Towards Embeddable Vision Architectures for Human Computing

Marten den Uyl
CEO VicarVision and president SMRgroep
Amsterdam, The Netherlands
denuyl@vicarvision.nl

Abstract

Human Computing is about perceptive, anticipatory interfaces that support natural and intuitive human-computer interaction by understanding human behavior, emotions and social signaling. Yet, machine understanding of human behavior and emotion is limited and fragmented until this day. It is proposed that in order to manage the complexities of multimodal, multilevel and contextual machine perception of human behavior, an embedded systems approach towards the design of vision architectures for human computing seems advisable. The problems encountered in such an approach are illustrated from past and present development projects on vision systems for watching humans.

1 Introduction

Sentient Machine Research was founded in 1990 as an R&D company in AI with the aim to contribute to developing machines into sentient interaction partners. What makes a machine 'sentient' from the third person point of view, in the eye of a human observing the system, is first of all whether the system seems to understand us, seems 'human aware'. Many fascinating AI systems have been developed since e.g. Weizenbaum's Eliza back in 1966, that create an illusion of user awareness. But in fact, and in spite of much progress in recent years, machine understanding of human behavior and emotion is limited and fragmented until this day [Pantic et al 2006]. Machines struggle with perceiving the utterances, grimaces and gestures and often fail to understand the intentions and feelings of the humans they interact with. Whether or when any such machine should moreover be considered sentient in the first person view, i.e. whether 'it is something to be that machine', is still a challenging philosophical question, that in the end may well be decided by empirical means. Man and other animals are the only truly sentient machines we have seen so far, thus the alternative aim of Sentient Machine Research is to achieve a better understanding of human nature and experience by synthetic research methods, that is by building AI artifacts. VicarVision, a subsidiary of the SMRgroep, was founded in 2001 with the mission to develop computer vision systems for perceiving humans in video streams. The long term aim is to develop general purpose vision systems that allow robots to classify and label objects and events in mundane environments in human understandable terms.

A problem for Human Computing, the automatic sensing and understanding of human behavior in an *ambient intelligent* environment, is that on the one hand computer vision system architectures tend to expand over time in complexity, with more components, connections, subroutines and dependencies and an ever increasing appetite for faster computers. Yet, on the other hand, there are good reasons to try to make vision systems embeddable, efficient and small. First of all, it would of course be nice if competent vision systems could be run on hardware sufficiently small, cheap and energy efficient to be embedded in affordable robots or mobile devices. Even if price, size and energy consumption are not major concerns in the design of a vision system, it might be worthwhile to strive for a vision architecture that is embeddable in principle. The big challenge is that vision, particularly when watching humans, really is a very complex computational problem, requiring many specialist processes, extensive knowledge and massive raw computational power to perform real world tasks in real time. Thus, the basic requirements for embeddable systems -cheap, small and energy efficient- are directly challenged by the large and highly variable computational loads that come with vision tasks, even with the smartest possible algorithms. Further requirements for embedded systems follow from their embeddedness –by definition- within a host system with more functions and components to care about. Embedded systems generally need to be dependable, predictable and collaborative. Dependable because the performance of the system as a whole may critically depend on the proper functioning of the embedded system, there might not be a fall back when it fails. Predictability and collaborativeness of embedded systems are particularly important in multitasking systems with concurrent processing. Embedded systems that are part of such architectures are likely to share, and thus compete for, inherently limited resources such as access to communication channels, or instruction sets for action, or central memory and cpu cycles. Predictability of performance allows for control and in collaborative multitasking each process performing a task is optimally transparent –for predictability- and interruptible –for co-cooperativeness.

Embedded systems preferably are self-supporting with minimal need for external maintenance, support or upgrades. It is at least inconvenient if the host system must be serviced or repaired because some embedded system shows some malfunctioning. Obviously, being dependable, predictable, collaborative and self-supporting are often desirable features for any information system that humans interact with, even if there is no need yet to realize the system by embedded systems technology.

2 Does Human Computing really need embeddable system architectures?

Ambient Intelligence is about disembodiment, the computer has disappeared out of sight, the actual processing occurs somewhere 'in the back end'. Embeddableness of perceptual processing may then seem just an engineering issue for the efficient technical realization of Human Computing systems. However, some defining aspects of Human Computing imply that embeddability is highly relevant for theoretical issues. Human Computing is multimodal, multilevel and contextual and it must proceed in real time. Various sensory modalities may contribute to understanding behavior besides vision; audio, tactile even olfactory modalities can be used to sense a human. Actually, Human Computing may also use senses underdeveloped in humans such as perception of electromagnetic fields, infrared radiation, ultrasound, etc. Within a single modality such as vision a number of analysis channels may be distinguished, as for example, one may choose to watch the face, or the body, or the eyes, or the gesticulating hands of a person. And sometimes multiple instantiations of analytic processes are required, e.g. when observing two people interacting. Thus, Human Computing requires concurrent processing architectures, and in such architectures some compromise must be made between dedicated and shared resources per modality -or rather per processing channel. If all perceptual processes have fully dedicated resources, this gives maximal robustness against interference by other processes, at the cost of extreme redundancy of computational resources that will remain idle much of the time, while the processes they support are not triggered by current inputs.

Human Computing is multi-level in the sense that the full path must be covered from registration by sensors, pixels from cameras or soundwaves from microphones, through various stages of processing, until arriving at some understanding of the intentions, actions and experiences of the humans observed. Typically, the lower levels have dedicated resources while the highest levels share resources. The balance that a given architecture strikes between dedicated and shared resources is related to the classical issue of early versus late fusion in multimodal sensory integration. Perhaps somewhat counter intuitively, it is much more difficult to perform early fusion in an architecture where low level processes must share resources. [Nock et al 2004] provide an interesting analysis of a classical Gestalt principle of perception, the detection of common cause by temporal contiguity, on the lowest possible level of multimodal analysis, temporal correlation between pixel and sound wave intensities. In shared resource models, computation of these correlations is very costly, because it requires explicit computation and memory buffering of large temporal index structures. In a dedicated resource model where image and sound are processed in parallel by transparently embedded systems, the correlation patterns for detecting common cause can be obtained at little extra cost by temporal correlation over small sets of system processing load parameters. The aim of Human Computing is high level analysis, a proper response requires an understanding of the meaning of the human behavior observed. A framework for high level analysis of action, expression and experience can be found in emotion theory [Frijda 1986], where it is proposed that event appraisal results from the matching of situation aspects to active concerns. From such a formulation it directly follows that some knowledge of context is required to be able to infer what concerns may be active in the person observed. Is this person showing a sad face because she has just received unpleasant news, or because she was asked to pose a sad face?

Context in fact is all important in Human Computing. Not just because at the highest level affect and behavior can only be understood to the extent one knows, i.e. has at least some general or default model of 'where a person comes from' and what moves the person in the situation. The extensive psychological literature on priming, the role of expectation and context effects in perception indicates that for humans, perception is context dependent on all levels. Perception is not a one-way processing stream from input to meaning but rather a cyclical process steered by top down anticipation as well as the bottom up input data stream [Neisser1976]. First time students of neuroanatomy are often surprised to learn that the neural circuitry for the vision system does not consist of just an upwards series of processing stages or projection fields, but that almost as much circuitry is dedicated to the downward modulation of these processing stages. The basic reason is that only through anticipation the complexities in real time visual –or auditory- processing can be mastered. Just as efficient tracking of a moving object requires some form of prediction in x,y coordinates, so does understanding of say the current face expression of the subject requires some form of prediction of direction in affective coordinates –dimensions or categories. And sensitive evaluation of the expression on a particular kind of face – young or old, Asian or European- requires a momentary specialization, an anticipation based on similar expressions seen on similar faces before. It has proven difficult to produce computational vision systems that come anywhere near the ability of natural vision for anticipatory and adaptive perceptual processing. The basic hunch is that one should first be able to realize vision components as predictable and self-supporting embedded systems, before the added complexities that come with adaptivity and context dependency can be successfully dealt with.

The conflicts that arise when vision architectures increase in complexity and size, while embedded systems require a small footprint and strong encapsulation, will be illustrated

by a short description of some vision systems that have been developed at SMR and VicarVision over the years.

3 Pires (1997)

PIRES (PIcture REtrieval System) is a forensic face recognition system that accepts pictures, 'mug shots' and constructs an extensive face index representation for each individual portrait. This index structure allows for search by image in a person database. When presented with a portrait as query, similar portraits will be found. PIRES also produces a detailed description of the characteristics of the face and is able to fill most of the Dutch police standard person description, *signalement,* reporting form with about 40 facial feature categories –from big/small nose to hairstyles and ethnic origin.

Figure 1: PIRES face analysis

The PIRES system performs the face analysis task in a three step perceptual processing architecture: *Find, Frame, Featurize.*

1) Face finding, the detection and localization of the face in the image is performed by standard template search. A template is constructed by averaging the grey pixel values of a set of aligned faces. One or more templates are evaluated in discrete steps over the image at a number of scales and the best match position is selected.
2) A visual object is 'framed' by establishing a correspondence between localised features of the image and an object topology model. PIRES employs a small set of local feature templates to find the position of eyes, nose and mouth in the portrait. From these positions the full face topology can be estimated.
3) The features of the framed face are derived by a set of neural network classifiers trained on various sets of annotated 'clips' taken from the face.

For each portrait a long representation vector is constructed, consisting of neural network hidden node activation values, explicit classifier labels and face topology coordinates. An associative search engine is used to find best matches for a newly analyzed portrait in a list of previously indexed portraits. PIRES performs quite reasonable as a forensic face analysis system under a limited range of conditions. The system can only handle high quality frontal images, but achieves respectable recognition rates / retrieval within best n matches, on police image databases. PIRES in the 1997 implementation may not be a good candidate for an embedded system because the template based face framing routine is limited to frontal faces and even with frontal faces sometimes framing errors are made. However, there are no principled reasons why PIRES could not be implemented as an embedded system, say for automatic indexing of portraits in a high end digital camera, if performance is improved within the same architecture. The three basic processing steps are performed by modules that each can be encapsulated and used in a predictable sequence of operations.

4 Vicar (2001)

The aim of the VICAR Video Explorer system -developed in the VICAR (Video Indexing, Classification, Annotation and Retrieval) HPCN/IST EU project- is to provide for semantic indexing and content based search for large amounts of video footage. The field of operation is the professional video archive market (broadcasting, movie productions and agencies, security). Types of contents indexed by VICAR include:

- shot detection and camera motion;
- moving object detection and segmentation;
- setting classification, evaluate the general setting and background of images [Israël et al 2004];
- object recognition; VICAR contains object recognizers for faces, cars and horses [Noorman et al 2002];
- recognition of individuals, VIPFinder recognizes faces from a short list of famous persons;
- classification of behaviors, e.g. walking, running, limping.

While VICAR demonstrated the feasibility in principle of indexing contents at many levels, performance at some lev-

els was not yet up to operational use in 2001. For present purposes it is of interest to look at a sketch of -part of- the VICAR system architecture.

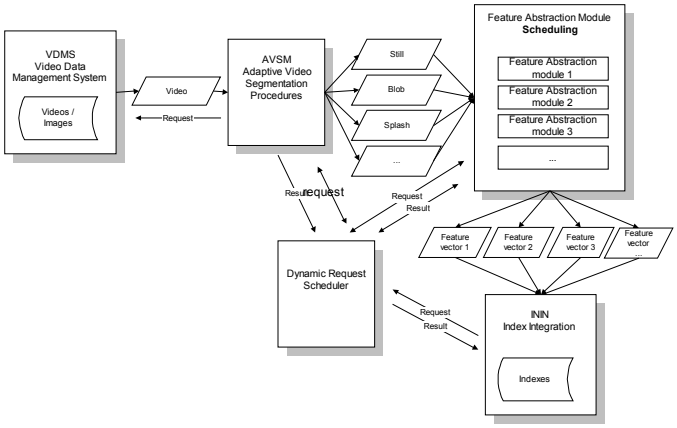

Figure 2: VICAR architecture, flow of control.

Even a superficial inspection of figure 2 suggests that the VICAR architecture looks like a worst case for an embeddable system, with many modules and datatypes and the control of processing going in all directions. Actually, VICAR was not designed to run as an embedded system, but on almost the opposite kind of platform, a multi cpu supercomputer. To review some of the problems, first again comes the need for computational power, e.g. some of the object recognition procedures at that time required many seconds of cpu time for processing a single image. Next, specific perceptual processes require specific subsets of the x,y,t pixel volume that makes up a videostream, e.g. when analyzing behavior of a walking person, the person must be tracked and the relevant pixels must be segmented from the stream. This requires either complex dynamic memory management, or huge amounts of random access memory. Both options are not popular with embedded system engineers. Then, since vision is data-driven, a new scene may spawn many perceptual processes, that will compete for available resources, while many interdependencies may exist between different levels of content analysis. Together, these complications make the processing that occurs within the architecture rather unpredictable. And since time-constrained operation –even if not real time- makes it unsure whether the relevant processes will all have managed to run to completion, the results of perceptual analysis may not be very dependable.

5 FaceReader (2005)

FaceReader is a commercially available product for real time analysis of facial expression [Den Uyl and Van Kuilenburg 2005]. FaceReader fits a face in a video stream with a mask computed by an active appearance model [Cootes and Taylor 2000] and derives persistent –gender, age, ethnicity- and changing features, particularly the emotional expression of the face. Expressions are classified in 7 emotion categories, 6 basic emotions -happy, sad, angry, surprised, scared and disgusted- and neutral.

Figure 3: FaceReader interface

FaceReader employs the same three step –find, frame, featurize- architecture as PIRES, though some of the steps have changed considerably. Where PIRES uses static templates for finding faces, FaceReader uses one or more 'Active Templates' [Song and Poggio 1998] for finding a face in the image. The Active Template Method moves a set of deformable face templates over an image, returning the most likely face position. To frame the face, FaceReader uses an Active Appearance Model [Cootes and Taylor 2000], able to produce good fits over a wide range of variation in persons and lighting, orientation and expression. Face features are derived by neural networks, trained on the appearance vector –the list of about 100 appearance parameters found for the best fit mask [Van Kuilenburg et al 2005].

The FaceReader architecture (Figure 4) shows a principled distinction between the online or execution model and the offline training environment. This contributes to the embeddableness of the online system, since different configurations, optimized for different tasks, can be obtained by replacing modules. FaceReader is an online system, expecting a face in a video stream, whereas PIRES is an event-driven system, triggered by the presentation of a still image. This is reflected by the inbound arrows in the FaceReader architecture. For face finding, once a face is found, a tracking subroutine will start tracking the face for further speed optimization, currently the FaceReader can do the full find, fit and classify cycle at around 20 frames per second on a quick PC. The inbound arrow on the classification box indicates temporal integration; identity, current expression and other feature estimates are based –by default- on temporal integration over a series of images. Although the active ap-

pearance modeling approach is particularly computationally intensive, even after extensive optimization, FaceReader could be implemented as an embedded system about as well as PIRES.

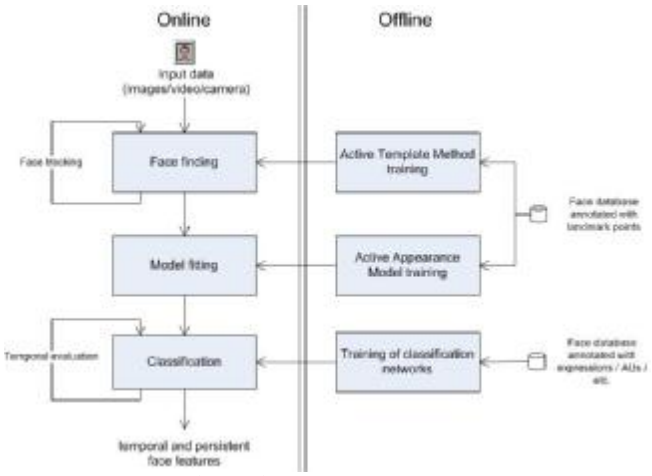

Figure 4: FaceReader architecture.

5 BodyReader (2006)

A system currently under development at VicarVision is the BodyReader, which aims to give an estimate of body pose – i.e. the localization of body, head and limbs- of a moving person in real time [Van der Meer and Metz 2006]. The global architecture is again a three step -find, frame, featurize- perceptual process, with a strict separation between online and offline facilities, just as for FaceReader. Almost all the analysis processes are however entirely different. For finding the moving body, a standard motion differential method puts a box around a suitably sized moving blob in the image. Framing the body within the box is a three step process in itself. First a neural network makes a rough guess at the location of 14 anchor points on the body (see figure 5). Then a PCA model trained on a body topology reference database is used to move the anchor points to more plausible positions. Lastly, for each anchor point placement a local refinement is attempted by an active search method. Static pose features –arms up or down- can be derived trivially from the framing model, the more interesting class is that of dynamic pose features, that is, body movements.

Note that the BodyReader can only track the movements of one person at a time, but it might well do that as an embedded system.

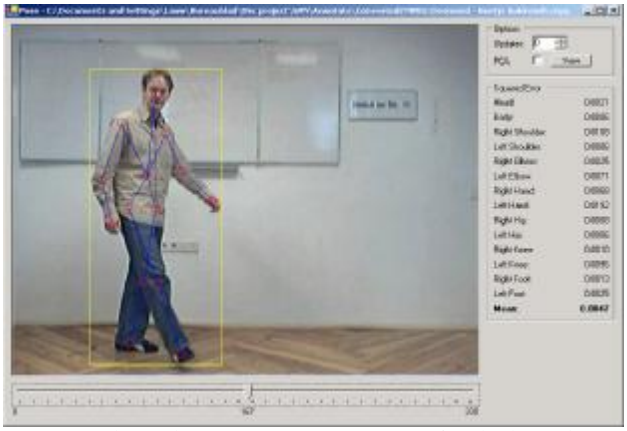

Figure 5: BodyReader: automatic annotation of 14 body anchor points.

6 AIVOS (2006)

AIVOS –Architecture for Intelligent Video Observation Systems- is a project-under-development at VicarVision aimed at developing vision systems that can fulfill a range of security and surveillance tasks. This is about as broad a task as the VICAR Video Explorer task of indexing any kind of video footage for archive. Even though most surveillance tapes show little of interest happening at all, pretty much anything worth reporting could happen at any moment. In fact it is mostly mundane human behavior that is to be observed, signaled and reported. Thus the main 'find' routine in AIVOS is motion based object detection, as in the BodyReader, except that surveillance requires that multiple moving objects can be detected and tracked, possibly with

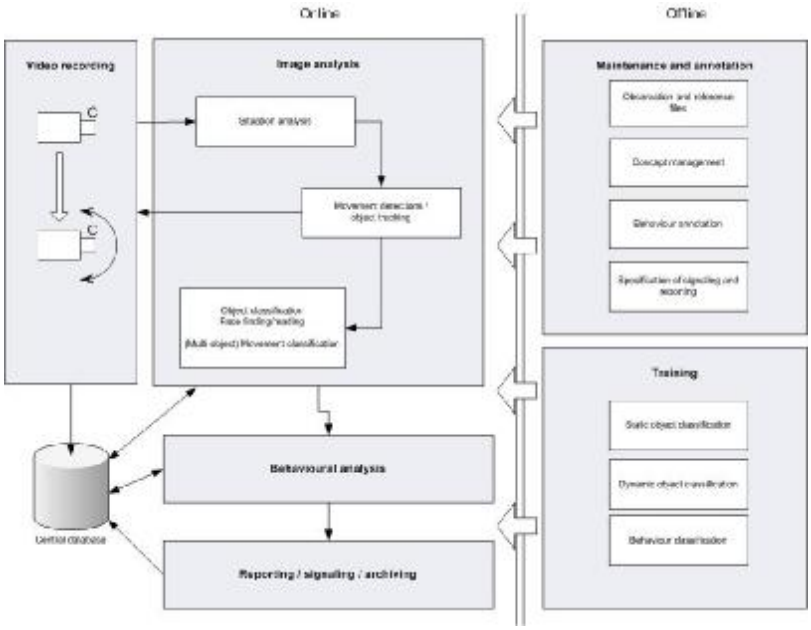

Figure 6: AIVOS main components

an active camera. Framing the body of moving persons is performed by the BodyReader, just as their faces are modeled and classified by FaceReader, when they are detected to be within scope of the face fitting appearance models.

AIVOS then is the framework for setting up virtual vision engines: object controlled perceptual analysis channels, that direct what pixels they want from the input and apply perceptual processes to this pixel stream, relevant to the object in the image and the purposes of the observation system. When the visual input is rich, a number of people moving around, managing a set of virtual vision systems that share physical resources becomes an intricate problem that easily leads to overload even for high capacity systems. It is not necessarily the case that a system is embeddable, even if all of its components are embeddable. But an AIVOS type system might well be made embeddable, by strictly limiting its span of visual attention, the number of virtual vision engines it can run concurrently.

7 Conclusion

Human Computing is all about managing the complexities of multimodal and multilevel perceptual processing, adaptive and context sensitive and in real time. Aiming for systems that are embeddable in principle is a basic 'divide and conquer' strategy. Only out of well-behaved, transparent and self-supporting components can we hope to be able to build such complex systems. A straightforward conclusion appears to be that a vision architecture is embeddable if it performs a single chain of tasks on a single class of visual objects and if its basic resources –cpu cycles and memory or knowledge and data access- can be managed 'on board'. This tends to limit candidate vision architectures for embedding to 'one trick ponies', which in turn tend to have limited use as an embedded vision system for general purpose hosts like robots or mobile devices; their users would like to see them do many tricks. A way out of this dilemma might be to develop vision architectures that support multiple virtual vision engines. This seems not so much a matter of developing new vision algorithms, rather of better compositions with existing algorithms, embedded or not .

It has been mentioned repeatedly that embedded systems should preferably be self-supporting. That implies in fact, in the case of vision, that systems should be able to teach themselves to see new things. This seems the aspect where current vision architectures are still the furthest away from target. It may be a long time before one can install the off-line training environment for a vision system on board and feel reasonably confident that the host will know what to do with it.

References

[Cootes and Taylor 2000] Tim J. Cootes,. and C. J. Taylor, (2000). Statistical models of appearance for computer vision. Technical report, University of Manchester.

[Frijda 1986] Nico H. Frijda, (1986) "The Emotions". Cambridge University Press: Studies in Emotion and Social Interaction series.

[Israël et al 2004] Menno Israël, Egon L. van den Broek, Peter van der Putten, and Marten J. den Uyl, (2004). Automating the construction of scene classifiers for content-based video retrieval. In L. Khan and V.A. Petrushin (Eds.), *Proceeding of the Fifth International Workshop on Multimedia Data Mining (MDM/KDD'04)*, p. 38-47. August 22, Seattle, WA - USA.

[Van Kuilenburg et al 2005] Hans van Kuilenburg, Marco Wiering and Marten J. den Uyl. A model based method for automatic facial expression recognition. Machine Learning: ECML 2005: *16th European Conference on Machine Learning*, Porto, Portugal, October 3-7, 2005. Proceedings pp. 194 - 205 Springer-Verlag GmbH.

[Van der Meer and Metz 2006] Desmond van der Meer and Lauwerens Metz (2006). AIVOS and Human Pose Estimation, internal report VicarVision/Delft University of Technology.

[Nock et al 2004] Harriet J. Nock, Giridharan Iyengar, Chalapathy Neti. Multimodal Processing by Finding Common Cause. Communications of the ACM, January 2004, Vol 47, 51-56.

[Neisser 1976] Ulric Neisser. Cognition and Reality. (1976) W.H. Freeman and Company, San Francisco.

[Noorman et al 2002] Merel Noorman, Kai Otto, Marten J. den Uyl, Rein van den Boomgaard. Horse Recognition: A General Approach to Object Recognition. Proceedings of the 12th Portuguese Conference on Pattern Recognition, 2002

[Pantic et al 2006] Maja Pantic, Alex Pentland, Anton Nijholt, Thomas Huang. Human Computing and Machine Understanding of Human Behavior: A Survey. International Conference on Multimodal Interfaces 2006.

[Sung and Poggio 1998] K.K. Sung and Tomaso Poggio. Example-based learning for view-based human face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1):39–51, 1998.

[Den Uyl and Van Kuilenburg 2005] Marten J. den Uyl, Hans van Kuilenburg, (2005). The FaceReader: Online facial expression recognition. Proceedings of Measuring Behaviour 2005, 5th *International Conference on Methods and Techniques in Behavioural Research*, 589-590.