Towards an Active Visual Observer

T. Uhlin, P. Nordlund, A. Maki & J.-O. Eklundh*

Computational Vision and Active Perception Laboratory (CVAP)
KTH (Royal Institute of Technology), Stockholm, Sweden

Abstract

We present a binocular active vision system that can attend to and fixate a moving target. Our system has an open and expandable design and it forms the first steps of a long term effort towards developing an active observer using vision to interact with the environment, in particular capable of figure-ground segmentation. We also present partial real-time implementations of this system and show their performance in real-world situations together with motor control. In pursuit we particularly focus on occlusions of other targets, both stationary and moving, and integrate three cues, ego-motion, target motion and target disparity, to obtain an overall robust behavior.

An active vision system must be open, expandable, and operate with whatever data are available momentarily. It must also be equipped with means and methods to direct and change its attention. This system is therefore equipped with motion detection for changing attention and pursuit for maintaining attention, both of which run concurrently.

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. A key mechanism in this process is fixation. Several researchers have argued convincingly for its importance in active machine vision, see e.g. [1, 6, 8]. Despite reported successful attempts, proposed computational approaches have been limited in scope. Experiments on realistic and complex scenes running in real-time are rare. Techniques for fixation and tracking are often based on a single cue and various restrictions on the motion, and sometimes also on calibration. The focus has often been on single algorithms in isolation and the theories underlying them. Hence there is a host of results on computing various features, optic flow and disparities, and algorithms for vergence and stabilization, but few reports on how useful these algorithms are in a broader context.

We believe that such methods, however elaborate they may be, are insufficient to provide robust behaviors in a complex and dynamically changing environment. Instead of refining and relying on single methods, we advocate a systems oriented approach in which multiple cues are used concurrently. In this way we can utilize the fact that the world is rich on information and rely on the cues that are actually available at a given instant to get a *well-behaved system* [13]. Our overall approach aims at real-time implementations, since we also consider continuous operation over time as a prerequisite for a seeing system.

Vision Systems and Active Vision. Whatever architecture one uses, a system of the type we discuss should be modularly built to be extendible. It cannot be designed to include everything from the beginning, nor be limited to do just what is momentarily implemented. For efficiency reasons the system should be designed to share data (e.g. data from a visual front end, VFE). The overall behavior of the system must not depend heavily on any single part of the system unless this part is correspondingly more robust. A system's performance is no better than the performance of its weakest part. It is important to remember that the system will never be complete. There will always be situations when the performance is not good enough for a given task. This must be taken into account in the design so that, as knowledge increases, more competent and specialized components can be incorporated.

Objectives. In our attempt to develop an active visual observer we have so far implemented what could be called its basic skills. These are functionalities that are natural and desirable for any active vision system, skills that are directly related to attention and fixation. The design here is guided by the idea that the system is meant to function as the bottom layer of a larger system, where new functionalities can be included which can make use of what is there, and the already existing

* The support from The Swedish Research Council for Engineering Science (TFR) is gratefully acknowledged.
Mail address: NADA, KTH, S-100 44 Stockholm, Sweden.
Email: tomas, petern, maki, joe, @bion.kth.se

system can benefit from what is included.

The way the system will function will of course in the end depend on its goals. At this stage we are considering behaviors based solely on bottom-up information, although the algorithms are tightly connected through feed forward and feedback mechanism, embodying the cue integration.

In summary, the system currently only has the innate goal of attending and visually stabilizing moving targets, while it is itself moving both its body (platform) and its head. We regard this as the basic behavior necessary for most other higher level processes. In particular, our work so far clearly demonstrates how the integration of monocular, binocular and motion cues resolves much of the figure-ground problem: the targets found are objects in the scene. This indicates that an integrated use of multiple low-level cues in general can be applied to come to grips with the figure-ground problem, without the often used high-level information.

The target hardware for our work is the KTH Head-Eye system [7], now mounted on a mobile platform. The current implementation comprises a general system for pursuit, described in detail in Section 3.1. The major parts of the current system are,

- selecting a target
- control of the system for saccade and pursuit
- measuring the speed of the target for pursuit
- measuring and selecting a disparity for pursuit

As reported in e.g. [8] we have already realized certain high performance experiments on our Head-Eye system. Our continued work aims at further developing this system by adding new features in an integrated way, keeping the basic components with adequate performance, and applying the overall systems perspective.

Related Work. Notwithstanding our remarks above, many researchers have presented work towards the same goal of achieving robust pursuit. Among one of the first to show real-time (15Hz) performance in an active vision system were [2]. In [6] an active vision system was demonstrated which performs monocular pursuit in real-time (25Hz). A biologically inspired monocular pursuit system which performs pursuit in real time (2-3Hz) was presented in [12]. More recently an approach to visual tracking was described in which one out of several tracking modules is chosen depending on the situation at hand [9]. [10] and related papers discuss visual servoing based on derived motion parameters, which also forms part of our work.

There is of course also a large amount of work on individual techniques used here, for instance the extensive work on optic flow segmentation as well as on disparity computations. Contributions that in particular have inspired this work are among others [3, 15].

2 System Description

Our aim is the design and implementation of a fully autonomous mobile observer. So far fundamental skills in terms of fixation, target pursuit and target discrimination, have been implemented. The basic 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 captured in real-time during the execution of the former processes.

The full 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 cues to changing the state of the system do not exist. In other words, any active vision system must 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, because the system would otherwise be without choices. We have implemented these two components in the form of a pursuit and a motion detection mechanism, thus obtaining what we believe is the most basic behavioral level for an active observer (Figure 1). (Currently, motion provides the only cue to changing attention, but other cues can easily be incorporated.)

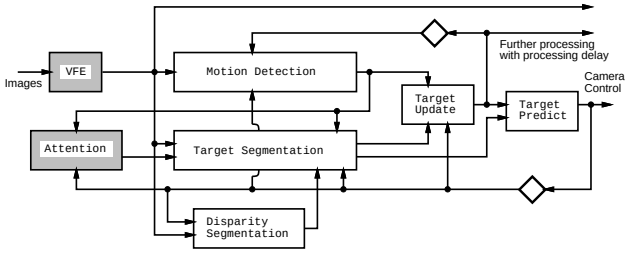

Figure 1: The system implementation is shown schematically. (The diamonds indicate a one frame delay.)

Terminology and Comments. Throughout this section we will use the term *target* and with this we mean the portion in the image which is currently attended and supposed to represent a scene-object. A *target model* is used, which is coded as an image mask, including pixels that are considered as belonging to the target. All segmentation modules produce similar masks after a thresholding step. These masks are used for the integration.

2.1 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 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.* [8], see Section 3.1. This technique works very well when no occlusions are present. We will here present an extension handling occlusions as well.

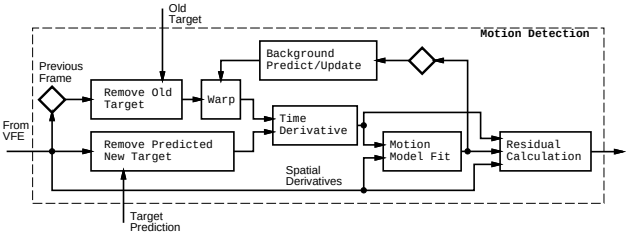

Figure 2: The motion detection is shown schematically. This is a detailed description of the `Motion Detection` box in Figure 1. (The diamonds indicate a one frame delay.)

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. This is provided in the system by a connection from `Motion Detection` to `Target Segmentation`, see Figure 1. 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.

2.1.1 Motion Detection

Motion detection provides the system with a recovering mechanism if the rest of the system for some reason should fail to pursue a target. (In real-world situations this will happen sooner or later.) It functions continuously during pursuit and is used as an integral part in selecting points where the target is lying (Figure 2).

Affine Background Motion Model. Using the brightness constancy constraint a global affine image velocity model is fit to the gradient and time derivative information (see `Motion Model Fit` in Figure 2). Two steps involving feedback are included to account for object motion and large background motion.

- The predicted and previous position and extent of the target are used to mask out parts of the image which likely belong to the object, so that they do not affect the calculation of the affine parameters for the background. See feedback into `Motion Detection` in Figures 1 and 2. This is a crucial step to avoid iterations that otherwise would have been necessary to get reasonable performance even if targets are large. We could say that the iterations are performed in time instead of in a single frame.

- The accumulated affine parameters are used over time to cancel out the majority of the time difference, see feedback into `Warp` in Figure 2. Thereby we can avoid iterations in fitting the affine model, and instead iteratively refine our motion model in time. This is very important to increase the computational speed and decrease the subsequent lag of the pursuit mechanism. We have here an example of how one can tradeoff the performance of a single algorithm for better overall performance.

Background Segmentation. Once the fit is made to the background, and the affine parameters are available, the residual normal image velocity is calculated in the entire image, `Residual Calculation` in Figure 2. The obtained residual is thresholded and high residuals are considered as possible independently moving areas in the scene. The threshold on the residual is adaptive and is set relative to the difference between target and background motion. The result of this can be seen in Figure 8.

2.1.2 Target Segmentation

The aim of target segmentation is to determine which parts of the scene are moving consistently with what is currently believed to be the target. This is the typical situation in connection with occlusions, when the target gradually disappears, to later emerge on the other side of the occlusion. It also supplies the system with the ability to discriminate between several moving targets, found by the motion detection, and distinguish parts of the scene that lie at the same depth, found by the disparity segmentation (Figure 3).

Affine Target Motion Model. The calculations on the target are performed in analogy with what

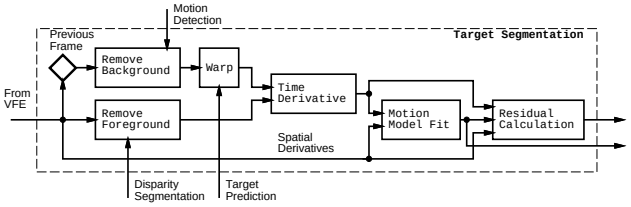

Figure 3: The target segmentation is shown schematically. This is a description of the `Target Segmentation` box in Figure 1. (The diamonds indicate a one frame delay.)

is done for the background motion, with the advantage that most of the computations spent on the background can be used also here (spatial derivatives and smoothed images from the VFE). By sharing these data we greatly increase the performance.

Target Segmentation. Analogous to the motion detection in Section 2.1.1, the residual normal image velocity is calculated and an adaptive thresholding is applied. A sample result of this can be seen in Figure 8 in the presentation of the experiments.

2.1.3 Disparity Segmentation

To increase performance of the system it has also proven useful to have relative depth information, if areas are closer than the target or behind the target (Figure 4). Of course having a disparity measure is necessary to obtain vergence.

Disparity Computation. The disparity computation is carried out with a phase based method. The phase based method has been chosen because of its low complexity, and because it fits well into the VFE framework, which means that we can use the same data for vergence and pursuit movements. It also performs well in a real-time implementation for vergence on the KTH-Head-Eye system (see Section 3.1).

Disparity Selection. The object of disparity selection is to select the disparities that belong to the target in the presence of disparities that arise from other locations in the scene. The in [5] has been further developed to take advantage of the feedback obtained when the system is run in a closed loop together with other cues to follow a single target, to ensure that version and vergence components in smooth pursuit remain consistent. The basis for selection is the predicted location and extent of the target. Also other researchers

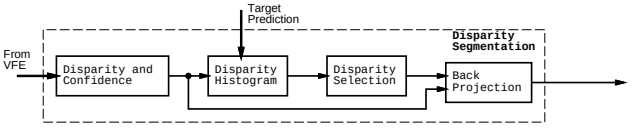

Figure 4: The disparity segmentation is shown schematically. This is a detailed description of the `Disparity Segmentation` box in Figure 1.

have presented work on disparity selection, e.g. [11].

2.2 Integration

The integration is performed mainly in `Target Update` (Figure 1), but in the `Target Segmentation` (Figures 1,3), motion detection information and disparities are also integrated. The target extent is kept in a mask. The integration produces an updated target mask using masks produced in Sections 2.1.1 and 2.1.2 (see Figure 8). There are two parts needed to update the target mask, one to exclude pixels and and one to include pixels.

Target Exclusion. Target areas that do not get support from either motion detection or target segmentation are excluded from the target model. Also the disparity module detects areas in the scene that lie in front of the pursued target, which are then excluded from the target model.

Target Inclusion. Areas in the scene that are detected as both moving independently, detected by motion detection, and are moving consistently with the target image velocity model from the target segmentation, are added to the target model. That is, if pixels arise in both in the top and bottom of Figure 8, they are considered as target pixels.

3 Real-Time Implementation

Presently we have implemented parts of what has been described above to perform real-time visual processing coupled to motor control of the KTH Head-Eye system. These are briefly described here, and visual results are presented in the experimental section.

3.1 The Real-Time Pursuit Mechanism

An implememtation of a pursuit system, running in real-time, includes not only parts from above, motion detection, target velocity and disparity estimation, but also motor control of the KTH-Head-Eye system, thus implementing smooth pursuit, vergence and saccades in a closed loop. The implementation is made on a T800 Transputer network, where both the motor control and visual processing takes place.

Image Velocity Estimation for Smooth Pursuit. The velocity estimation is done in the center of a cyclopean image using a binomial pyramid with four layers, in which the velocity estimation is derived coarse to fine (see [14] for details).

The maximum measured motion is 15 pixels between frames in both the horizontal and vertical direction, and it operates in 25 Hz with about 50 ms delay from the occurrence in the world.

Disparity Estimation. In the most integrated version of the pursuit system, a log-polar mapping has been used to give disparity estimation for vergence control. The disparity estimates are computed at 5 Hz, while the pursuit signal computed from image velocity is measured, and the pursuit controlled, 25 times per second. For sample experiments see Section 4.1.

Maki *et al.* [4] have demonstrated the use of phase based disparity estimation for vergence control on our Head-Eye system. Vergence errors were measured with this phase based disparity estimator at 25 Hz to control the cameras to continuously and smoothly verge correctly on an approaching moving target.

Frame Differencing for Target Selection. For detecting targets to attend to, the real-time system uses frame differencing. This process runs at 10 Hz, and a saccade is triggered with about a 150 ms delay.

3.2 The Real-Time Motion Detection Mechanism

A real-time implementation of a more complex algorithm for detecting moving targets in the presence of camera and observer ego-motion, similar to what is presented in Section 2.1.1, has also been made.

This system runs continuously, capturing image data directly from one camera on the robot head, and in a closed loop controls the camera motion to track the moving object. The loop runs in 25Hz. The image processing takes place on a MaxVideo200. The robot head is controlled by a T800 Transputer network.

Figure 5: The experimental platform is a Head-Eye-System mounted on a mobile platform.

4 Experiments and Results

We will here show how the system performs in a number of different situations, both in the presence of all cues and when some cues are not present. The target will in most examples undergo complex movement, including rotations and movements towards and from the camera, while the observer undergoes ego-motion. No knowledge about the motion is assumed.

Figure 6: A sequence captured during real-time (25 Hz) pursuit. Shown are every 24^{th} frame.

During all the experiments (except the experiment with the rotating umbrella) the images used were recorded during real-time pursuit performed by the already existing pursuit mechanism on the Head-Eye system (see Section 3.1). 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 image sequences were captured at 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). This facility enables us to experiment on a variety of sequences under different conditions exemplifying the performance of the system presented in Section 2 in many different situations. For performance figures see [14].

4.1 Real-Time Pursuit with Motor Control

The performance of the real-time pursuit mechanism is shown in Figures 6, 9 and 11. For an explanation of the overlay we refer to the next section. We can see how the pursuit mechanism centers a target visually while the target is moving across the room. The visual processing and control are performed at 25 Hz and these sequences are used in the subsequent three experiments to show the increased performance achieved with the system described in Section 2, especially in

Figure 7: Resulting target pixels as extracted by the system (after integration and update) from the sequence shown in Figure 6.

conjuntion with occlusions. In Figure 6, when the head moves behind the occluding object in the last frame, it is not pursued across to the other side, but stays on the occluding pole.

4.2 Handling Occlusions

With figure ground segmentation provided by the system descibed in Section 2, the system can handle partial occlusions and segment out the target, see Figure 7. We can see how the occluding pole is excluded from the target. The target was lost by the simplified real-time pursuit process, but is now correctly pursued past the occluding object.

The result of the affine fit to the target is shown in Figure 7 as a rectangle which is allowed to distort accordingly. The absolute location and shape of this distorted rectangle is of no relevance. The initial location is determined when a moving target is detected and its subsequent movement is driven by the affine fit to the target image velocity. The rectangle's relative movement between frames, reflects target movement, lateral, vertical, rotational and expansion, all of which are important descriptors in the pursuit of a target. 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

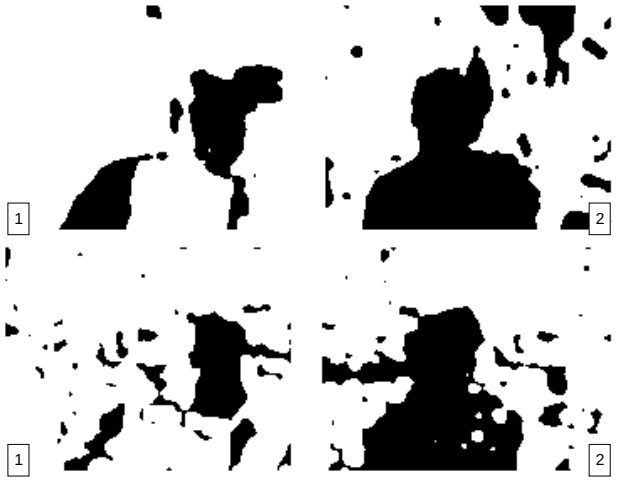

Figure 8: Motion detection (1^{st} row) and Target segmentation (2^{nd} row) return masks (before integration is performed) used to produce the final target masks in Figure 7 (compare with the 1^{st} row).

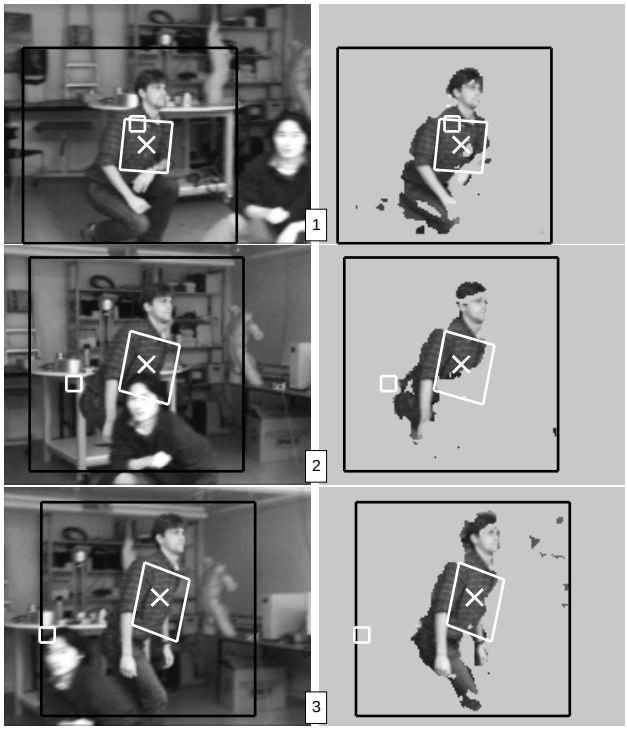

Figure 9: A sequence captured during pursuit. The pursuit is performed at 25 Hz, and shown are every 12^{th} frame. the target masks produced by the system. The large black rectangle shows a window that is automatically placed around the centroid of the target pixels which are flagged as belonging to the target in each frame. Only pixels inside this rectangle are kept to the next frame. To achieve this result the system uses two masks (see Sections 2.1.1 and 2.1.2, and Figure 8).

4.3 Occlusion and Other Moving Objects

To show also how the system performs in the presence of other moving targets, we have performed experiments when another object not only moves in the scene, but also partially occludes the pursued target as it moves across the scene, see Figure 9.

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 illustrate that, we have removed the disparity detection of areas in front of the target with the result shown in Figure 10. 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, the system is easily distracted by moving occluding targets, while with the disparity cue, successful pursuit is achieved even in the presence of distractors.

4.4 Expanding Targets

The ability to detect and pursue an object that moves toward the observer is of great importance since such an object presents a potential collision. It is then important to receive a measure which is directly linked to the approach of the target so that an avoidance maneuver can be deployed in time. Although just qualitatively shown here, the affine model provides such a measure, which is depicted by the distorted white rectangle in Figure 11, together with the result of the target pursuit and segmentation. This measure is however not used explicitly here, but we see that pursuit is functioning also when the target is approaching.

4.5 Rotating Targets

The performance during target rotation is shown in Figure 12, where the segmentation is shown with black overlay. It is seen that the affine model does pick up the rotation of the target and that a correct pursuit is performed.

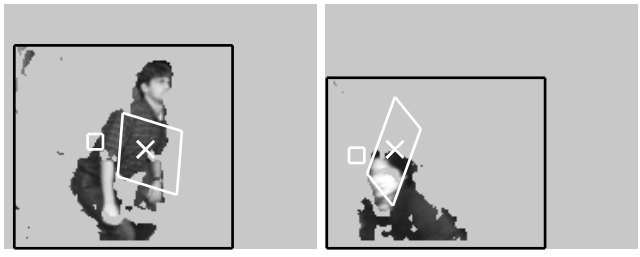

Figure 10: Target pixels as extracted by the system from the sequence shown in Figure 9, but without the disparity cue. The attention shifts to the other moving person. Every 12^{th} frame is shown here.

4.6 Real-Time Motion Detection with Motor Control

Figure 13 shows the result of an experiment with the algorithm described in Section 3.2. The system is tracking the white paper-box moving diagonally through the room from left to right, seen from the camera. While the camera tracks the box, the platform moves approximately straight ahead, the platform slows down, stops and moves straight backwards. Notice that the system is capable of tracking the object both in the presence of a rather smooth untextured background and a quite cluttered background.

5 Discussion

In this paper we have argued that to develop active "seeing" systems we gain notable advantages by considering the problem from a systems perspective. This perspective in particular includes three key components.

- Continuous operation over time and real-time response to different types of events.
- An open and expandable design such that the system can grow while data and results from all levels are shared.
- The integrated use of multiple cues, which is shown to help solving the figure-ground problem in complex, non-static situations.

The emphasis on the systems approach and real-time performance has led us to consider simple algorithms. Hence, we do not claim any particular contributions to any of the partial problems on deriving information

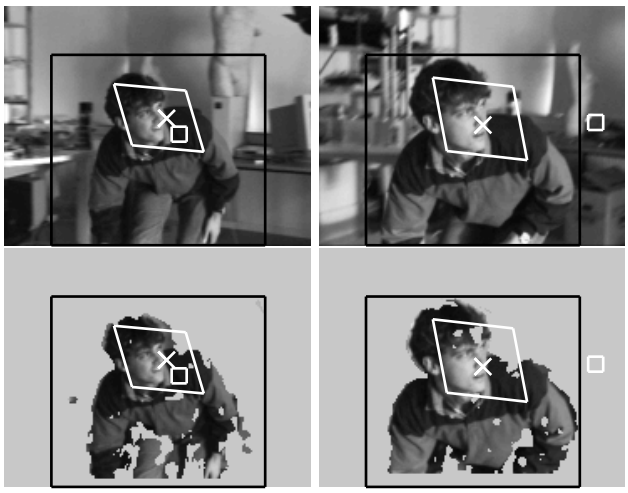

Figure 11: Pursuit during target expansion. On the top is shown the original sequence recorded during real-time pursuit, where every 12^{th} frame is shown. In the bottom row is shown the result of the corresponding segmentation.

from a certain cue. However, we argue that robust performance can be obtained even with simple methods, given an integrated approach of the proposed type. Our experiments support this claim.

References

[1] D.H. Ballard, "Animate vision", *AI*, vol. 48, pp. 57–86, 1991.

[2] D. Coombs and C. Brown, "Real-time smooth pursuit tracking for a moving binocular robot", in *CVPR'92*, pp. 23–28, 1992.

[3] M. Irani and S. Peleg, "Motion analysis for image enhancement: Resolution, occlusion, and transparency", *Journal of Visual Communication and Image Representation*, vol. 4, no. 4, pp. 324–335, 1993.

[4] A. Maki, T. Uhlin, and J.-O. Eklundh, "Phase-based disparity estimation in binocular tracking", in *8th SCIA*, pp. 1145–1152, 1993.

[5] A. Maki, T. Uhlin, and J.-O. Eklundh, "Disparity selection in binocular pursuit", in *4th IAPR Works. on Machine Vision Appl.*, pp. 182–185, Dec. 1994.

[6] D. W. Murray, I. D. Read P. F. McLaughlin, and P. M. Sharkey, "Reactions to peripherial image motion using a head/eye platform", in *4th ICCV*, pp. 403–411, 1993.

[7] K. Pahlavan, *Active Robot Vision and Primary Ocular Processes*. Ph. D. dissertation, Royal Inst. Tech., 1993. ISRN KTH/NA/P--93/16--SE.

[8] K. Pahlavan, T. Uhlin, and J.-O. Eklundh, "Dynamic fixation", in *4th ICCV*, pp. 412–419, 1993.

[9] P. N. Prokopowicz, M. J. Swain, and R. E. Kahn, "Task and environment-sensitive tracking", in *CVPR'94 Works. on Vis. Beh.*, pp. 73–78, 1994.

[10] V. Sundareswaran, P. Bouthemy, and F. Chaumette, "Active camera self-orientation using dynamic image parameters", in *3rd ECCV*, vol. II, pp. 111–116, 1994.

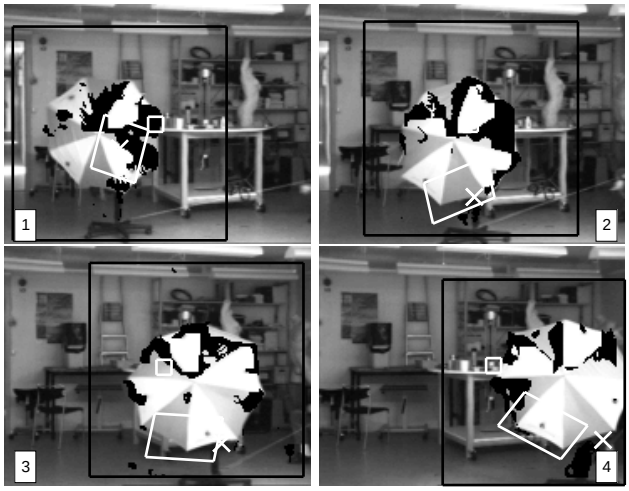

Figure 12: Pursuit of a rotating umbrella. The sequence shows every 18^{th} frame in a 25 Hz sequence. The black areas show which areas are used for the affine fit depicted by the distorted white rectangle.

Figure 13: The figure shows the motion detection mechanism with real-time motor-control feedback, tracking the white box (see Section 4.6). (8 seconds between frames)

[11] J. Taylor, T. Olson, and W. N. Martin, "Accurate vergence control in complex scenes", in *CVPR'94*, pp. 540–545, 1994.

[12] S. Tölg, "Gaze control for an active camera system by modeling human pursuit eye movements", in *Proc. of the SPIE: Intelligent Robots and Computer Vision IX*, vol. 1825, pp. 585–598, 1992.

[13] T. Uhlin and J.-O. Eklundh, "Animate vision in a rich environment", in *14th IJCAI*, Aug. 1995.

[14] T. Uhlin, P. Nordlund, A. Maki, and J.-O. Eklundh, "Towards an active visual observer", Tech. Rep. ISRN KTH/NA/P--95/08--SE, Royal Inst. Tech., 1995.

[15] John Y. A. Wang and Edward H. Adelson, "Spatio-temporal segmentation of video data", in *Proc. of the SPIE: Image and Video Processing II*, vol. 2182, 1994.