Animate Vision in a Rich Environment

Tomas Uhlin **Jan-Olof Eklundh**
Computational Vision and Active Perception Laboratory (CVAP)
Department of Numerical Analysis and Computing Science
KTH (Royal Institute of Technology), S-100 44 Stockholm, Sweden

Abstract

Most research in computer vision has been directed towards minimalistic approaches, in which problems are addressed on how properties of the environment can be computed from as little information as possible.

Although such approaches may be scientifically well motivated they have only resulted in limited progress towards our understanding of seeing systems. Ballard, Bajcsy and others have pointed out the importance of vision being an active process which is tightly connected to behaviors. We support this thought and also propose that utilizing that the world is rich on information is essential.

We develop this idea to show how attention and figure-ground segmentation by an active observer using multiple cues can be separated from analyzing and recognizing what is seen in a consistent way. Continuous operation over time and early use of three dimensional cues are important in this context. We illustrate our proposed approach by some experiments on a real-time active system.

1 Introduction

Vision is a sense by which seeing creatures acquire information about a dynamically changing environment and thereby guide many of their behaviors and actions. Computer vision research aims at understanding and developing computer based systems with such capabilities. Despite extensive efforts for more than three decades we still seem to be very far from such a goal. Although there exists ample knowledge of how information about the environment can be computed from visual cues, we see little progress towards what can be called seeing systems. Research on *active* or *animate vision* [Bajcsy, 1985; Ballard, 1989, 1991] has pointed out that the major reason for this is that vision, as we know it from biology, is an active process and that traditional computer vision approaches take no heed of this fact.

Ballard [1989, 1991] analyses this discrepancy and its consequences. Here we will further Ballard's arguments and discuss some additional issues which we believe are crucial. We will also report on recent progress towards the realization of animate vision systems. Emphasizing the strong ties between vision and behaviors Ballard particularly considers the need for gaze control and what he terms "quickly computable features"[1]. The point that we want to stress in this context is that the real world is rich on information and that a multitude of such features can be computed when needed. We will argue that this suggests a paradigm of attentional mechanisms coupled to possibly *independent* mechanisms for deriving scene characteristics and information about objects. Notably this implies that the environment itself influences what should be computed. We will also discuss possible ways of performing such computations and describe some illustrating experiments.

2 The world is its own repository

Minimal information vs. salient characteristics
Current research in computer vision is largely concerned with how information about the world can be derived from single cues, like motion, stereo or texture. The approaches are generally *minimalistic*[2]: problems are addressed on how properties of the environment, like motion and structure, can be determined from *as little information* as possible. Such considerations may be well motivated scientifically and adapt to the standards in psychophysics. To understand the importance of a specific piece of information influences from other sources must be eliminated or controlled. Hence, there is a wealth of results on e.g. how many points or lines are needed to compute scene or object structure from a set of image frames. We contend that however important it may be to understand such limit cases, it hardly brings

[1] He also discusses aspects of memory and learning but these items are beyond the scope of the current paper.

[2] Ballard [Ballard, 1991] used this term referring to the assumed generality of the models.

us closer to achieving a seeing system. More precisely such a system should function in a real environment, which inevitably will contain a rich amount of information and therefore finding and maintaining a stable perception these features will often be more difficult than computing the scene structure from them.

This is well understood by computer vision researchers, who approach the problem by performing grouping operations, including uncertainties and the like. However, if this is done on the basis of prespecified features, as is customary, the available visual information is not used to its advantage. What makes it possible to detect an object or observe a feature, and even what constitutes them, depends very much on the situation and the surrounding environment. A grey lump of materia may be observed as being an object by its color if it rests on or moves over a white background. However, if e.g. due to the relative displacement or a change of illumination the background also becomes grey, then this feature is no longer appropriate. The optic flow pattern or the binocular disparities may on the other hand then be used to select out the lump as an "object". The key point is that it is the world that determines both what is useful for the computations and what constitutes an object. So, while the complexity of the real world poses problems for some feature detectors, it may well make the job for others quite easy. To utilize this, we must be prepared to compute a number of different features simultaneously and in the particular situation use the ones that stand out sufficiently well against their surroundings[3].

It is also important to note that this lump of materia by standing out from its environment becomes an object and thereby gets an *identity*, whether we have a memorized category for it or not. We can hence ascribe properties to this entity, based on what we can observe, rather than on some model-based features given a priori. Such considerations are certainly well established in psychology and are elaborated e.g. by Gibson [1979], but computer vision approaches have hitherto not been based on them.

What to compute? At this time it may be enlightening to discuss what kind of features are of interest. Many of the features traditionally computed and used in machine vision are such that they can be interpreted geometrically and that they have implications on the three dimensional structure of objects. Such features are for instance edges and corners, which especially can be used to characterize man-made objects. Other approaches using color to recognize known objects, and cues that directly estimate three dimensional object shape such as stereo and shape-from-X methods, apply in more general contexts, but these techniques have mainly been used for scene reconstruction, or specific tasks like obstacle avoidance, or model-based recognition, and hardly ever yet for behavioral vision.

It is clear that such systems based on specialized features can be used only in limited environments (that is, almost exactly the ones that they were designed for) and efforts to generalize such approaches have not yet proven to be successful. Although they may provide the most straight forward way to achieve results on scene understanding today, it is hardly appropriate for a system that is to function in a more unpredictable environment. Moreover, even if there are different views on the appropriateness of the reconstructionist approach (see [Tarr and Black, 1994]), it is yet to be shown that it can be used by a "seeing system". In particular, it postpones the solution of the figure-ground problem to later stages.

The features utilized by an animate vision system must not necessarily be very different even though three dimensional features are especially attractive for a fixating system. In general the features used are based on well-established physical and geometric knowledge. However, a few things are worth noting concerning their computation. The first is that, if we have already identified "something" as being of interest, then we have a basis for computing global characteristics of it. This can provide a guide to grouping e.g. luminance, shape or motion features in a very specific way. We can observe that an object is elongated or mainly yellow and use this as its characteristic. Secondly, a system able to exploit the richness of the world must have the capability of computing a large set of features and also to choose between them. This puts an emphasis on control aspects and on integrated use of multiple cues. We will return to these issues later, discussing systems aspects. Finally, given that a system has these capabilities, we obtain a way of deriving or learning invariant or quasi-invariant features *from the world*, which complements the more narrow approach of using a priori known invariants often used today.

Attention and expectations What we have discussed so far in this section can be regarded as an attentional step, where objects or features pop up from the environment by being different from nearby structure. Such mechanisms have been considered in computer vision by e.g. Culhane and Tsotsos [1992], who argue for the need for attention to overcome the complexity offered by visual tasks in the real world. Here, we want to stress that attention offers a way to utilize what can be observed from the world and also a mechanism that tells us that we see "something" and at what location this "something" is at the moment. We will later return to how these computations can be performed. Let us now just point out that they depend critically upon the use of multiple cues.

Of course, the process described could be mediated by

[3]It is worth stressing that this is true also for "objects" like the ground plane or the sky, as long as there is a way of characterizing them. We will use the word "object" in a somewhat fuzzy sense to denote any configuration of interest in the environment.

various expectations even if the saliency of the computational features nevertheless is essential. Notably there are two cases when expectations play a central role. One, mentioned by Ballard, is when we have specific knowledge about a particular object, like the color of "my own coffee cup". Another case is when we keep track of something that we just noticed. The expectation then comes from our observations rather than from memory. However, in both cases we can do with very simple computations: we can either establish some coarse global feature, e.g. if the object has some dominating color, texture or shape feature, or adapt our computations to finding some quite characteristic local feature that we either know since before or just observed. Again, it is the situation at hand that tunes our computations, not some prespecified agglomeration of a set of bottom-up structures.

3 The systems approach

As is obvious from our previous discussion a vision system capable of providing robust behaviors in a complex and dynamically changing environment must be considered in its entirety. Such systems oriented approaches have recently been suggested by various researchers, see e.g. [Crowley and Christensen, 1995], but their impact on computer vision research is yet limited. Several important questions arise in this context. The most central one is of course how we model the seeing agent displaying the behaviors guided by its vision. This is a very far-reaching question that actually touches upon our understanding of intelligence as such. We must therefore narrow the scope considerably. With today's knowledge we have to limit ourselves to discussing a visual observer who is able to look at, identify, and to some extent recognize objects and structures, and on the basis of that capable of reactive behaviors like obstacle avoidance during navigation, and certain limited active tasks, like searching for more or less well-specified known features and categories. However, even in such a limited scenario two major questions arise: which algorithms and features should be used for acceptable system performance (i.e. for solving the tasks) and how can the system choose between and integrate the different features?

Feature selection Deciding which features to incorporate and if an algorithm for deriving them is good or not should mainly be determined at the systems level. It is virtually impossible to evaluate how well a single algorithm performs in extracting a feature when it is studied in isolation. Of course, it is possible to measure performance and accuracy in absolute terms, like metric reconstruction errors, but the meaning of such measurements in a particular task is not obvious, neither is it clear how we compare them for selecting between algorithms or features. Strictly speaking all efforts in trying to answer such questions about an algorithm which is functional in isolation will be meaningless for judging their usefulness in a complete system. However, if there exists a system performing certain tasks, the true impact of incorporating a new feature and a change in the respective algorithm may be examined.

When an algorithm is evaluated in terms of overall system performance an important problem arises since we at the same time are examining how to control the many features. The result of such an evaluation will strongly depend on the specific system it is being tested in. During system development this will result in a reevaluation of algorithms that have previously been examined. A formerly useful feature may well prove to have little or no effect, and even be harmful to the performance of the system, while what was discarded before may come in handy later. An open systems design, say, using a layered architecture, is therefore necessary.

Control Next question is how the system should control its use of all these features. How can it decide that this is currently a better descriptor of what it is looking at? How does this depend on the expected action of the system? These are questions that will go unanswered in this paper, but are essential in the continuation of the approach outlined here. Individual features can be computed in parallel, but conjunctions of features are sometimes derived in a sequential step, if they are complex enough. Such interactions between parallel and sequential computations are well-known in the literature on human attention, and they should therefore hardly be unexpected in computer vision systems.

Continuous operation A final general point to make on the systems approach, obvious from the focus on behaviors, is that the time aspect is crucial. A seeing system naturally functions over time. Processing is, in principle, continuous and the system must respond to events and changes in the environment as they occur. This puts an emphasis on real-time processing that goes beyond the mere goal of fast algorithms. The important point is rather that the system continuously receives input which it uses to solve certain tasks, that in turn also may vary with time. These thoughts have been elaborated by other researchers and we will here just note that they necessitate experimentation not only with algorithms but also with real systems.

4 The Visual Front-End

Current research on early processes in computer vision to increasing extent seems to converge on certain basic principles for early vision which fits well in with our previous considerations. Generally speaking most approaches rely on the computation of directional derivatives of low orders and combinations or functions thereof, computations performed at several different scales. The manner in which this is done varies: some approaches use Gaussian or Gabor filters, others tunable filters of

a more general nature, wavelets or more sophisticated anisotropic models. It is beyond the scope of this paper to discuss these different techniques in detail, we refer the reader to the literature on the topic. However, we note that such methods in general are indeed very appropriate for developing systems of the type we are aiming at. By implementing a first layer of retinotopic processing, a Visual Front-End in the terminology as proposed by Koenderink and van Doorn(1987), we obtain a highly efficient implementation at the same time as we can base our low level computations on state-of-the-art techniques. We also get a scalable design, which allows the extensions we foresee, without cumbersome additions.

More precisely, by computing a set of low order derivatives at multiple scales in a VFE layer, we obtain output that can be shared by all our subsequent modules to derive monocular, binocular and motion cues at later stages. Without arguing for or against whether such a model provides the most reasonable architecture for a computer vision system, we observe that at least it maps well onto already existing hardware. Existing pipeline and signal processors are well suited for such retinotopic computations, but less so for the more general and less image oriented ensuing computations, which usually are performed in coarse grained parallelism. In the end and in our general spirit we may want to develop other pathways for more direct computations of, say time-to-collision or certain scene characteristics, but currently we favor the VFE structure. The details of the approach we have used in our experiments is described elsewhere, see e.g. [Gårding and Lindeberg, 1995].

5 Summary of the basic principles

Our argument so far, building upon the ideas of Ballard [1991], has been that to obtain a machine based seeing system that uses vision to interact with its environment we should utilize the fact that the world is rich on information, rather than minimum information approaches. Although the latter provide a theoretical foundation they do not address how the information can be extracted, neither do they use available information fully. Consequences of this argument are that such a system should

- be capable of using and integrating many cues
- work continuously over time
- base its decisions and actions on what cues are salient at the particular instance and location
- separate between what attracts the attention to something and the analysis of it, as well as keeping track of it.

These views fit well with what other researchers have suggested. Ballard [1991] in his concluding discussion proposes the idea of computing features on demand and adaptively. Tsotsos and his co-workers have pin-pointed attentional mechanisms as essential also in machine vision. Moreover, since a system of the type we are discussing necessarily must be able to control gaze, our scheme adapts well to recent theories that recognition mainly is view-based, see e.g. [Bültoff and Edelman, 1992]. Finally, our approach allows early inclusion of three dimensional information as favored by Nakayama and his co-workers, see e.g. [Nakayama and Silverman, 1988; Shimojo et al., 1988].

We shall now describe an experimental system by which we are able to demonstrate some of these ideas, which however with current hardware are somewhat difficult to implement.

6 An experimental system: the mobile observer

The design of a fully autonomous mobile observer is a major long-term goal of our research. So far fundamental skills in terms of fixation, target pursuit and target discrimination, have been implemented. In doing so we have followed the philosophy of what has been said earlier, both in terms of systems design and real-time considerations. At this time we would like to stress that since fixation is not itself the final goal, but that it should function in cooperation with other parts of a larger system. Unlike what is customary in much of other recent work, we want processes involved in fixation to provide much more than just directing gaze. This system is at the moment implemented partly in real-time on an existing mobile platform, and partly as a post-processing stage working on images taken in real-time using the former processes.

The system includes the integration of three cues for target selection and target discrimination. These are used by the moving observer to smoothly pursue moving or stationary targets binocularly while maintaining vergence. Mechanisms for discovering moving targets also form integral and vital parts of the system, since that provides means of attention, without which clues to changing the state of the system do not exist. Hence, we have two components, one to maintain attention, and another one to find and select new locations to attend to. Moreover, these must function in parallel, while otherwise the system would be purely reactive and without choices. We have implemented these two components in the form of a pursuit and a motion detection mechanism[4], thus obtaining what we believe is the most basic behavioral level for an active observer. This is shown schematically in Figure 1.

6.1 Attention and Smooth Pursuit

A key feature of the system lies in its ability to smoothly pursue an arbitrary target that is described by its location, extent and visual appearance. We would like to

[4] Currently, motion provides the only cue to changing attention, but other cues can easily be incorporated.

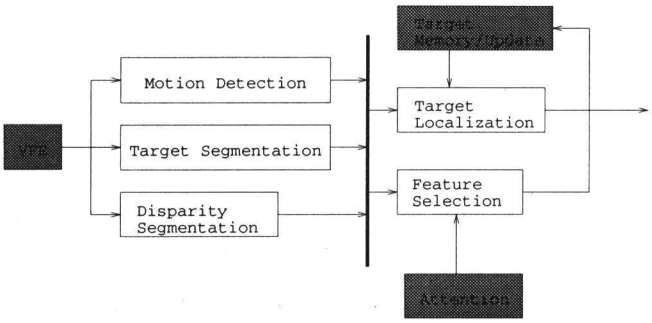

Figure 1: The system implementation is shown schematically.

stress here that no a priori knowledge about the target's visual appearance, i.e. texture or shape, is built into the system, although we indeed leave open that additional constraints or knowledge can be included dynamically as information becomes available, and similarly that such information can be excluded as it becomes obsolete. The features that are seen, are assumed by the pursuit module to change smoothly. Non smooth changes will trigger attentional mechanisms.

As the basis for this implementation we have chosen a coarse to fine correlation scheme similar to the one reported in [Pahlavan et al., 1993]. This technique works very well when no occlusions are present. We will here present an extension handling occlusions as well.

In order to take care of occluding objects and distracting things in the background, motion detection is integrated to filter out parts of the scene that can be parts of the target, namely those that are moving. We will see later in the experiments that this is not enough in many cases when there are occluding objects that are themselves moving, since they will also be detected as moving. Disparity, which is an essential component in our binocular system can in these situations aid in depth discrimination by also providing a clue to where occluding objects may lie.

Issues on Iterative Algorithms and "Anytime Vision" In the motion literature quite a few iterative algorithms have been presented. In a real-time active system it is of importance that the modules that control the system respond within a given time, although this time demand will vary from task to task, and no general statement can be made about a maximum time without knowledge of the task and the type of control involved. What can be said, though, is that this time constraint applies to all steps in the processing, unless the system specifically deals with the modules that involve slower computations. More specifically, a module that runs in real-time and in turn depends on the input from a slower module must explicitly deal with the fact that it is going to receive *old data*. For instance, many algorithms in the motion segmentation literature involve an initialization step to produce an initial segmentation in the beginning of a sequence, a step which is many times slower than the ensuing updating of the segmentation as the sequence evolves. Unless such algorithms explicitly deal with the initial delay of "bootstrap", the effective real-time of such algorithms in an active vision system is that of the bootstrap time and not the consequent higher processing rate.

We believe in the notion that information should be made available as soon as possible, so that the system can react to the stimuli in time. Even though an early reaction may be wrong, it is often better than no reaction at all. Hence, the "bootstrap" should be made over time while the system is functioning, and data be made available during this phase so that the system always has access to current data, even if in the beginning it is not of the best quality. In general any part of an active real-time system should have access to data whenever it is needed and modules involved in the system should be designed to provide this. We refer to this as *Anytime Vision*[5].

6.2 System overview

The system consists of a visual front-end, three feature maps, algorithms for feature selection, target localization and target memory. This is shown schematically in Figure 1, omitting some connections for readability. An attention module is provided that currently control the choice of moving target and does so by controlling the selection of features. Omitted are connections for tuning filters based on target characteristics.

Memory and feature selection The system keeps a memory of where it expects the target to be[6] and what its motion parameters are. To update the memory the system selects what features to extract in terms of disparity, target and background motion. The memory will update itself by including pixels to, and excluding from, the target, and by updating an affine motion model of the pursued target.

Target localization The target is localized by matching the memory contents with the features coming from the feature-maps. The feature-maps are tuned depending on the current expectations about the target to emphasize the expectation.

Feature-maps (cf. Figures 2 and 6)

- **Motion detection** involves computation of a background affine motion model based on normal flow, and subsequent residual calculation.
- **Target segmentation** involves computation of a target affine motion model based on normal flow, and subsequent residual calculation.

[5] This is a term used in personal communication with Kristian Simsarian and we make no claim to be originators.

[6] Note that this also includes its position in depth using disparity information

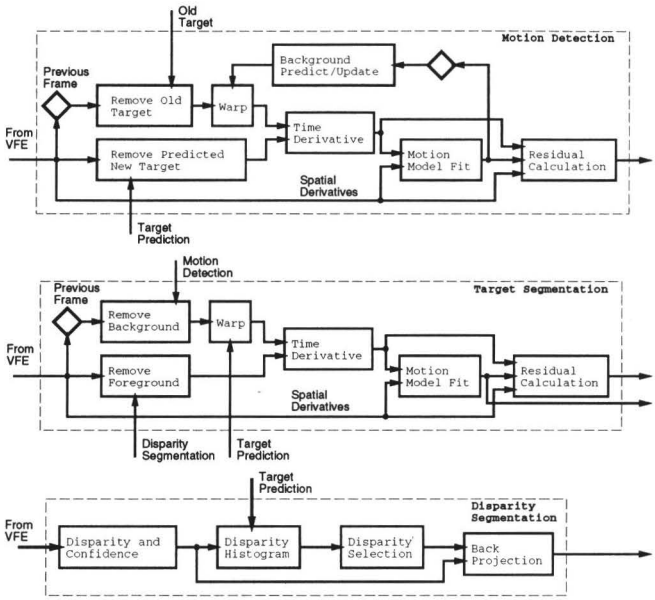

Figure 2: The motion detection, target segmentation and disparity segmentation are shown schematically. These are detailed descriptions of the respective boxes in Figure 1.

- **Disparity segmentation** involves disparity calculation and an equal depth estimation to segment back/target/front.

7 Experiments and Results

In this section we will show how the system performance. During the experiments the images used were recorded during real-time pursuit performed by the already existing pursuit mechanism on the Head-Eye system. Therefore the experiments are performed on quite realistic data since all the noise due to inaccurate control of the head, motion blur, out of focus blur and vergence errors are present. All in all the image sequences are captured in 25 Hz, during camera motion which is purely image driven with no human interference (except that there are humans walking in front of the cameras, being pursued). The experiments were carried out with the head-eye-system constructed by Pahlavan. This head-eye-system has now been mounted on a mobile platform, see Figure 3. The real-time computations performed in the experiments are described in detail in [Uhlin et al., 1995].

7.1 An example with real-time pursuit

The experiment we use to illustrate our principles runs as follows. The system, which is moving about, detects that some target is independently moving in the scene. It then directs its (binocular) gaze towards the object. Observe that what constitutes the object is defined by the motion. The target hence has an identity and it is possible also to locate it in three-space by selecting the corresponding binocular disparities. The importance of this is illustrated when occlusions occur.

To show also how the system performs in the presence of other moving targets, we have performed experiments

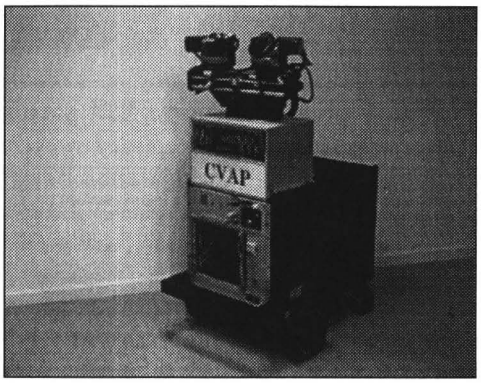

Figure 3: The experimental platform is a Head-Eye-System mounted on a mobile platform.

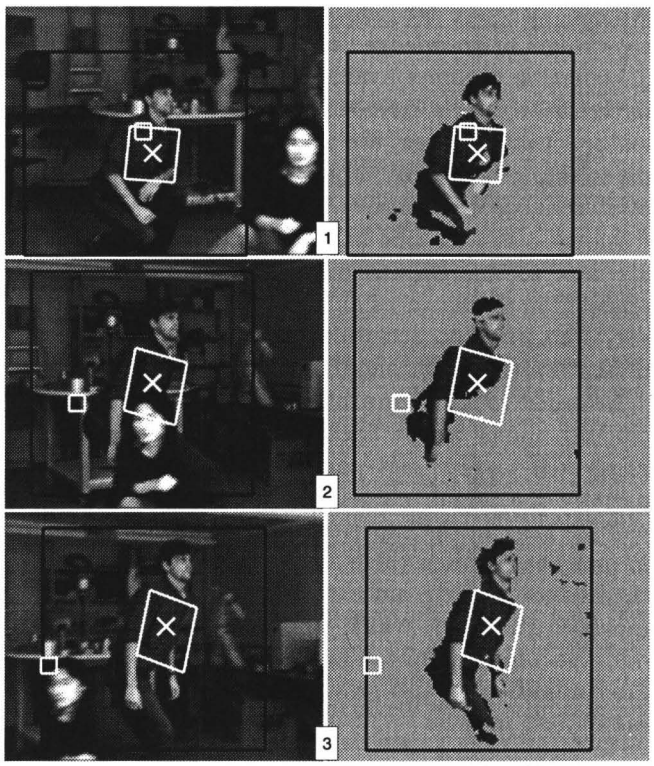

Figure 4: A sequence captured during pursuit. The pursuit is performed at 25 Hz, and shown are every 12th frame.

when another object not only moves in the scene, but also partially occludes the pursued target as it moves across the scene, see Figure 4.

When an occluding object is present, and it also is moving, it will be included by motion detection as a possible target location. If this object in some way dominates, it may well take over the attention of the system if it relies only on motion. To show that, we have removed the disparity detection of areas in front of the target with the result shown in Figure 5. The attention is shifted to the person moving in front, although the attention was initially on the person moving behind. Even though this may be an unwanted behavior of the system, it anyway shows that the system can stably change its attention.

To conclude, we see that without the disparity cue,

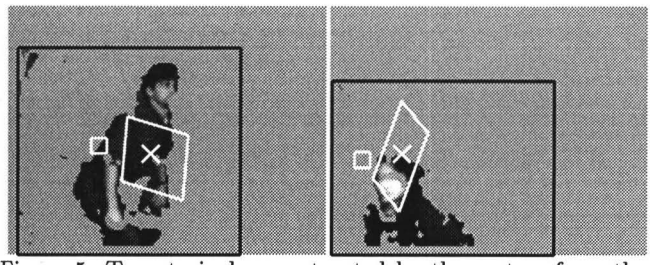

Figure 5: Target pixels as extracted by the system from the sequence shown in Figure 4, but without the disparity cue. The attention shifts to the other moving person. Every 12th frame is shown here.

the system is easily distracted by moving occluding targets, while with the disparity cue, successful pursuit is achieved even in the presence of such distractors.

The feature-maps that are used by the system to determine where and how the target is moving is shown in Figure 6.

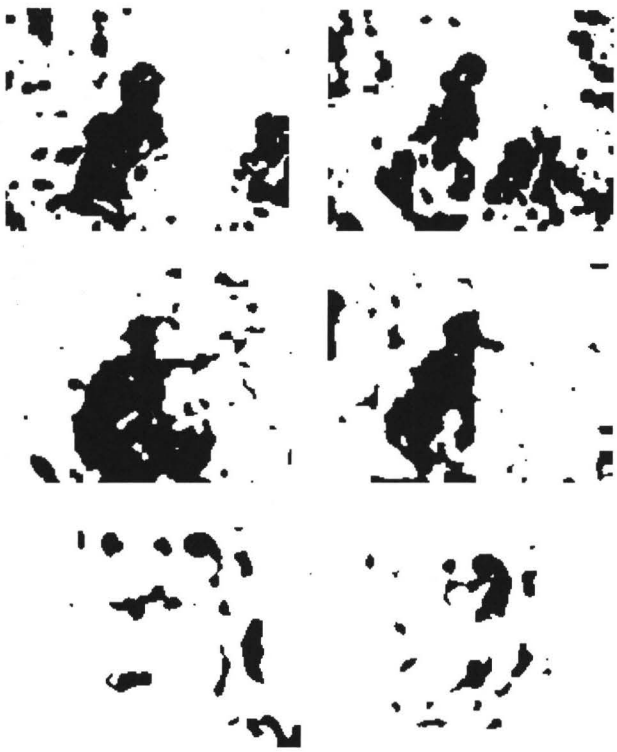

Figure 6: Motion detection returns areas that possibly belong to a moving target. Shown in the top row are the areas which the motion detection marks as possible target pixels. Target segmentation returns areas that are believed to belong to the pursued target. Shown in the middle are the areas which are consitent with the target image velocity model. Disparity segmentation returns areas belong to areas, "behind", "on", and "in front" of the target. Shown in the bottom row are the areas which the disparity segmentation marks as lying in front of the target. These are the masks used to produce the final target masks.

The change in the parameters of the target affine motion model is shown in Figure 4 as a rectangle which is allowed to distort accordingly. The small rectangle shows a fixed point on the background as calculated during background cancellation in motion detection. The white cross shows the result of coarse-to-fine correlation when performed with the target masks produced by the system. The large black rectangle shows a window that is automatically placed around the centroid of the target pixels which are flagged as belonging to the target in each frame. Only pixels inside this rectangle are kept to the next frame. experiments.

8 Conclusion

We have argued that to develop machine based seeing systems one needs to utilize that the world is *rich on information* rather than relying on common minimum information. We have from this viewpoint deduced a number of desirable properties such systems should have and shown that these agree with other current trends in computer vision. One particular consequence is that the systems aspects become essential. We have also experimentally shown some examples of the implications of our suggested approach.

References

Bajcsy, R. (1985). Active perception vs. passive perception. In *Proceedings Third IEEE Workshop on Computer Vision*, pp. 55–59, Bellair. IEEE.

Ballard, D.H. (1989). Animate vision. In *11th IJCAI*.

Ballard, D.H. (1991). Animate vision. *AI, 48*, 57–86.

Bültoff, H. and Edelman, S. (1992). Psycophysical support for a 2-d view interpolation theory of object recognition. In *Proc. Nat. Acad. of Sci.*, volume 89, pp. 60–64.

Crowley, J.L. and Christensen, H.I., editors (1995). *Vision-as-Process*. Esprit Basic Research Series. Springer-Verlag, Berlin.

Culhane, S.M. and Tsotsos, J.K. (1992). An attentional prototype for early vision. In *2nd ECCV*, pp. 551–562.

Gårding, J. and Lindeberg, T. (1995). Direct computation of shape cues based on scale-adapted spatial derivative operators. *IJCV*. (To appear).

Gibson, J. J. (1979). *The echological approach to visual perception*. Houghton and Mifflin, Boston.

Koenderink, J. J. and van Doorn, A. J. (1987). Representation of local geometry in visual system. *Biol. Cyb., 55*, 367–375.

Nakayama, K. and Silverman, G. H. (1988). Serial and parallel processing of visual feature conjunctions. *Nature, 320*, 264–265.

Pahlavan, K., Uhlin, T., and Eklundh, J.-O. (1993). Dynamic fixation. In *4th ICCV*, pp. 412–419.

Shimojo, S., Silverman, G. H., and Nakayama, K. (1988). An occlusion-related mechanism of depth perception based on motion and interocular sequence. *Nature, 222*, 265–268.

Tarr, M. J. and Black, M. J. (1994). A computational and evolutionary perspective on the role of representationin vision. *CVGIP: IU, 60*, 65–118.

Uhlin, T., Nordlund, P., Maki, A., and Eklundh, J.-O. (1995). Towards an active visual observer. In *5th ICCV*, Cambridge, MA. (To appear).