DESEO: An Active Vision System for Detection, Tracking and Recognition

M. Hernández, J. Cabrera, M. Castrillón, A. Domínguez, C. Guerra, D. Hernández, and J. Isern *

Grupo de Inteligencia Artificial y Sistemas
Departamento de Informatica y Sistemas
Edificio de Informatica y Matematicas
Campus Universitario de Tafira
Universidad de Las Palmas de Gran Canaria
35017 Las Palmas - SPAIN
Phone: +34 928 458758/00
Fax: +34 928 458711
emailmhernandez@dis.ulpgc.es

Abstract. In this paper, a basic conceptual architecture aimed at the design of Computer Vision System is qualitatively described. The proposed architecture addresses the design of vision systems in a modular fashion using modules with three distinct units or components: a processing network or diagnostics unit, a control unit and a communications unit. The control of the system at the modules level is designed based on a Discrete Events Model. This basic methodology has been used to design a real-time active vision system for detection, tracking and recognition of people. It is made up of three functional modules aimed at the detection, tracking, recognition of moving individuals plus a supervision module. The detection module is devoted to the detection of moving targets, using optic flow computation and relevant areas extraction. The tracking module uses an adaptive correlation technique to fixate on moving objects. The objective of this module is to pursuit the object, centering it into a relocatable focus of attention window (FOAW) to obtain a good view of the object in order to recognize it. Several focus of attention can be tracked simultaneously. The recognition module is designed in an opportunistic style in order to identify the object whenever it is possible. A demonstration system has been developed to detect, track and identify walking people.

1 Introduction

The design of Computer Vision Systems (CVS) has experienced diverse evolutions and reorientations during its almost thirty five years of existence. These have gone from the most naive approaches, that tried the design of general CVS in a technologically poor framework, to more actual trends aimed at the design

* This research is sponsored in part by Spanish CICYT under project TAP95-0288.

of CVS capable of solving specific tasks in a robust manner showing continuous operation and real-time performance.

During the last decade, the design and construction of CVS has been largely influenced by the Active Vision paradigm, initiated by the seminal works [1, 3]. It is actually understood as a methodological framework [18] in which the design of artificial vision systems is based on elaborated mechanisms for the control of sensors parameters and processing with the aim of achieving a more robust operation, sometimes exploiting the restrictions of the environment in a Gibsonian or ecological sense.

Many systems have been built within the active vision paradigm, contributing new ideas and powerful techniques to solve specific problems. Andersen [2] includes an extensive survey of vision systems developed using robotic heads up to 1996. However few of them have considered the problem of extending the designed systems to new contexts other than those initially considered at design time or connecting the system to higher level modules [8].

Crowley [7] describes the methodological foundations for the integration of several reactive continuous processes in a Discrete Event Model with a supervisory control scheme. Using this approach, a system for detection, fixation and tracking is implemented. This system employs a combination of visual processes that rely on blink detection, color histogram matching and normalized cross correlation for face detection and tracking [9]. XVision [12] is a modular portable framework for visual tracking designed as a programming environment for real-time vision. It consist in a set of image-level tracking primitives and a framework for combining tracking primitives to form complex tracking systems. Perseus [13]is an architecture developed for the purposive visual system of a mobile robot and lets it to interact visually with the environment. It is based on object representation obtained from certain feature maps and in the use of visual rutines [21] paradigm. In order to perform person detection in the scene, Perseus uses an oportunistic strategy. Kosecka [14] proposes an approach for systematic analysis and modelling visually guided behaviours of agents, modelling behaviours as finite state machines and resolving the conflicts of parallel execution of them via supervisory control theory of Discrete Event Systems [19].

In this paper we describe DESEO (acronym of Detection and Tracking of Objects in Spanish), a development based on a modular conceptual architecture for the design of perception-action systems, using active vision for the detection and tracking of mobile objects. Its modular nature eases the mapping of complex activities over a network of modules that implement more simple behaviors. DESEO integrates a commercial binocular head, a DSP-based motor controller board and a TI TMS320C80 (C80) parallel DSPs on a Dual Pentium II PC running Windows NT 4.0. The system is able of real-time continuous operation and made up of several modules that perform different operations but share a common internal structure. Due to the modular nature, the system can be tailored to perform different tasks. As an experimental development, a system aimed at the detection, tracking and recognition of individuals is described.

The organization of this paper is as follows. Section 2 is devoted to introduce the main ideas underlying the design of DESEO. Section 3 describes its architecture and provides a description of the modules developed for the demonstration system. The internal organization of a generic module is presented in Section 4. The last sections are dedicated to the description of some experiments and conclusions.

2 Considerations for the Design of CVS

Actually Computer Vision seems to lack a common approach for the design of Vision Systems, situation which is extensible in general to the design of more complex robotic systems [7]. This lack of methodology normally provokes that CVS are designed as closed systems that are hardly reusable for solving other vision problems than those originally considered. When designing a CVS is necessary to consider the system within the framework of a conceptual methodology that assumes the a priori considerations, goals and design criteria. In this scenario the availability of a conceptual architecture has the mission of closing the gap between the resources provided by the affordable technology and the design restrictions and requirements. Pursuing these ideas we are trying to develop vision systems in a systematic way using for design the following three simple considerations:

1. Computer Vision systems should be programmed to be built in an incremental and modular way. So the concept of module, as the minimum logical entity that performs something useful, is a basic element. Also to cope with this objective and favor the reconfigurability of the system in terms of its modules, these should share a common structure and interface.
2. Internal details of each module, regarding specifically what that module computes, are going to vary between modules. However, all of them are going to share some common aspects. Most modules will exhibit three distinguishable functional parts respectively dedicated to perform the computations to obtain results from the data, to evaluate the results and to control the performance of the algorithms.
3. For sake of versatility, a control scheme which combines data driven and objective driven mechanisms supplies a powerful tool in order to obtain diverse and complex behaviors.

These three ideas are the basement of the proposed Bottom-Up/Top-Down Module (BU-TD). It is basically a percepto-effector unit with the following main functions:

- The obtainment of a description or computational result from some input data. Its objective is to derive results or descriptions from the analysis of the input data in a bottom-up or data driven fashion. This is the task of the bottom-up (BU) unit depicted in Figure 1. In general, a BU unit is conceived as an interpreter that operates cyclically.

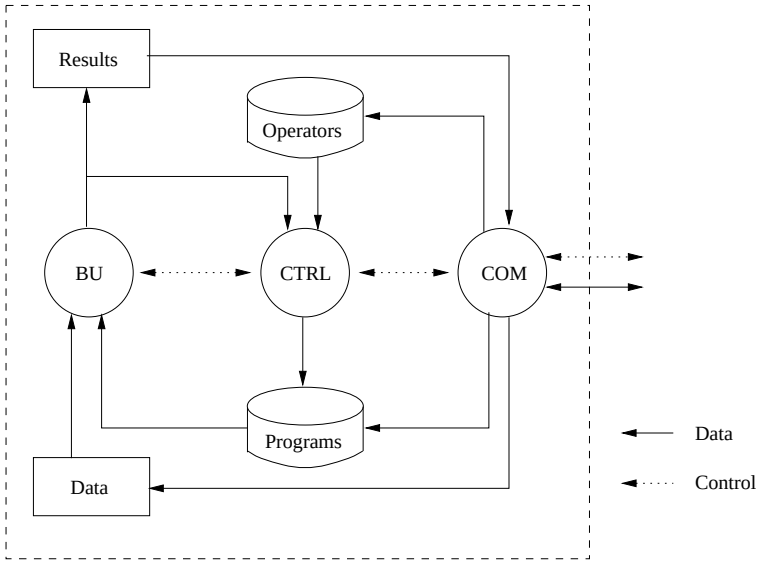

Fig. 1. Conceptual Architecture of a BU-TD Module.

- There is normally a need to evaluate the results obtained by the operation of the BU unit and control its operation through available parameters. This is performed by the top-down control unit (CTRL) to close a first control loop. This unit has also the task of decoding incoming events and messages into the control actions (Operators in Figure 1) meaningful to the module.
- Distribution of computations is normally a necessity, in terms of required computational power as well as in terms of functional modularity, so we normally do not want to design systems where the different parts executes synchronously in a single thread. The tasks in the system should be assigned to dedicated modules that exchange events and messages but that may run at different frequencies. The communication unit (COM) allows for sending or receiving data to other modules without interrupting the operation of the module. This capacity of modules to operate asynchronously is a essential characteristic in the design of reactive perception-action systems.

The state of each module is controlled at different levels. At the lowest level, the control unit of the module is monitoring its own operation through the evaluation of the results achieved. This internal control loop lets the module adapt its operation parameters and react to changes in the environment in a reactive way. At a second level, control is modeled using a Discrete Event System (DES) paradigm [19, 15]. Thus modules can receive and send events to other modules to share results or signal certain circumstances. Aside from these two bottom-up control mechanisms, there is a top-down control which is arbitrated by a supervisor module. The main goal of this module is to decode complex commands received from an upper layer into commands that are issued to the

modules of the system to set its behavior and to serve results obtained from the lowest levels towards the upper layer.

3 Demonstration System

Using the ideas introduced in previous sections, we have developed a prototype capable of performing different types of task. By default, the system can scan a zone trying to localize moving objects which are supposed to be upright position walking individuals. After fixating on moving objects, it tries to localize the head and identify the person. A possible variation of this behavior is to command the system to find a specific individual. The system comprises the following set goal-oriented functional modules (GOFM):

Supervisor: This module serves as an interface between the system and outside and its basic task is to decode the commands received from upper layers or the user into a pattern of behaviors of the system modules.

Detector: The goal of this module is to direct the attention of the system towards areas of the scene where motion is detected.

Tracker: As its name suggests this module is responsible of controlling the active vision system which is made up of a Helpmate Robotics' BiSight binocular head, a motor controller board and a PCI board containing a C80 parallel DSP. The prototype uses only one camera so that movement is performed on that camera's vergence and head tilt angles. Image acquisition is done using the frame grabber available on the C80 board. The tracking of moving targets is done at frame- rate in the C80 DSP as is explained bellow.

Recognizer: The recognition of known individuals is the task of this module. It operates in two steps. At a first stage, it analyzes the small window returned by the Tracker module to determine if this image corresponds to a head or not and its pose (front, lateral or rear) if it is the case. At the second stage, zones identified as heads in frontal pose are analyzed in order to attempt the identification of the person.

The behavior of the whole system is the resulting pattern of conmmutation between the GOFM modules as is dictated by Supervisor module.

3.1 Detection Module

This module is really a pipeline of two modules, respectively devoted to detect areas exhibiting coherent movement, and to produce an estimate of the head position within the selected area. The movement detection acts as the basic mechanism for capturing the attention of the system towards potentially interesting areas. For the sake of simplicity, in the context of this application it is assumed that moving areas a priori correspond to walking individuals. This hypothesis is later confirmed or unconfirmed based on evidences, as is explained bellow.

The detection of movement is done first computing the optical flow and then segmenting it to select areas that exhibit a coherent movement. The optical flow is computed according to the technique presented in [5]. This technique approximates the flow field by estimating the optical flow only at the center of the rectangular patches in which the image is divided. The basic assumption of the technique is that within the rectangular patches a linear approximation of the flow field holds. Clearly, the computed optical flow is an approximation to the "real" flow field. In order to compute the optical flow at points other than the center of the patches, a linear interpolation can be used.

This technique provides a flexible solution that allows to balance numerical accuracy with computational requirements by varying the size of the patch. This makes possible to compute the optical flow at different levels of accuracy, for example at high resolution in the FOAW for fine tracking and at a coarser in the periphery of the visual field for detecting large moving objects [17].

This module can account for the egomotion of the robotic head. Using the known kinematics of the head it is possible to subtract the component of the optical flow due to the egomotion of the head [17]. This allows for detection of moving objects even when the head is moving.

The second stage of this module uses thresholding of the computed optical flow and blob detection to extract "patches" showing coherent movement. These patches are ranked according to their size and an area centered on top of the best ranked blob is extracted as a potential area of interest. This area is expected to contain the head of the individual if any. Its size and position inside the blob are determined using knowledge about normal width and height proportions of human heads and the measured mean velocity of the that zone. A similar strategy is used in many other systems as in the Perseus system [13].

3.2 Tracking Module

The tracking process is capable of tracking several targets on a sequential manner. Each target is an element in a list of focus of attention built from high saliency areas returned by the detection module. For the focus of attention that is active at each time, the tracking is performed first commanding a saccade to the predicted location of the target in the static frame of reference of the head, and then proceeding with a normal tracking until the next focus of attention is scheduled for visual control. The tracking is performed by the correlation procedure explained below. For the sake of real-time performance, this procedure is restricted to a relocatable window of $m \times m$ pixels (80×80). This FOAW can be placed anywhere in the visual field [11] to rapidly follow a moving object, somehow alleviating the latencies introduced by the electromechanical system of the head. In parallel, the head is commanded to the expected target's position using an alfa-beta predictive filter, that takes into account all the latencies of the tracking process. The head is commanded to new positions every 40 ms.

Another interesting feature of the FOAW is that it can switch to a new focus of attention at frame rate if the new and old focus are visible within the same visual field. The switching policy between focus of attention is based on

priority queue. The length of the queue, just the number of focus of attention that the system can track simultaneously, can be variable but in our experiments has been fixed to 2 or 3. Initially, the list of focus of attention is empty until the supervisor or the detection module start sending high saliency areas to the tracker to fix on them. The management of the focus of attention is carried out by a simple scheduler that assigns a priority to each focus of attention. This priority is computed on the basis of the estimated velocity of the target and the error in the predicted position. This means that even if there are other focus of attention pending for visual attention, the system may not be able to attend them if the active focus of attention moves very fast or in an unpredictable manner. In practice, tracking several targets is only feasible if they move slowly and with constant velocity. Switching between focus of attention is made by means of a saccade followed by fixation. The tracker must assign the active focus of attention a time slice enough for performing the saccade and stabilizing on the target to update the estimates of position and velocity.

The basis of the tracking module is the determination of the new target location using a real-time correlation operator that returns the best match position between a series of patterns and the current FOAW image. This correlation process must accomplish two constraints: temporal constraints, it must return a new position estimate every new frame, and it should accommodate the evolution of the object's view appearance. This is a need if the object is moving in an unrestricted manner (change of scale, rotations, deformations, ...) in an unstructured environment where the illumination is non uniform or may change during the tracking process.

Several iconic correlation algorithms can be implemented for real-time performance on a parallel DSP like the C80. However, its performance is very limited if the object being tracked change its appearance while the model is kept constant. This is due, both to the characteristics of the matching measure [10] and to the static nature of the model that does not track the variations in the visual appearance of the area being tracked. Obviously, to accomplish with the second condition a mechanism is needed to update the model. To deal with this problem, we have developed a new algorithm based in the assumption that if the update is performed frequently enough, literally at frame-rate, the change in the visual appearance of the object with respect to the observer is expected to be smooth. The algorithm uses several patterns corresponding with different views of the object of interest. This set of iconic models constitutes a stack of L elements that may be assimilated to a short-term visual memory. This model of visual memory is created, used and updated autonomously by the algorithm through a function that evaluates its relevance both in terms of difference or error with the model and its perdurability and obsolescence.

Every focus of attention object has an associated stack (STK), whose first element, $STK(0)$, represents the most recent model of the object. Every model in the stack is represented by: an array $m(i)$ of $n \times n$ pixels, that stands for the $i\text{-}th$ iconic model of the object, the obsolescence (t_s) of that model or the time when this model was recalled last time, the persistence time (t_p) or the amount

of time that this model has been actived stack (STK), whose first element, $STK(0)$, represents the most recent model of the object as the identified model, and finally, a flag that can be used to lock a model in memory so that it can not be removed by another model, either because it corresponds to a view of the object that has been previously recognized or because it corresponds to a view of an object that has been uploaded from a model database to be searched. With this data and for each model $STK(i)$, an utility index (U) can be computed as $U(t_a, i) = t_p(i)/[1 + t_a - t_s(i)]$, where t_a represents the actual time. This utility measure takes into account the stability or persistency of a model and the time interval since it was recalled form the stack. Using this measure it is possible to quantify the relative "liveness" of the different models in the stack. Thus a model with a low utility measure is a candidate for being removed from the stack when refreshing the memory. Three type of processes act over the stack of models: stack creation, tracking by comparison and model updating.

A. Stack creation: A stack is created whenever a tracking is initiated for a new focus of attention. This may result from a detection process or due to an executive order received from the supervisor module. In this moment the data space is allocated and the memory allocating process begins with the first element, $STK(0)$, corresponding to the active model of the object being tracked. The memorization of new models in the stack is done by transferring the active model, $STK(0)$, to a new position, $STK(j)$, whenever the active model is going to be updated and certain conditions hold. For the active model to be memorized, its persistence (t_p) must be long enough or, equivalently, it needs to have been active for a sufficiently long period of time (actually 5 frames, or 200 ms). In order to avoid introducing weak models in the memory, the candidate model must also contain certain level of variability or structure. This is checked by computing the mean of the absolute value of the gradient over every pixel of the model, which must be larger than a threshold. Finally, it must be somehow different to models already present in the memory, this being measured directly as the Sum of Absolute Differences (SAD) between the candidate and the rest of models. To determine a sufficiently large difference a threshold is employed.
B. Tracking by comparison and updating the model. The process is controlled by two limits, corresponding respectively to a lower (E_{min}) and upper (E_{max}) limits for the minimum correlation error. Given a minimum or best match error at current time, $E(t_a)$, between the active model $STK(0)$ and the FOAW, if $E(t_a) < E_{min}$ the active model remains unchanged and only its persistence is incremented. Else, if $E(t_a) > E_{max}$, this means that possibly the object has been lost and a search mode is triggered. Within this mode every model in the memory, starting with the model with largest persistence (t_p), is tried as the active model and the best match point is searched with the FOAW located at different positions until a good match is found. Otherwise, $E_{min} < E(t_a) < E_{max}$, the active model is updated through substitution by the $n \times n$ image window centered on the best match position. After this, a correlation is repeated for each of the models present in the memory

over an area centered in the best match position whose size is a fraction of the FOAW. Actually, this is a window of $(m/2) \times (m/2)$ pixels, being m the side length of the FOAW. From this correlation the model providing minimum error is selected, and if this error is less than the value obtained with the previous active model, it is taken as the new active model and its persistence (t_p) and obsolescence (t_s) are updated. This process is carried out to compensate the drift on the location in the image of the best match that is observed when the active model is updated.

Thus the system has its "memory" structured at three levels depending on its function and persistence. The active model used for the tracking can be considered as a very short term memory that needs to be refreshed frequently. The collection of models associated with a focus of attention object act as a semi-permanent working memory with a medium degree of persistence. Finally, the face database used by the recognition module constitutes the long term or permanent memory of the system.

3.3 Recognition

The recognition module operates over a small window returned by the tracking module, centered at the point of best match in the image. Given the complex characteristics of the recognition process and its inherent limitations, the recognition is tried as a chain of opportunistic classifications. Some preprocessing is done at the beginning, basically to locate the center of the head in the window returned by the tracker. At the present implementation this is done exploiting the context at our lab, where the background appearing in the images is normally uniform and lighter than people appearing in the image. Also the area of the image containing a face presents a higher variance than the rest of the window. Thus we use a combination of gray level and variance thresholding to select points likely belonging to a head/face to compute the centroid and size of the head. Actually, our prototype is restricted to operate at short range around a predetermined distance so that we don't take care of changes of scale. Clearly, the approach followed for locating the subject's head is rather simplistic and exploits the conditions present at a particular escenario. We plan to utilize the skin color as the basic mechanism for head/face detection as in [23],[9].

After this preprocessing, the rectangular area enclosing the selected points is extracted and warped to a predetermined size using Fant's resampling algorithm [22]. This image is supplied to a two steps classification process. At the first step in the recognition process, the selected area is classified to determine the pose of the head. If the selected pose is not frontal the recognition process stops and awaits a new image. When a view is classified as a front view of the head, a second classifier is used to determine the identity of the person. Normally, several positive identifications are accumulated before a recognition is considered definitive.

Both classifiers use the Fisherfaces method, a recognition procedure based on linear projections in the space defined by the eigenvectors of the image [4], a technique conceived as a derivation of the Eigenfaces method [20]. These methods are

based on subspace classification in the vector space obtained as a lexicographical ordering of the image pixels. The Eigenfaces method strongly degrades when, as in our case, there are changes in illumination conditions or slight changes in pose or gestures. In many situations, also in our case, the variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to change in the face identity [16]. Both techniques have been tested in the context of our application, having obtained with the Fisherfaces method a substantial increase in the quality of recognitions in different conditions of illumination.

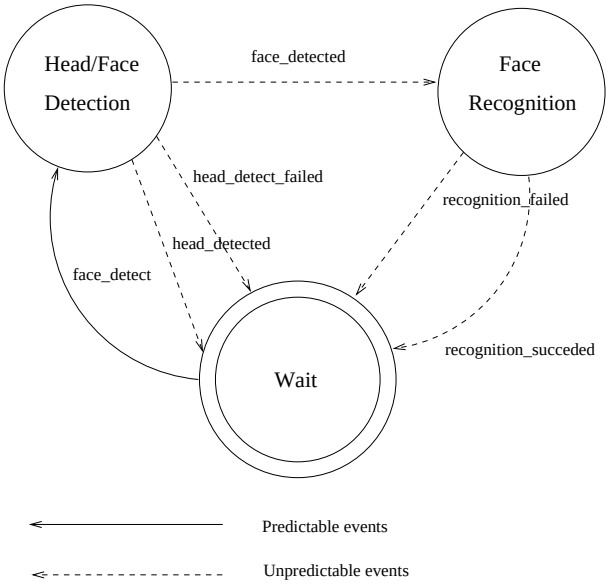

Fig. 2. Detection module control.

4 Module Design

The modules that integrate the DESEO system share a common conception that follows the methodological considerations exposed in section 2. Internally, each module is modeled as a nondeterministc automata comprising a set of states and events. Each state is assimilated to an elementary computational task that is carried out using the module's computational network. Associated with each computational task there is a control program that allows the control unit to evaluate the output of the bottom-up unit and trigger the corresponding control actions which may provoke a transition to a new state or select a new parameter set to perform a new cycle in the current state.

Adopting the BU-TD model as a valid model for the design of a generic module, several desiderable properties emerge. First, a clear separation is introduced between diagnostics and control what makes facilitates the reusability of diagnostics code and clarifies the control scheme. The existence of a control unit for every module allows for closing tight and reactive control loops without dening the existence of a control at a higher level. At the same time, the BU-TD model may serve as a basis element to articulate a distributed system approach. Even more, the existence of a communications unit in the model permits the system to cope with modules with different latencies so that communications are carried out asynchronously.

Figure 2 shows the control scheme of the Detection module as an illustration of these ideas, where states are depicted as circles and events as arrows. The init state for this module is the WAIT state denoted here by a double circle. Not included in this figure are the signals this module can receive or send. The module is activated by the reception of a DETECT signal from the TRACKER module that activates the head/face detection task. The outgoing signals allow other modules to know the results achieved at each state. Thus, this module can send six different signals corresponding respectively to success or failure in head detection, face detection and face recognition.

From the implementation point of view, the modules involved in DESEO share a common control and communications structure that ease the engineering of distributed perception-action systems. The functional modules comprising the system are programmed as threads that employ a message passing paradigm for event signaling among them. The integration of a new module into the system is done simply by registering the events the module (thread) may receive and/or send, detailing the allowed senders or recipients. The whole system then behaves as a multithreaded system, where the modules perform asynchronously and concurrently.

5 Experiments

As an experimental application of DESEO, a system oriented to detect, track and identify people in real time has been developed. The recognition process is made based on facial information. Within the last several years, numerous algorithms have been proposed for face recognition [6] and, although much progress has been made towards recognition of faces under small variations, there are not still reliable solutions for recognition under more extreme variations, for example, on illumination or in pose.

Due to the complex nature of person recognition by face, in our system an opportunistic solution to recognition has been implemented, based on a cascade scheme of confirmations. As the system is real time, its first objective is the detection of moving blobs (Figure 3), confirming that this is a person and detecting and isolating his head.

While the person is moving in the environment, the system will track him/her, centering the FOAW in what is supposed to be the head (Figure 4), waiting

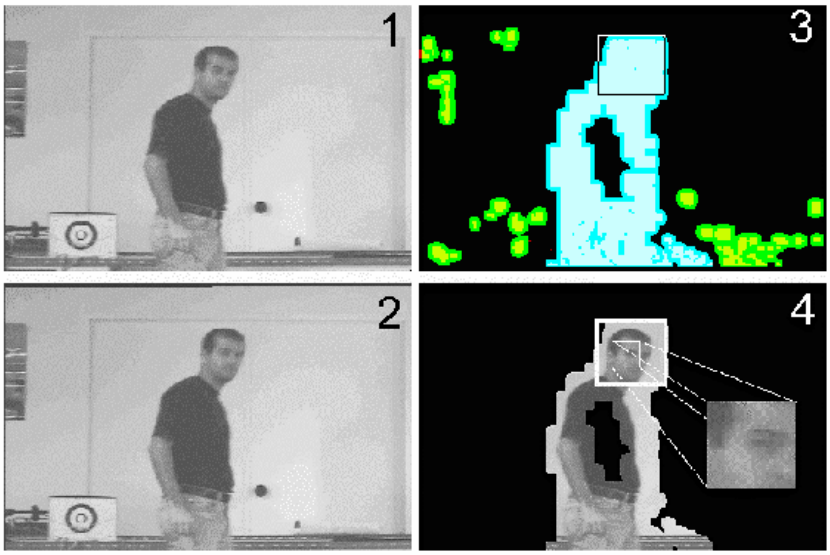

Fig. 3. Data images from optical flow analysis stage. Images 1 and 2 represent two consecutive frames that are the input to moving detection module. Image 3 shows a blob image provided by the optical flow algorithm. Image 4 represents the located head and the extracted pattern for the correlation algorithm.

opportunistically to obtain a good front view to identify the individual and confirming it while being tracked.

Once the person has been identified (Figure 5), the identity and an image on his/her face are sent to the supervisor. Depending on the supervisor policy, the tracking process can be continued, or in other case, its activity is shiffted towards other focus of attention areas. Due to the design of the system, several focus of attention can be tracked and followed simultaneously.

At this moment, an advanced prototype is being evaluated. An element which will be modified in order to obtain a more robust global behavior is related to the head detection procedure. At short term our objective is to modify it in order to obtain a better and more general head detection technique to more accurately locate the face, which in fact will result in a more robust face recognition. Another aspect of the system that will deserve further work is the immunity of the recognition procedure to scale variations due to different separations between the sensor and the individual. In the present implementation it is only partially solved through image warping.

The current prototype of the system runs in real-time on a Dual 350 MHz Pentium II PC, equipped with a PCI board containing a C80 DSP that are used as a slave processor for the tracking module.

Fig. 4. A set of frames selected from a sequence where a person is tracked by the correlation algorithm exclusively. Several automatically chosen patterns can be observed under the images (crossed boxes are still empty patterns). The black rectangle represents the position of the FOAW. Note that in absence of higher level information, the correlation module tries to keep in the pattern database the most stable views of the target. This explains the shift of the FOAW towards the neck of the subject.

Fig. 5. Some examples from the database with heads at different views, including frontal, lateral, back and no-head samples.

6 Conclusions

A methodology or approach may be termed as superior to others as the systems implemented following its guidelines tend to be easier to build, easier to maintain or simply show a better performance in some sense. However the benefits or drawbacks of using a certain methodology or approach in the design of perception-action systems can only be stated a posteriori, arising from the lessons learned after extensive experimentation. While we don't make definitive claims about the methodology used in the development of DESEO, we think we can draw two different types of conclusions.

On one side are the basic considerations, which permits the conception, design and development of perception-action systems in general and CVS in particular. It provides a versatile though conceptually simple architecture, conceived in a modular fashion in order to facilitate the incremental development and updating of the system. The generic BU-TD module promotes a clear separation of diagnostics and control, what facilitates code reusability and usually makes control simpler. On the other side is the application developed as a preliminary experimental evaluation of this approach. This has allowed us to solve a real-world complex problem as is the detection, tracking and recognition of individuals in indoor environments using an active vision approach. The system built as prototype, although still in development, shows real-time continuous operation using mainstream technology. The experimental results achieved so far are promising and seem to validate the design considerations on which DESEO has been based.

References

1. J.Y. Aloimonos, I. Weiss, and A. Bandyopadhay. Active vision. *Inter. Journal of Computer Vision*, pages 333–356, 1988.
2. C.S. Andersen. *A Framework for Control of a Camera Head*. PhD thesis, Laboratory of Image Analysis, Aalborg University, Denmark, 1996.
3. R. Bajcsy. Active perception. *Proceedings of IEEE*, 76:996–1005, 1988.
4. P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on PAMI*, 19(7):711–720, 1997.
5. M. Campani and A. Verri. Motion analysis from first-order propierties of optical flow. *CVGIP: Image Understanding*, 56(1):90–107, July 1992.
6. R. Chellappa R., C. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. *Proceedings IEEE*, 83(5):705–740, 1995.
7. J.L. Crowley and J.M. Bedrune. Integration and Control of Reactive Processes. *Proc. ECCV'94*, 47-58, Springer-Verlag, 1994.
8. J.L. Crowley and H.I. Christensen, editors. *Vision as Process*. ESPRIT Basic Research Series. Springer, 1995.
9. J.L. Crowley and F. Berard. Multi-Modal Tracking of Faces for Video Communications. *Proc. IEEE Conf. on Comput. Vision Patt. Recog.*, Puerto Rico, June 1997.
10. F.R.Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel. *Robust Statistics. The Approach Based on Influence Functions*. John Wiley & Sons Inc., New York, 1986.
11. C. Guerra, F.M. Hernandez, and J. Molina. A space-variant image structure for real-time purposes. implementation in a c80-based processing architecture. In J. Vitria A. Sanfeliu, editor, *Proc. VII National Simposium of the Spanish Assoc. Of Pattern Recognition and Imge Analysis (AERFAI)*, Barcelona, 1995.
12. G.D. Hager and K. Toyama. The XVision System: A General-Purpose Substrate for Portable Real-Time Vision Applications. *Computer Vision and Image Understanding*, 69(1):23–37, 1998.
13. R.E. Kahn, M.J. Swain, P.N. Prokopowicz, and R.J. Firby. Gesture Recognition Using the Perseus Architecture. *Proc. CVPR'96*, 1996.
14. J. Kosecka, R. Bajcsy and M. Mintz. Control of Visually Guided Behaviors GRASP Lab Tech Rep., num 367, Univ. of Pensilvania, 1993.
15. J. Kosecka. *A Framework for Modelling and Verifying Visually Guided Agents: Design, Analysis and Experiments*. PhD thesis, GRASP Lab, University of Pennsylvania, 1996.
16. Y. Moses, Y. Adini, and S. Ullman. Face recognition: The problem of compensating for changes in illumination direction. In *Proc. European Conf. on Computer Vision*, pages 286–296, 1994.
17. D. W. Murray, K. J. Bradshaw, P. F. McLauchlan, I. D. Reid, and P. M. Sharkey. Driving saccade to pursuit using image motion. *International Journal of Computer Vision*, 16(3):205–228, 1995.
18. K. Pahlavan, T. Uhlin, and J.O. Eklundh. Active vision as a methodology. In Y. Aloimonos, editor, *Active Vision*, Advances in Computer Science. Lawrence Erlbaum, 1993.
19. P.J. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of IEEE*, 77(1):81–97, January 1989.

20. M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, 1991.
21. S. Ullman. Visual Routines, *Cognition*, 18:97–159, 1984.
22. G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1990.
23. J. Yang and A. Waibel. A Real Time Face Tracker, *IEEE Workshop on Appl. Comput. Vision*, 142–147, Los Alamitos (CA), USA, 1996.