. M. Wessler, L.A.Stein, “Robust Active Vision from Simple Symbiotic Subsystems”, Technical Report, MIT, 1997.
9. K. Pahlavan and J.-O. Eklundh, “A head-eye system --- Analysis and design”, *Computer Vision, Graphics, and Image Processing:Image Understanding*, vol. 56, no. 1, pp. 41--56. Also ISRN KTH/NA/P--92/08—SE, 1992.
10. F. Tièche, C. Facchinetti & H. Hügli, “Behaviour-Based Architecture to Control an Autonomous Mobile Robot”, in *Proc. Intelligent Autonomous Systems* IAS-4, Karlsruhe, March, 1995.

11. Kosecka, J.: “A Framework for Modeling and Verifying Visually Guided Agents: Design, Analysis and Experiments”, PhD thesis, GRASP Lab, University of Pennsylvania, 1996.
12. R.A.Brooks. “Intelligence without reason”, TR 1293, MIT, 1991.
13. J.S. Albus, “Outline for a Theory of Intelligence”, in *IEEE Trans. Sys. and Man. Cyb*., 473-509, 1991.
14. R.P. Bonasso, E. Huber and D. Kortenkamp, “Recognizing and Interpreting Gestures within the Context of an Intelligent Robot Control Architecture”, Technical Report, Metrica Inc. Robotics and Automation Group, NASA Johnson Space Center, 1996.
15. R.P. Bonasso and D. Kortenkamp, “Using a Layered Control Architecture to Alleviate Planning with Incomplete Information”, *AAAI Spring Simposium on Plannng with Incomplete Information for Robot Problems*, 1996.
16. H.I. Christensen, P.Pirjanian, “Theoretical Methods for Planning and Control in Mobile Robotics.” *Knowledge Engr. Systems* 97, 1997.
17. Kjell Brunnström, Jan-Olof Eklundh and Tomas Uhlin, “Active Fixation for Scene Exploration”, *International Journal of Computer Vision*, vol. 17, pp. 137--162, 1996.
18. Crowley J. L., Christensen H. I. (eds.), *Vision as Process*. Springer-Verlag, Berlin, 1995.
19. F.Panerai, C. Capurro, G. Sandini, “Space variant vision for an active camera mount”, TR 1/95, LIRA-lab-DIST University of Genova, Italy, 1995.
20. Anne Wright. “A high speed low latency portable vision sensing system.” In D.P. Casasent, ed., *Intelligent Robots and Computer Vision XII: Algorithms and Techniques*, vol. 2055, 263-270, SPIE August 1993.
21. H. Kimura and J. J. E. Slotine. “Adaptive visual tracking and gaussian network algorithms for robotic catching”. *In Advances in Robust and Nonlinear Control Systems*, 67-74, 1992.
22. S.A. Brock-Gunn, G.R. Dowling and T.J.Ellis, “Traking using colour information”.TR TCU/CS/1994/7, City University London, 1994.
23. I.D.Reid and D.W. Murray, “Tracking foveated corner clusters using affine structure”. *In Proceedings of the Fourth International Conference on Computer Vision*, pages 76-83, Berlin, IEEE Computer Society, 1993.
24. E. C. Hildreth, “Computations underlying the measurement of visual motion”. *Artificial Intelligence*, 309-354, 1984.
25. H.Inoue, T. Tachikawa and M. Inaba, “Robot Vision system with a correlation chip for real time tracking, optical flow and depth map generation”. *In Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, 1621-1626, France, IEEE Computer Society, 1992.
26. C. Wong, D. Kortenkamp and M. Speich, “A Mobile Robot that Recognizes People”, *IEEE International Conference on Tools and Artificial intelligence*, 1995.
27. R. Chellappa, S. Sirohey, C.L. Wilson, C.S. Barnes, “Human and Machine Recognition of Faces: A Survey”, CAR-TR-731, Center for Automation Research, Maryland, 1994.
28. Turk, M. And Pentland, A.: “Eigenfaces for recognition”, *J. Cognitive Neuroscience* 3(1), 71-86, 1991.
29. Belhumeur, P., Hespanha, J. And Kriegman, D.: “Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection”, *IEEE Trans. On PAMI* 19(7), 711-720, 1997.
30. *TMS320C8x Software Development Board Technical Reference*, Texas Instruments, 1997
31. Campani, M. And Verri, A. “Motion analysis from first order properties of optical flow”, *CVGIP: Image Understanding* 56 (1), 90-107, 1992.
32. C. Guerra-Artal, M. Castrillón-Santana, J.D. Hernández-Sosa, J. Isern-González, J. Cabrera-Gámez, F.M. Hernández-Tejera. “A C80 DSP-based Active Vision System for Real-Time Tracking”. *Proceedings ICSPAT*, 1998.
33. Posner, M. I., “Orientation of attention”, *Quart. J. Experimental Psichology*, 32:3-25, 1980.
34. Yaakov Bar-Shalom, Xiao-Rong Li, *Estimation and Tracking: Principles, techniques and software*, Artech House, Boston, 1993.
35. Wolberg, G. *Digital Image Warping*, IEEE Computer Society press, 1990.
36. M. Hernández, J. Cabrera, M. Castrillón, A .C Domínguez, C. Guerra, D. Hernández and J. Isern, ”DESEO: An Active Vision System for Detection, Tracking and Recognition“, *Submitted to the International Conference on Vision Systems*, Las Palmas de Gran Canaria, January 1999.