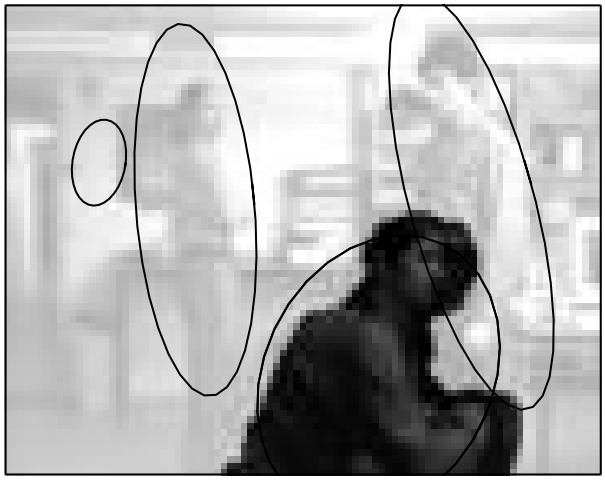
KUNGL
TEKNISKA
HÖGSKOLAN

STOCKHOLMS
UNIVERSITET

Department of Numerical Analysis and Computing Science
TRITA-NA-P98/05 • ISSN 1101-2250 • ISRN KTH/NA/P--98/05--SE • CVAP218

Figure-Ground Segmentation Using Multiple Cues

Peter Nordlund



Dissertation, May 1998
Computational Vision and Active Perception Laboratory (CVAP)

KUNGL
TEKNISKA
HÖGSKOLAN

STOCKHOLMS
UNIVERSITET

Figure-Ground Segmentation Using Multiple Cues

Peter Nordlund

Dissertation, May 1998
Computational Vision and Active Perception Laboratory (CVAP)

Akademisk avhandling för
teknisk doktorsexamen vid
Kungl Tekniska Högskolan

Maj 1998

NADA, KTH, 100 44 Stockholm

ISRN KTH/NA/P--98/05--SE
ISSN 1101-2250: TRITA-NA-P98/05
ISBN 91-7170-261-X

KTH Högskoletryckeriet
Stockholm 1998

Abstract

The theme of this thesis is *figure-ground segmentation*. We address the problem in the context of a visual observer, e.g. a mobile robot, moving around in the world and capable of shifting its gaze to and fixating on objects in its environment. We are only considering bottom-up processes, how the system can detect and segment out objects because they stand out from their immediate background in some feature dimension. Since that implies that the distinguishing cues can not be predicted, but depend on the scene, the system must rely on *multiple cues*. The integrated use of multiple cues forms a major theme of the thesis. In particular, we note that an observer in our real environment has access to 3-D cues. Inspired by psychophysical findings about human vision we try to demonstrate their effectiveness in figure-ground segmentation and grouping also in machine vision.

An important aspect of the thesis is that the problems are addressed from a *systems perspective*: it is the performance of the entire system that is important, not that of component algorithms. Hence, we regard the processes as part of perception-action cycles and investigate approaches that can be implemented for real-time performance.

The thesis begins with a general discussion on the problem of figure-ground segmentation and thereafter the issue of attention is discussed. Experiments showing some implementations of attentional mechanisms with emphasis on real-time performance are presented. We also provide experimental results on closed-loop control of a head-eye system pursuing a moving object. A system integrating motion detection, segmentation based on motion and segmentation based on stereo is also presented. Maintenance of an already achieved figure-ground segmentation is discussed. We demonstrate how an initially obtained figure-ground segmentation can be maintained by switching to another cue when the initial one disappears. The use of multiple cues is exemplified by a method of segmenting a 2-D histogram using a multi-scale approach. This method is further simplified to suit our real-time performance restrictions. Throughout the thesis the importance of having systems with a capacity of operating continuously on images coming directly from cameras is stressed, thus we prove that our systems consist of a complete processing chain, with no links missing, which is essential when designing working systems.

Acknowledgments

First of all I am grateful to my supervisor Jan-Olof Eklundh, who enthusiastically have guided me through the world of computer vision. Without his support and interest this thesis could not have been made.

I am also very grateful to Tomas Uhlin who has provided lots of invaluable help both as an inspiring collaborator, and also with programming, especially for the MaxVideo and Transputers.

An other person formerly at this laboratory that I wish to especially tribute is my co-author Atsuto Maki, who now has returned to Japan.

Throughout my work at the Computational Vision and Active Perception laboratory I have been working with the head-eye system designed and constructed by Kourosh Pahlavan without which lots of this work had not been possible to carry out.

I would also like to thank Daniel Fagerström for long and interesting discussions over the years, Jonas Gårding for being a constant source of good ideas, Magnus Andersson, Demetrios Betsis, Henrik Christensen, Ambjörn Naeve, Stefan Carlsson, Lars Bretzner, Niklas Nordström, Göran Olofsson, Tony Lindeberg, Kjell Brunnström, Mengxiang Li, Antônio Fransisco, Antonis Argyros, Anders Orebäck, Mattias Lindström, Mårten Björkman, Jorge Dias, Fredrik Bergholm, Pär Fornland, Peter Nillius, Kristian Simsarian and David Jacobs for stimulating discussions.

I would like to thank Harald Winroth and Matti Rendahl for their support in all matters concerning programming and computers. Carsten Bräutigam is another member of the group that I would like to thank for his tremendous patience regarding my LaTeX questions.

I would like to thank Elenore Janson for proof-reading, and for her whole-hearted support during the time of writing this thesis.

In one of my first projects I had the exciting opportunity to collaborate with Jean-Paul Bernoville, Henri Lamarre and Yann Le Guilloux, Michel Dhome, Jean-Tierry Lapreste and Jean-Marc Lavest.

This work has in part been supported by TFR, The Swedish Research Council for Engineering Science. It has during its latter stages also received support from CAS, The Centre for Autonomous Systems. We gratefully acknowledge this support.

Finally I would like to mention all the other members and former members of the group, Birgit Ekberg-Eriksson, Ann Bengtsson, Lars Olsson, Anders Lundquist, Kiyoyuki Chinzei, Kenneth Johansson, Kazutoshi Okamoto, Anna Thorsson, Pascal Grostabussiat, Akihiro Horii, Wei Zhang, Cornelia Fermüller, Yannis Aloimonos, Carla Capurro, Mattias Bratt, Björn Sjödin, Martin Eriksson, Danica Kragic, Lars Petersson, Danny Roobaert, Mikael Rosbacke, Hedvig Sidenbladh, Daniel Svedberg, Dennis Tell, Maria Ögren, Ron Arkin, Patric Jensfelt, Oliver Mertschat, Olle Wijk, Andres Almansa, Svante Barck-Holst, Josef Bigün, Jørgen Bjørnstrup, Henrik Juul-Hansen, Rosario Cretella and Uwe Schneider. Thank you all.

Contents

List of papers

This thesis also exists in a long version with the papers below included. This version of the thesis only contains part one, the summary.

Paper A
Nordlund, P. and Uhlin, T. (1996). Closing the loop: Detection and pursuit of a moving object by a moving observer, *Image and Vision Computing* **14**(4): 265–275.

Paper B
Uhlin, T., Nordlund, P., Maki, A. and Eklundh, J.-O. (1995a). Towards an active visual observer, *Proc. 5th International Conference on Computer Vision*, Cambridge, MA, pp. 679–686.

Paper C
Uhlin, T., Nordlund, P., Maki, A. and Eklundh, J.-O. (1995b). Towards an active visual observer, *Technical Report ISRN KTH/NA/P--95/08--SE*, Dept. of Numerical Analysis and Computing Science, KTH, Stockholm, Sweden. Shortened version in Proc. 5th International Conference on Computer Vision pp 679–686.

Paper D
Maki, A., Nordlund, P. and Eklundh, J.-O. (1998). Attentional Scene Segmentation: Integrating Depth and Motion from Phase, Extended version of tech report ISRN KTH/NA/P--96/05--SE submitted to Computer Vision and Image Understanding.

Paper E
Nordlund, P. and Eklundh, J.-O. (1997b). Towards a seeing agent, *First Int. Workshop on Cooperative Distributed Vision*, Kyoto, Japan, pp. 93–123. Also in tech report ISRN KTH/NA/P--97/05--SE.

Paper F
Nordlund, P. and Eklundh, J.-O. (1998). Real-time figure-ground segmentation, *Technical Report ISRN KTH/NA/P--98/04--SE*, Dept. of Numerical Analysis and Computing Science, KTH, Stockholm, Sweden. Will be submitted to Int. Conf. on Vision Systems, Jan 99.

When any of the papers above are cited, the citation is in **boldface**. Some of the papers also exist as technical reports or conference contribution, here are these references:

Paper A
Nordlund, P. and Uhlin, T. (1995). Closing the loop: Pursuing a moving object by a moving observer, *Technical Report ISRN KTH/NA/P--95/06--SE*, Dept. of Numerical Analysis and Computing Science, KTH, Stockholm, Sweden. Shortened version in Proc. 6th International Conf. on Computer Analysis of Images and Patterns. Also in Image and Vision Computing vol. 14, no 4, May 1996, pp 265–275.

Nordlund, P. and Uhlin, T. (1995). Closing the loop: Pursuing a moving object by a moving observer, *in* V. Klaváč and R. Šára (eds), *Proc. 6th International Conf. on Computer Analysis of Images and Patterns*, Prague, Czech Republic, pp. 400–407.

Paper D

Maki, A., Nordlund, P. and Eklundh, J.-O. (1996). A computational model of depth-based attention, *Technical Report ISRN KTH/NA/P--96/05--SE*, Dept. of Numerical Analysis and Computing Science, KTH, Stockholm, Sweden. Shortened version in Proc. 13th International Conference on Pattern Recognition.

Maki, A., Nordlund, P. and Eklundh, J.-O. (1996). A computational model of depth-based attention, *Proc. 13th International Conference on Pattern Recognition*, Vol. IV, IEEE Computer Society Press, Vienna, Austria, pp. 734–739.

Paper E

Nordlund, P. and Eklundh, J.-O. (1997). Maintenance of figure-ground segmentation by cue-selection, *Technical Report ISRN KTH/NA/P--97/05--SE*, Dept. of Numerical Analysis and Computing Science, KTH, Stockholm, Sweden. Also in First Int. Workshop on Cooperative Distributed Vision 1997, Kyoto, Japan, pp 93–123.

Chapter 1

Introduction

Humans looking around in the world can seemingly without effort segment out and distinguish different objects in the world. Despite this, the corresponding capability has largely eluded the efforts of researchers in computer vision. Figure-ground segmentation, or as it is generally considered in computer vision, image segmentation remains a difficult problem even when a more precise definition is given. Of course, it is worth noting that most work in computer vision concerns pictorial vision, i.e. analysis of (usually pre-recorded) images, and not any kind of natural vision, in which the observer directly samples the three dimensional world. One can argue that figure-ground segmentation is different in the two cases. Indeed, the work presented in this thesis supports this view. Nevertheless, one can often assume that very much the same information is available in the two cases and that the discrepancy in performance between humans and machines therefore may appear puzzling.

In this work we are not considering the segmentation or grouping problems in their full scope. Rather we are investigating the applicability in computer vision of psychophysical results about the human visual system indicating that three dimensional cues play an important role in such processes (see e.g. Nakayama and Silverman, 1986). More generally, there are findings indicating that the percepts, and not only the retinal information as such, are crucial. A recent collection of results in this direction is given in (Rock, 1997), and an example described in Chapter 2.2.

A second and central theme of our work deals with the use of multiple cues. A human observer is normally gifted with a visual apparatus that can process binocular and monocular information as well as motion, color and form (see e.g. Zeki, 1993, for a deliberation on this multifunctional structure). Hence, it is reasonable to conclude that a machine vision system capable of functioning in a real environment analogously should be able to capitalize on whatever information the world offers and also to integrate this information. If we expect to achieve even a fraction of human performance in an unknown world we should consider systems that have similar capabilities.

The third major theme of the thesis concerns the systems perspective. Much of the traditional work in computer vision uses a reductionist approach and in isolation addresses specific visual problems, such as reconstruction of the scene from stereo-pairs of images, segmentation of an image into sets of regions optimizing some criterion, or, recognition of faces or objects seen against a simple background. Such approaches provide us with valuable insights into the limits of machine perception and the feasibility of algorithms. However, often this work only superficially touches upon questions about to what processes or system (possibly a human interpreter) this information is going, or even from where it emanates. Therefore, it does not directly tell us how to design systems that can *see* in the sense of a human observer or even a seeing robot. In this work, which forms part of the longstanding work on active vision within the CVAP group (see e.g. IJCV, Vol 17:2, Eklundh, 1996), we are interested in processes that work in an integrated system of the latter type, and ultimately in a system that functions in a closed loop with the environment. Truly, not all the work we report here has this emphasis, but the perspective is such. In particular, the work forms an attempt to support the general claim that from a systems perspective many of

the difficult problems arising in machine vision can be solved, and that the algorithms required sometimes are both simple and coarse, and exactly because of that also robust. The ESPRIT-projects VAP I and II formed a step towards demonstrating this principle (see e.g. Crowley and Christensen, 1995, especially the introduction). The work described here continues in the same spirit.

1.1 Outline of our approach

The main theme in this thesis is *figure-ground segmentation*. We approach this problem by using *multiple cues*, especially 3-D cues, in order to utilize the fact that the *world is rich* on information. We also apply a *systems approach* to these problems, in the spirit of earlier work by Pahlavan (1993) and Uhlin (1996) in our laboratory, who had a focus on systems operational in real-time in interaction with the environment. Our goals have been:

1. To demonstrate that robust figure-ground segmentation can be achieved through the use of 3-D cues.
2. To demonstrate that integration of multiple cues is essential in this context.
3. To demonstrate that the components can be integrated into an operational system if they are formulated, designed and implemented in the context of a fully operational system.

With the available resources at hand today we are able to aim for the following:

1. Continuous processing of incoming images at a frame-rate of about 5-25 Hz.
2. Operation in an indoor environment.
3. Feedback for controlling the camera head.
4. A moving observer, i.e. a platform with a camera head.
5. A binocular camera setup to perform stereo processing to get 3-D information.

In the included papers we have performed experiments to segment out deformable objects in an indoor scene. To create motion in depth we have mainly been limited to consider people moving around. In addition, we have made experiments with rigid objects, both almost untextured and with repetitive patterns on the surface. Our aim is to provide suitable input for higher level processes, e.g. recognition. To demonstrate that the segmented objects are represented with image masks. Example of such masks can be seen e.g. in Figures 1.5(b) and 1.8

The clustering needed to segment the objects is performed in feature-space and tracking in both feature-space and images at the same time.

In the next section a few examples of results are briefly presented, just to show what kind of images we have performed our experiments on.

1.1.1 Experimental examples

In this section follows a few examples from the papers included in the thesis to let the reader get a feeling of what kind of images we have been working with.

In **(Nordlund and Uhlin, 1996)** we perform tracking of a moving object in a feedback loop controlling a camera-head (Figure 1.2) with a frame-rate of 25 frames/s. An example of tracking results can be seen in Figure 1.3. The tracking algorithm works as follows: By exploiting the brightness constancy constraint (see e.g. Horn and Schunck, 1981) and a translational or affine velocity model[1] we can compensate for the background motion. Using two consequtive frames we can create a difference image. This difference image is thresholded. In the resulting image we compute a centroid which we track. See Figure 1.1 for example of difference images and centroid.

[1] In the real-time implementation we use the translational motion model.

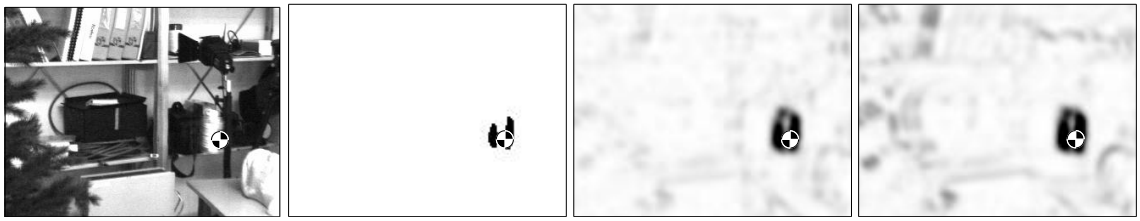


Figure 1.1: *Motion detection of a swinging can* **(Nordlund and Uhlin, 1996)**. *A motion detection algorithm computes a centroid on every frame.* 1^{st} *column shows original image.* 2^{nd} *column shows the thresholded error image which is used to compute the centroid, (affine velocity model).* 3^{rd} *column shows the error image before thresholding, (affine velocity model). In columns 1-3 the centroid is marked as computed from the* 2^{nd} *column. The* 4^{th} *column shows the error image before thresholding, (translational velocity model). In column 4 the centroid is marked as computed from the image in column 4 but thresholded, (not shown here). Frame numbers as marked.*

Figure 1.2: *The head-eye platform used for the experiments in* **(Nordlund and Uhlin, 1996)**.

Figure 1.3: *Real-time tracking of a moving paperbox* **(Nordlund and Uhlin, 1996)***, at a frame-rate of 25 frames/s with head motion motor control in closed loop (every* 50^{th} *frame as seen by the camera are shown).*

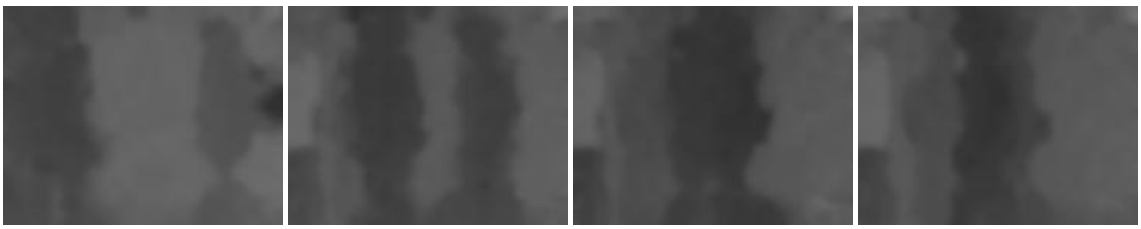
In **(Maki et al., 1998)** we have used a phase-based stereo algorithm to extract binocular disparities. We have also used the same algorithm to extract horizontal motion disparities. The images shown in Figures 1.4 and 1.5 come from the article. A brief overview of the implementation follows: The binocular disparities (Figure 1.4(b)) are histogrammed. The histogram is analyzed, peaks in the histogram are found and backprojected, thus creating segmentation masks for different depths. In the same way the horizontal flow (Figure 1.4(d)) is segmented. A centroid coming from the motion detection module tells where in the image to attend. Not yet thresholded images from the motion detection can be seen in Figure 1.5(a). Next, one of the binocular masks is chosen, namely the one that covers the coordinate to attend to. The same attention-procedure goes for the motion mask. As the last step the two chosen masks are fused by a logical AND, thus creating resulting target masks as in Figure 1.5(b).

In **(Nordlund and Eklundh, 1997)** the approach from **(Maki et al., 1998)** is further developed. Instead of making the segmentation of binocular disparities[2] and motion disparities independent of each other we do it concurrently. We do this by making a 2-D histogram with depth in one dimension and motion in the other. To analyze a 2-D histogram is a difficult problem, which can be solved in many ways. We have here chosen an approach outlined in (Lindeberg, 1993). In later work a faster algorithm was developed **(Nordlund and Eklundh, 1998)**.

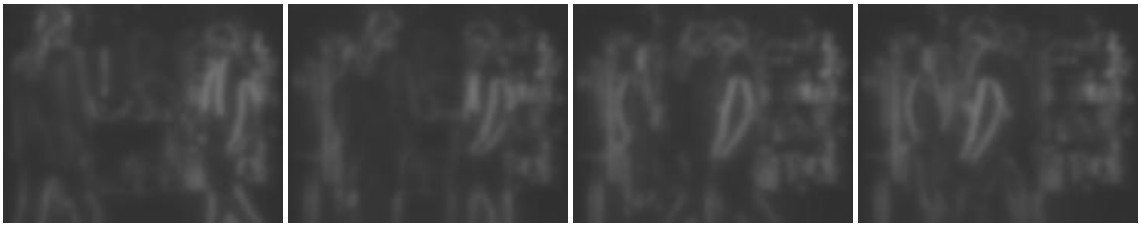
2-D histograms at different scales can be seen in Figure 1.6. As can be seen in the figure the number of peaks is depending on how much the histogram is blurred. An example of detected peak regions (blobs) can bee seen in Figure 1.7. Detected peaks are backprojected into the original image thus creating masks. In Figure 1.8 the backprojected blobs from Figure 1.7 are shown.

[2]Sometimes we use the term depth when talking about binocular disparities; in that case we mean relative depth, which is proportional to the disparity.

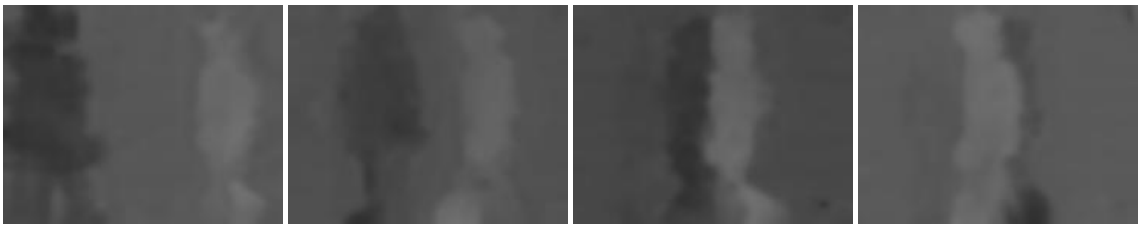
(a) Original images from the left camera



(b) Disparity maps. The darker, the closer.



(c) Certainty maps computed corresponding to the disparity map. The lighter, the higher certainty.



(d) Horizontal flow maps. The lighter, the more leftward motion.

Figure 1.4: *An example sequence with 3 moving persons taken by a stationary binocular camera head. Every* 10^{th} *frame of the left image is shown (40 msec between frames).*

(a) Detected motion. Darker grayscale represents more motion.

(b) Computed target masks.

Figure 1.5: *Continuation of the example beginning in Figure 1.4.*

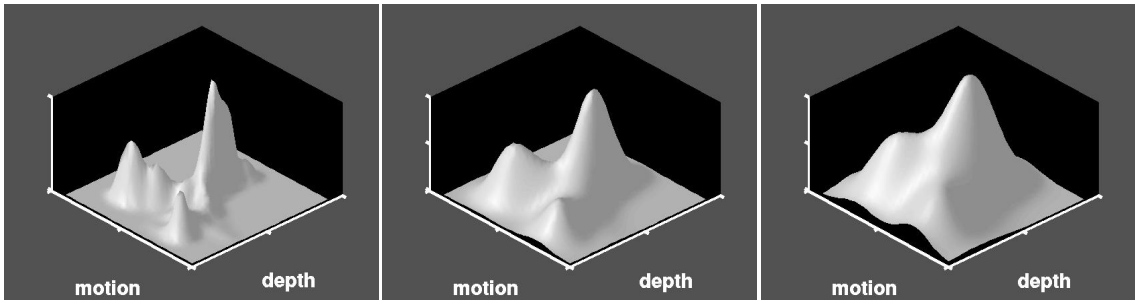


Figure 1.6: *Example of a 2-D histogram with different amounts of blurring. Coarser scale rightwards.*

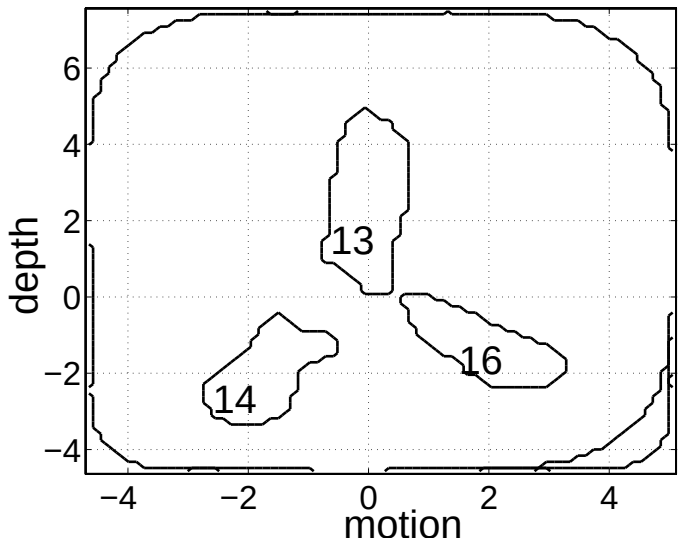


Figure 1.7: *Example of blobs extracted from the 2-D histogram shown in Figure 1.6. Blob number 13 represents the background. Blob number 14 represents the leftmost detected person. Blob number 16 represents the rightmost detected person.*

Figure 1.8: *Example of backprojected blobs (marked with darker colour). Number in upper right corner refers to blob-number in Figure 1.7.*

Chapter 2

Figure-Ground Segmentation

In a system of the type we have in mind such as a seeing robot there are both top-down and bottom-up processes. The first ones are task-driven and include various predictions of what is of interest or to expect for the perceptual system. The last ones are data-driven and serve to attract the system's interest to structures and events in scene that may be relevant to process further. These processes are of course also task oriented in the sense that they serve a set of background tasks for e.g. detection of obstacles, object of interest and other robots. In our work we focus on these types of processes.

2.1 Segmentation and grouping

Image segmentation forms the first and most difficult task in many approaches to image interpretation. Whenever one wants to compute properties of objects, such as their shape, location in space or in the image, or motion, one needs to segment out the objects in the image, or the features of the objects in the feature-space. A particularly important case is object recognition: much work in this area assumes that the objects are viewed against a simple background, or that the characteristic features can be found by some grouping or segmentation processes.

There may be other reasons why a system needs to segment out or keep track of objects, e.g. in visually guided grasping or obstacle avoidance in robotics, or, in tasks such as surveillance and gesture and motion interpretation. Although it might be possible to perform the required computations on the entire images, it is often assumed that the objects of interest can be isolated beforehand. A good reason for this is, of course, that the statistics of the features of interest are simpler over the objects alone than over the full image.

In the thesis we are addressing this problem in the context of an observer consisting of a camera head looking at a 3-D scene[1]. Among other things this puts emphasis on algorithms that potentially can be implemented with real-time performance. Before we elaborate on our approach we will first briefly survey earlier approaches to the general problem and discuss how they apply to this case.

2.1.1 Segmentation of static monocular images

Considerable work has been devoted to the general problem of image segmentation. Classical methods such as (Yakimovsky, 1974; Schachter et al., 1979) and Hanson & Riseman (1978) are usually based on textures or edges. More modern methods often perform some optimization (Mumford and Shah, 1985; Blake and Zisserman, 1987) or apply (anisotropic) diffusion methods (see e.g. Perona and Malik, 1990; Nordström, 1990a; Nordström, 1990b). Examples of methods using multispectral (usually color) images can be found in (Ohlander, 1976; Eklundh et al., 1980)

[1] We are generally aiming for a mobile observer, but this is not the case in all our experiments.

Common of the abovementioned approaches are that local or global criteria and the image information determine the segmentation. There is no direct influence from 3-D cues or relation to the task or scene contents. These methods are computationally demanding and presently hardly useful in real-time systems with high frame-rate.

A class of methods that is more relevant to our work deals with the use of deformable contours. Following the introduction of snakes (Kass et al., 1987) such methods have been used to find subjective contours, thereby giving a segmentation of objects of interest. Such methods are particularly powerful in the dynamic case, which we will return to below. They also allow real-time implementations. The main problem with such contour based methods is the initialization of the model. Often the initialization problem is not addressed. Sometimes the initialization comes from some traditional segmentation method in which edges are detected, linked and grouped, for instance using the Minimum Description Length algorithm (see e.g. Leclerc, 1989). So far such methods are rather slow and require offline processing.

This brief survey is of course not in any sense exhaustive. It is just intended to highlight some of the main lines of research on the topic and discuss their relevance to our case.

2.1.2 Grouping

Grouping methods are intended to group data into meaningful structures on the basis of properties such as similarities, proximity, and good continuation. The inspiration to such approaches in machine vision derive from the Gestalt theory (Koffka, 1935). As stressed by Witkin and Tenenbaum (1986) finding structure beyond what local feature detectors return is a crucial step in deriving information about scenes from images. In fact, this has been accepted wisdom in machine vision for a long time and is a theme also in (Marr, 1976).

Most of the extensive work on grouping has considered the structural organization of local features. The use of "non accidental" properties, e.g. co-terminations of extracted edges has been studied by (e.g. Biederman, 1985; Lowe, 1985; Fischler and Bolles, 1986). In a sense, the commonly used edge-linking methods belong to the same category, but due to their serial character they are difficult to include in the context of our work.

Grossberg (1993) suggests that occluded shapes may be filled in and may be perceived as coherent. Barrow and Tenenbaum (1993) speculate that there is feedback and cooperation/integration between output from vision modules in biological systems.

Another type of methods try to find coherent structures based on global criteria. An example is given by (Sha'ashua and Ullman, 1988; Ullman, 1996) who optimize a salience measure to find objects in a cluttered background. The segmentation is made in a hierarchical manner. A first parallel step extracts highlights in the image. A second serial step can then be applied to more thoroughly investigate the highlights.

A method using an affine transformation of image patches is presented by (Leung and Malik, 1996). Their method groups image elements and can be seen as a texture grouping.

Grouping techniques form a natural basis for any approach to figure-ground segmentation. Hence, much insight can be gained by considering the properties used in previous work on finding image structure using grouping. However, most of the attempts in the literature work sequentially over the images and rarely lead to potential real-time implementations. Moreover, the work we have touched upon in this section focuses on *image* structure and does not include 3-D organization explicitly. Therefore, we have only in a limited sense been able to directly include such methods.

2.1.3 Clustering

The objective of cluster analysis is to separate a set of features or objects into groups so that the members of any group differ from one another as little as possible, while the difference between groups is large according to some criterion. The objects simply consist of a set of values. Clustering methods are central in pattern recognition and also important tools in our work. The goal is to identify and possibly interpret an existing structure of such a set of objects and thus achieve a data

reduction. In the work presented in the thesis we are interested in finding clusters in multidimensional feature-spaces. An important tool for doing that is given by Bayesian decision theory, to which a nice introduction can be found in (Duda and Hart, 1973). A drawback with the Bayesian approach is that it requires a priori knowledge about probabilities, which is seldom available in a practical computer vision application.

General references to clustering can be found in e.g. (Spät, 1980; Hartigan, 1975; Jain and Dubes, 1988). The way we have dealt with the clustering problem in our multidimensional and real-time situation will be described in detail in Chapter 5.

2.2 The role of 3-D cues in figure-ground segmentation

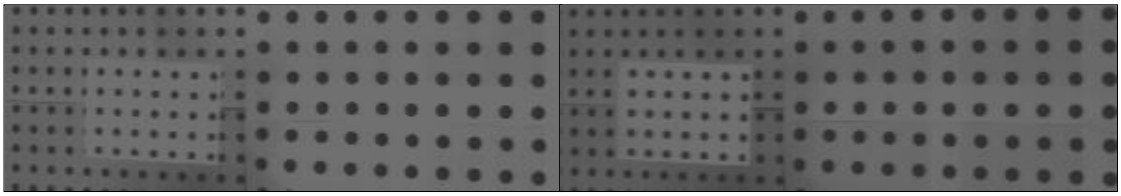
As we have mentioned several times, most of the image segmentation and grouping research has considered segmentation on the basis of features as they appear in the images. For instance, proximity is taken to be retinotopic proximity, similarity in some property analogously. In the introduction we pointed out that research on human vision suggests that we group things differently when cues suggest that a three dimensional configuration is being viewed.

An example of this was given by Rock and Brosgole (1964). They had subjects look at a set of dots arranged in columns and rows. If the stimulus was viewed frontally and the dots in each column were more closely spaced than the spacing between rows, than they were grouped into columns. Using an experimental setup in which the dots were light bulbs hanging freely in a dark room, the situation could be monitored so that the subject perceived the stimulus either in 2-D or in 3-D. In this way Rock and Brosgole (1964) "showed that tilting the array did not change the perceived organization as long as observers had sufficient information to perceive depth accurately". Their conclusion from this and other experiments was that experienced 3-D proximity influenced the grouping. Other work along the same line is also reviewed in (Rock, 1997).

In the coming chapter on how we include attention in our system we will point to another type of influence of 3-D cues observed in humans. With these observations in mind and with the general observation that humans seldom have problems with figure-ground segmentation in the natural world we have based our figure-ground approach for the "seeing" robot on the inclusion of 3-D cues. Two types of cues have been used so far: binocular disparities and motion. We will next briefly discuss these processes, in view of other techniques proposed in the literature.

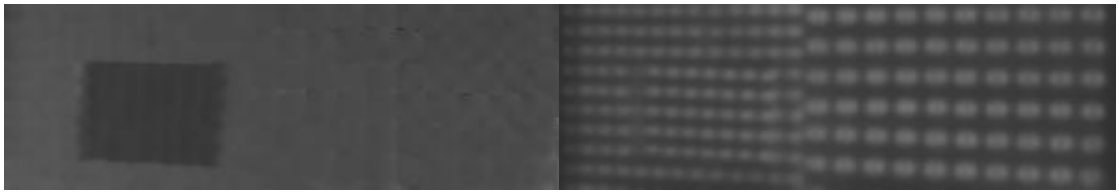
2.2.1 Binocular disparities

The problems of computing binocular disparities and reconstructing the scene from stereo images have been studied in computer vision for about 25 years and long before that in photogrammetry. The emphasis of this work has been on stereoscopic reconstruction. If it is possible to obtain a full 3-D reconstruction, then an important step towards figure-ground segmentation would indeed be taken. However, there are several reasons why such approaches are difficult to apply in our context. First, stereo reconstruction requires calibration or a way of deriving the calibration information. Solutions for the latter problem exist, but results indicate that although they allow computation of the appearance of the image, they may give rise to large errors in attempts of reconstruction. Secondly, most of these methods are far from useful in real-time applications, unless specialized hardware is used. Exceptions exist, such as the work by (Little, 1997) and his coworkers. Thirdly, a full reconstruction, like image segmentation, only provides an implicit figure-ground segmentation. More processing is needed to actually segment out interesting objects. This may well be possible, also in real-time, but we have not pursued that line of work further. More interesting from our point of view are approaches that allow us to segment areas in depth or to find depth discontinuities. Fermüller and Aloimonos (1995) has pointed out the importance of ordinal depth, but shown few results on how to do it on real scenes. Malik (1996) uses a stereo method to find occlusion junctions. Depth discontinuities can also be obtained with other stereo methods, e.g. the phase-based approach used by Maki (1996) which has the benefit of a low



(a) Left image (b) Right image

Figure 2.1: *Original images of a stereo image pair of a slanted background and a sheet of paper in front of the surface.*



(a) Disparity map. Brighter grayscale indicates higher confidence.

(b) Confidence map. Brighter grayscale indicates furhter distance to the cameras.

Figure 2.2: *Result from phase-based algorithm by Maki (1996).*

computational cost. See Figures 1.4(b), 1.4(c), 2.1 and 2.2 for examples of results from the method by Maki.

2.2.2 Motion segmentation and tracking

Tracking moving targets is an important issue for a system that is supposed to maintain an interpretation of a moving object over time. Early work on a real-time video tracking system with the focus on missiles can be found in (Gilbert et al., 1978; Gilbert et al., 1980) and simple correlation trackers were in use already in the 1960's. Other interesting tracking approaches are based on deformable templates, e.g. in (Zhong et al., 1998) a moving hand is tracked in front of a weather forecast map for 15 frames using color region constancy. Common for most methods based on deformable templates is that they require a manual or very simple initialization, e.g. Shen and Hogg (1995) track a moving vehicle but their initialization relies on a static pre-recorded background image.

Toyama and Hager (1996) in their interesting work present a layered architecture for robust tracking. The topmost layer performs the highest precision tracking. The layers below have less precision. After a tracking cycle, control can move up or down the framework depending of the success of tracking during the previous cycle. The authors stress the importance of mechanisms for algorithm failure detection. This is necessary for the transitions between the layers. The selection mechanisms in this approach have roots in areas such as radar target tracking and have promise of great applicability in more general cases in computer vision. Another example of a tracker with different states is presented by Isard and Blake (1998). Their method is an extension of (Isard and Blake, 1996), where a stochastic framework is used to track the outline of moving objects in cluttered scenes. They study tracking of contours as in the case of deformable templates. In addition, their motion tracker is able to switch between three different states, i.e. motion models. In one experiment they report a frame-rate of 0.33 Hz, image size not reported, on a Silicon Graphics O2 R5000 180 MHz workstation. They conclude that the tracker needs a speedup by a factor of 100 times to be useful in applications. In Baumberg and Hogg (1994) moving people are segmented out using a deformable model. Their approach relies on a static camera and a pre-recorded background image, and has real-time performance.

Darrel et al. (1998) present a tracking system using stereo and color to extract the outline of people appearing in front of the camera in real-time. The segmentation is achieved in a serial

manner, first the stereo is used for an initial coarse segmentation, this to not confuse the color algorithm, which performs a refined segmentation with a cluttered background.

(Davis and Bobick, 1997) use a viewbased approach to characterize actions by the observed motion from temporal sequences of images. As an intermediate step in the experiments described, they segment out moving people in real-time, but their approach relies on a static background.

Segmentation of image sequences can also be obtained using flow methods (Irani and Peleg, 1993; Wang and Adelson, 1994; Wang et al., 1994; Uhlin et al., 1995a; Black and Jepson, 1996). In (Irani and Peleg, 1993) an affine model of the motion is used. This is an approach with a potential of real-time performance. However, for most flow methods the computational cost associated with the segmentation of the computed flow is very high. (Shi and Malik, 1998) show very good results on segmenting images using a flow technique, but they do not present any performance figures.

We here also mention the work by Camus (1997) who present a flow-based method with a typical performance of 9 frames/s for an image of the size 64×64. This algorithm runs on a 80 MHz HyperSparc computer. Although no segmentation is presented, the performance is such that it seems possible to further develop the approach to also consider segmentation. Camus also make interesting points on the question of time-space trade-offs.

2.2.3 Tracking and "seeing"

Object tracking is obviously a thoroughly investigated topic, not only in computer vision but also in fields such as radar signal processing. This work is indeed systems oriented in the sense that the perception-action cycle is closed. In many cases the perceptual goal of tracking has been to determine where the target is, either assuming that its identity is known or not dealing with that aspect at all. In other cases, including ours, tracking implies that a moving object is extracted from the scene for a period of time. One can say that the system "sees" the object, which therefore is explicitly represented in some manner and that its representation can be ascribed properties computed from the imagery. In our case the representation is simply a mask and the properties we have in mind concern the color and texture, shape and motion of the corresponding object. Recently there has been an increasing interest in what is called perception of action, see e.g. (Bobick, 1997). This work in the same spirit tries to reach beyond mere target tracking, however, with a different goal in mind. In both cases tracking is not the final goal. Instead motion is used as a key to understanding the moving objects, on our part using also information of other types.

2.2.4 Using multiple cues

As a final remark in this overview of figure-ground segmentation we would like to briefly stress that for a system to be robust in a natural environment it must rely on multiple cues. What makes something possible to segment is that it differs significantly in some property from its immediate surroundings. In a rich environment we can not know a priori what property that could be. Hence the system should in parallel process many different cues to be able to capitalize on the relevant ones. In Chapter 4 we will dwell on this problem. Let us now point out that even if 3-D cues are the key to detecting an object, properties such as color, texture, and shape may be essential to recognize them or relocalize them when they do not move. This forms a motivation for the way we have approached the problem.

Chapter 3

The Role of Attention

There are several reasons for narrowing the channels of input information by focusing on specific areas in the image. The most obvious one is to reduce the huge amount of input data. This is particularly important in a system operating in real-time continuously processing incoming images. Another motivation is that by doing so, complex problems can be divided into simpler subproblems. However, attention does not only mean focusing the processing. It also requires mechanisms for inhibiting and shifting attention, steps often overlooked in machine vision, since systems aspects are seldom included.

In our case we need mechanisms that indicate to our system where the objects that stand out from the background are (see also the introduction to Chapter 2). Following our general goal of investigating an active observer in a 3-D world we will mainly use motion and binocular disparities for the initial attention step. Before we go into detail about this we will first briefly review other attempts to introduce attentional processing in computer vision and discuss their relations to our work.

3.1 Attention in machine vision

The work on attention in computer vision has mainly dealt with finding regions of interest and then tracking them. In many cases the whole process occurs at the retinal level and involves no camera movements for fixational shifts or tracking. Such work is presented by Burt (1988), who describes mechanisms for control of dynamic vision. His approach has three parts: foveation, tracking and high level interpretation. The foveation is formed by a Gaussian pyramid. "Tracking may be said to isolate selected regions of a scene in time just as foveation isolates regions in space." The high level interpretation consists of an object representation, a *pattern tree* which is a multiresolution graph over object features. This graph is used for fast search to identify subpatterns in a scene.

In contrast to this work (Clark and Ferrier, 1988) consider attentional control of a binocular head-eye system. The emphasis is on the control mechanism and tracking is restricted to simple objects such as bright blobs, or objects distinguished by their 0^{th}, 1^{st} or 2^{nd} order moments. In this type of work, as in (Coombs and Brown, 1992) and (Pahlavan and Eklundh, 1993), and the monocular approach by Murray et al. (1993) the system does not really extract or "see" anything in the sense discussed in Chapter 2.2.3 They can mainly be seen as fixation mechanisms.

An interesting computational approach to visual attention is proposed in (Tsotsos et al., 1995). They present an attentional model consisting of a set of hierarchical computations involving the idea of *selective tuning*, which earlier was called *inhibitory beam* (Culhane and Tsotsos, 1992). Their approach consists of a top-down hierarchy of winner-take-all processes. Tsotsos et al. claim that their model matches the theory of the neurobiological model of attention well. A feature that makes this approach difficult to apply in real-time is that it involves a search for what gives rise to the maximal response in order to obtain the focused beam. Although this can be parallelized

it is rather time-consuming on standard hardware, see (Okamoto, 1994) for attempts to overcome this problem.

There are a number of other proposed approaches in the literature. Westelius (1995) uses edge information and rotation symmetry as the features to form a potential field for driving attention. Phase information from quadrature filters is used to generate a potential field drawing attention towards and along lines and edges in the image. Objects are found by a potential field generated by a rotation symmetry estimator. This work has mainly been performed on simulated data and real-time performance is not reported.

Brunnström (1993), and Brunnström et al. (1996) present an active vision approach to classifying corner points to examine the structure of the scene. By actively zooming in on interesting areas in the scene high resolution can be obtained, which facilitates the classification.

The multi-cue work by Milanese (1993) lies closer to ours. He computes a set of feature maps, which are integrated by defining energy-measures. We will discuss this approach in more detail in Chapter 4, since it deals more with computing salience from multiple cues than attention.

In this thesis we have used attention for a few purposes that we will examplify. In **(Nordlund and Uhlin, 1996)** we use attention for pursuit of a moving object. This is thought as an initial step to further investigation of the moving object. See Figure 1.3 for example of results. Analyzing the object will be facilitated by having it in view for a longer period than would be the case if pursuit is not used.

In **(Maki et al., 1998)** we use a depth based attention scheme. The system attends to the person closest to the camera. See Figure 3.1 for results.

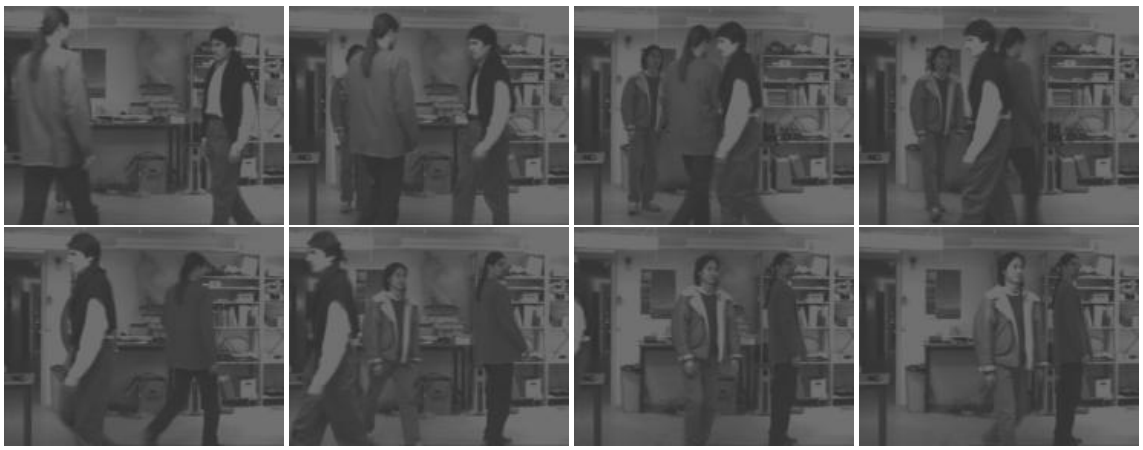
3.2 The importance of 3-D cues

Research on human visual attention emphasizes the role of motion and depth, especially in target localization and pop-out mechanisms that are central here.

For instance, in the case of motion McLeod et al. (1991) propose the existence of a movement filter in the brain. Several authors argue for the existence of separate channels for motion and stationary objects, e.g. (Enroth-Cugell and Robson, 1966; Tolhurst, 1973; Yantis and Jonides, 1984) cited in (McLeod et al., 1991). McLeod et al. (1991)'s theory does not exclude the existence of separate channels, but they claim that there must exist something more than separate channels, namely a motion filter. At least a simple motion filter would be motivated in most creatures with eyes. A moving target may symbolize food or danger, so motion detection is a very basic capability. Another very useful capability is to maintain a moving object in view. Maintaining the moving object in view can facilitate figure-ground segmentation and recognition, since the target will stay in view for a longer time period than it would if not pursued. The algorithm presented in **(Nordlund and Uhlin, 1996)** is a simple movement filter in accordance with what was proposed by McLeod et al. (1991).

By performing psychophysical experiments Ramachandran (1988) conclude that perception of motion must necessarily be preceded by a computation of three-dimensional shape. This observation may seem a bit contradictory with the findings by McLeod et al. (1991), but as stated their findings do not concern motion perception in general, just a simple movement filter.

Treisman (1985) has proposed the well-known so-called feature integration model for preattentive processing. She performs experiments focusing on texture segregation and visual search, where subjects are asked to look for a particular target in displays containing varying numbers of distractor items. If the target is defined by a simple visual feature, detection appears to result from parallel processing; the target "pops out" and the search time is independent of how many distractors surround it. Such experiments turn out to be consistent with the idea that early visual analysis results in separate maps for separate properties, and that these maps pool their activity across locations, allowing rapid access to information about the presence of a target, when it is the only item characterized by a particular preattentively detectable feature. Though there is no complete agreement about the set of basic features, there is agreement about many members of the set, such as color, orientation, size, motion and stereoscopic depth (Wolfe and Cave, 1990). It is



(a) Original sequence.

(b) Target masks, computed using a depth based criterion (nearest object).

Figure 3.1: *An example sequence with 3 moving persons taken by a stationary binocular camera head. Top-left to bottom-right. Every* 10^{th} *frame of the left image is shown (40 msec between frames).*

interesting to note that the search for an item defined by the conjunction of two stimulus dimensions is conducted serially, and the search time increases as the number of items becomes larger. Thus, it seems that the visual system is incapable of conducting a parallel search over two stimulus dimensions simultaneously. Nakayama and Silverman (Nakayama and Silverman, 1986) extend this conclusion for the conjunction of motion and color. Interestingly, they also point out two exceptions through similar experiments: if one of the dimensions in a conjunctive search is stereoscopic disparity, a second dimension of either color or motion can be searched in parallel. We have in our work built upon this observation and used these features, indeed obtaining similar effects in our computer vision experiments. As stated in the introduction this has in fact been one of our key ideas.

3.3 Summary

In this chapter we have here reviewed relevant work concerning bottom-up and pop-out mechanisms and shown the importance of 3-D cues. We have also described how we have used 3-D cues for attention. We have not discussed how task dependence affects such mechanisms in a deeper meaning. However, the approaces we present in **(Uhlin et al., 1995a; Maki et al., 1998)** are open for top-down mechanisms.

Chapter 4

Cue Integration

Traditional computer vision approaches attempt to extract scene characteristics or even fully analyze the scene using single cues such as edges or binocular disparities. Attempts to integrate more information of course exist and an early example is given in the framework proposed by Barrow and Tenenbaum (1978). They present a method for integrating cues such as surface reflectance, distance, orientation, incident illumination, specularity and luminosity, but their approach to a great extent relies on the extraction of more or less perfect edges, which may work in some restricted environments but probably not in a more cluttered environment. In fact, the intrinsic image model was never implemented.

Bülthoff and Mallot (1987) carried out psychophysical experiments to investigate how people perceive 3-D shape given different shape cues. A Bayesian framework solution of how to integrate shape cues for computing 3-D shape is presented in (Bülthoff, 1991). Particularly interesting are the theories for inhibition of conflicting cues. Bülthoff and Mallot (1987) classify depth cues in three categories: 1. primary depth cues that provide direct depth information, e.g. from binocular stereo, 2. secondary depth cues, that may also be present in monocular images, e.g. shading, and 3. cues to flatness, inhibiting the perception of depth, e.g. a frame surrounding a picture. In case of conflict, the primary cues override secondary cues.

Clark and Yuille (1990) present a comprehensive framework for fusion of different types of information, also using a Bayesian framework and also aiming at surface reconstruction. Another proposed approach using in principle analogous techniques, but based on regularization appears in (Shulman and Aloimonos, 1988). See also (Nielsen, 1995; Das and Ahuja, 1995).

Typical of these efforts is that they aim at reconstructing the whole scene rather than selecting areas of interest, according to some criterion. An attempt in the latter direction is given by Milanese (1993) who computes a set of feature maps. On the feature maps "conspicuity"-maps are then computed. The integration is made by defining energy-measures for the conspicuity-maps, both intra-map and inter-map energy functions. The total energy is then to be minimized in a non-linear relaxation scheme. The approach has a sound basis, but no decision is taken before the integration, to exclude feature maps that for some reason are totally wrong, e.g. if the assumed primitives are not observable or missing. Such maps will of course affect the performance in a negative way. To improve performance it would be desirable to exclude false results at an early stage, maybe even before the integration. Milanese's approach is an example of a weakly coupled system (see Section 8.6) in the sense that all feature maps integrated have been extracted independently. Typical for all these approaches is that all the information is fused. A regularization framework, or computed uncertainties provide the only way of discarding or downplaying erroneous information. Such methods therefore only indirectly select between different cues or processes, when these are informative or uninformative, a problem we will return to below.

Studies of cue integration in biological systems have been conducted by many researchers, (see e.g. Nakayama and Silverman, 1986; Grossberg, 1993; Grossberg and Wyse, 1991; Maloney and Landy, 1989). Neural network solutions to the figure-ground problem are presented in (Grossberg, 1993; Grossberg and Wyse, 1991). As mentioned several times we have been inspired by the spe-

cific role depth plays in these observations. Maloney and Landy (1989) suggest an architecture integrating different depth-modules linearly. The contribution from the different modules are weighted depending on ancillary measures. The ancillary measures represent side-information relevant to assessing the quality of the depth information. E.g. egomotion information from the vestibular system should influence the ancillary measure for depth from motion parallax. Bhanu, Roberts and Ming (1989) suggest that inertial sensor information should be used to facilitate motion segmentation. Attempts to do so in machine vision have been reported by e.g. Panerai and Sandini (1997).

When building robot systems that work continuously in time, it soon becomes evident that one single vision algorithm is not enough to get sufficient performance if the system should have more than one functionality. The systems perspective that we advocate and its implications is considered by Hager (1990): "The *task* the system must carry out determines what information is needed and to what level of refinement.". An experimental system containing three components: image processing, tracking and motion, and fusion and planning is presented by Hager (1990), who states that the problem can be seen as information gathering. Since the computational resources are limited, the solution is not to run all possible algorithms at the same time and fuse all the output data. The solution is to *select* which algorithms to run. Moreover, the concept of *information cost* is discussed more extensively in (Hager, 1990, pp 60).

By letting included algorithms produce some kind of reliability measure the selection process could be facilitated. A method to switch between different features, depending on the value of a significance measure, to obtain a better texture segmentation is presented by Porter and Canagarajah (1995). The importance of having included algorithms signaling success or failure is pointed out by Firby (1992), where a robot control architecture is described. Courtney et al. (1997) also point out that algorithms used in a larger system must provide some kind of reliability measures. These can be used to determine when the system should switch between the different algorithms and rely on other cues.

In our work we have so far applied fusion rather than selection techniques in the first step, but selection occurs later, see Chapter 8. **Maki et al. (1998)** combines binocular and motion disparity in series to obtain a depth-motion segmentation of the scene. According to Clark and Yuille (1990), who classify fusion algorithms as being either weakly or strongly coupled, (see Section 8.6) this approach would be classified as weakly coupled fusion. In the work presented in **(Nordlund and Eklundh, 1997)** the same preprocessing steps are used to obtain disparities as in **(Maki et al., 1998)**, but instead the fusion is then strongly coupled. More detailed overviews of these papers are given in Chapter 11.

In summary, we contend that a system capable of functioning in a changing and in part unpredictable environment must rely on multiple cues or processes[1]. Our work in the papers cited in the previous paragraph gives evidence to the converse statement that a system having such capabilities indeed can deal with cluttered environments about which little is known. More details about how the multiple cues aid in this process are provided in the next chapter.

[1]Example: Sometimes there could be different processes working on the same type of information, such as several stereo or motion algorithms. If the algorithms were correlation based, they would fail when no texture was present. On the other hand edge based algorithms would then be useful, since edges would be easy to find.

Chapter 5

Clustering of Information from Multiple Cues

The vast field of clustering algorithms has not been covered in this thesis. We have mainly used histogram methods. A few comparison experiments using the k-means method (Hartigan, 1975), with 2-D histogramming methods can be found in **(Nordlund and Eklundh, 1998)**. There are a number of issues also for such simple approaches, especially in the case of real-time applications.

If no information about the features[1] to be clustered are available the labeling of a large feature set can take surprisingly long time. Many approaches, like e.g. Maximum Likelihood-methods are iterative and thus potentially too slow for real-time implementations. Other methods, like k-means has the disadvantage that the number of classes has to be decided beforehand.

When little or nothing is known about an image and the goal is to extract salient structures having some properties invariant to translation, rotation and scale, a good first step to take is to make a multi-scale representation (e.g. Crowley and Parker, 1984). Lindeberg (1993) has also proposed a method to detect salient image structures, which we have applied on 2-D histograms to partition data.

To decide how many feature dimensions to cluster concurrently is a difficult problem which is also discussed in Chapter 8.6, page 31. We have chosen 2 dimensions as a reasonable tradeoff for achieving real-time performance.

5.1 2-D histogramming

By making a histogram of the data coming from an image, classes of data having uniform values can be detected. Ideally clusters are identified as peaks in the histogram, but having noisy data or data not containing uniform values the question of what a peak really is arises. This problem can in part be solved by a multi-scale approach.

5.1.1 Analyzing the histogram

By having 2 independent feature domains it is possible to produce a 2-D histogram. The feature domains could be e.g. a dense depth map and a dense 1-D motion map as in **(Nordlund and Eklundh, 1997)**. If also confidence maps are available a weighted 2-D histogram can be produced. In most of the examples in **(Nordlund and Eklundh, 1998)** 2 color components obtained by projecting the RGB color cube onto a plane were used (Shuster, 1994).

Peaks in the histogram reflect areas in the image which have approximately constant values in both feature domains at the same time. We have applied the multi-scale method developed by Lindeberg (1993) which detect peaks at different scales. The detected peaks are given a significance measure and the peaks are also tracked over scale. This method works well, and was

[1] Usually extracted from some preprocessing step, e.g. disparity maps, or some image transform.

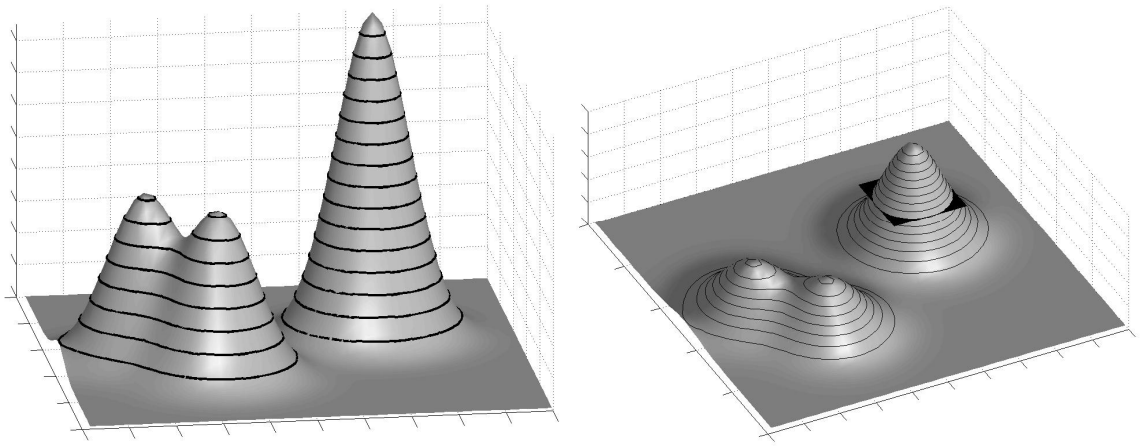


Figure 5.1: *Leftmost image shows level contours of a 2-D histogram to illustrate how the histogram is sliced up. Rightmost image illustrates the bounding box approximation for the same histogram seen from a slightly different angle. The black rectangle is the bounding-box approximation of one of the (in this particular example elliptical) slices of the highest peak.*

used in **(Nordlund and Eklundh, 1997)**, but it has been considered as too computationally demanding for real-time experiments, so a simplified peak-detection scheme has been developed. This simplified method will be described biefly below, for more details see **(Nordlund and Eklundh, 1998)**.

The simplified peak detection scheme

The 2-D histogram is sliced up giving a binary image in each slice. For each connected region in a slice, the bounding box (bb) is computed, see Figure 5.1 for a graphical illustrating example.

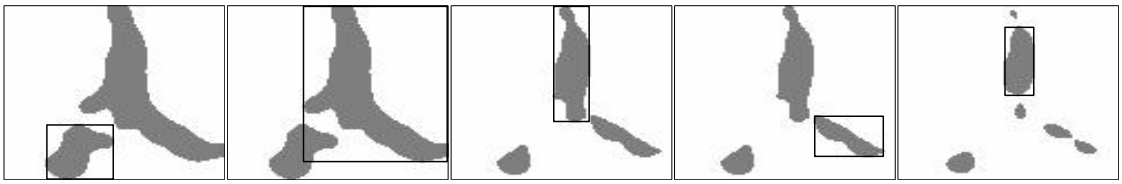
The bounding boxes are matched from the bottom of the histogram and upwards to create blobs (which will be the simplified correspondence to the scale space blobs) by the following scheme: If a bb from a level below overlaps with a bb on the next level a match has been found. This matching gives us three categories of matches:

1. No match.
2. A single match
3. Multiple matches

When case 1 or 3 occurs, the blob is terminated as "no match" or "split". When a split occurs new blobs are created, starting on the level where the split occurred. After this matching we have a number of blobs which are represented by:

1. Their bounding box from their start level (bottom level).
2. Their start and end level.

The blobs can simply and fast be backprojected to the original image, see Figure 5.2 for results form the simplified peak detection scheme. Compare with results from the method used in **(Nordlund and Eklundh, 1997)**, Figure 5.3. This method uses much heavier calculations. The processing time is typically 3 seconds on a SparcStation Ultra, excluding the backprojection. For our new method we get typical processing time including backprojection of less than 0.1 s using a Silicon Graphics Octane workstation. As can be seen in the Figures , the new method gives a somewhat less accurate result in this example, but the speedup is so significant that the method can be used in real-time applications. For more details about the performance of our simplified method, see **(Nordlund and Eklundh, 1998)**.



(a) The found blobs from analysis of a 2-D histogram of horizontal motion and binocular depth disparities. In the sliced histogram shown, motion is in the horizontal direction and relative depth in the vertical direction. The bounding box blob approximation that we use is circumferring the peaks in the histogram. Shown in grey is the histogram slice at the start level of each blob.

(b) Back-projection of the the vertical dimension of the peaks shown in Figure 5.2(a).

(c) Back-projection of the the horizontal dimension of the peaks shown in Figure 5.2(a).

(d) The final backprojection. The backprojections shown in Figures 5.2(b) and 5.2(c) combined with logical AND.

Figure 5.2: *Figure-ground segmentation masks from our simplified method. Each column corresponds to one segmentation.*

By observing the appearance of leftmost column of Figure 5.2 one can easily conclude that there are several occations where a one dimensional histogramming would not suffice to make a segmentation. For example in the leftmost column the depth cue alone would not discriminate the two persons in the foreground since they have approximately the same depth.

5.2 Discussion

Our main tool for integrating different cues has been multidimensional clustering methods. More specifically we have formulated the problem as peak finding in a 2-D histogram. This has allowed us to find regions in the images characterized by the property that they differ from their immediate neighborhood by a conjunction of features. We have developed methods to perform such an analysis fast and robustly, see in particular **(Nordlund and Eklundh, 1998)**. So far we have not investigated higher order conjunctions but our experiments indicate that two may often be sufficient. There are obviously competing techniques, such as the robust Hough transform. It is an open question if such methods provide better results, given the efficiency constraints and the coarse data.

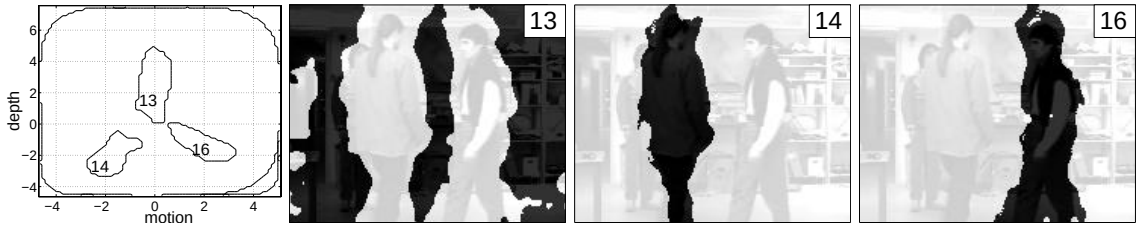


Figure 5.3: *Results from* **(Nordlund and Eklundh, 1997)***, where a method using much heavier calculations was used. Labels in upper right corners of figures refers to blob labels in the leftmost frame. Compare with Figure 5.2.*

Chapter 6

The Visual Front-End

Current research on early processes in computer vision to an increasing extent seem to converge on certain basic principles for early processing. The idea is that in an unpredictable environment the first layer of processing should be unbiased to the input. Results by (e.g. Koenderink and van Doorn, 1992; Florack, 1995) and others show that this assumption leads to approaches that fit well with one of the major awkward problems, namely: *how to keep computational cost down*.

Ideas presented in (Koenderink and van Doorn, 1992; Gårding and Lindeberg, 1996) and earlier by (Marr and Hildreth, 1980; Marr, 1982; Burt, 1988) can be utilized to get good overall performance for a vision system. Most feature extraction algorithms rely on the computation of directional derivatives of low orders and combinations or functions thereof at several different scales. Most certainly a complete vision system will contain feature extraction algorithms, and by having a first layer of retinotopic processing, a Visual Front-End (VFE) in the spirit of (Koenderink and van Doorn, 1987), a highly efficient implementation can be obtained, if the feature extraction algorithms can exploit the derivatives provided by the VFE.

More precisely, by computing a set of low order derivatives at multiple scales in a VFE layer, an output is obtained that can be shared by all the subsequent modules to derive monocular, binocular and motion cues at later stages. We observe that such an approach maps well onto existing hardware architectures. Present 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 are usually performed in coarse grained parallelism.

Burt (1988) have shown how such approaches can be used e.g. in target tracking. In our work we have applied such notions in some of our implementations and experiments. In **(Uhlin et al., 1995a; Maki et al., 1998; Nordlund and Eklundh, 1997)** we use the calculated low order derivatives both for the stereo and motion computations.

There are still more research to do before the wealth of knowledge about VFE-computations have been fully exploited. Recent results on the use of wavelets and tunable filters indicate other options besides using Gaussian filters, as is the case in our work.

In Figure 6.1 is shown an example from **(Uhlin et al., 1995a)** of a system implementation using a VFE. As can be seen we use output from the VFE as input for several of the other modules in the system.

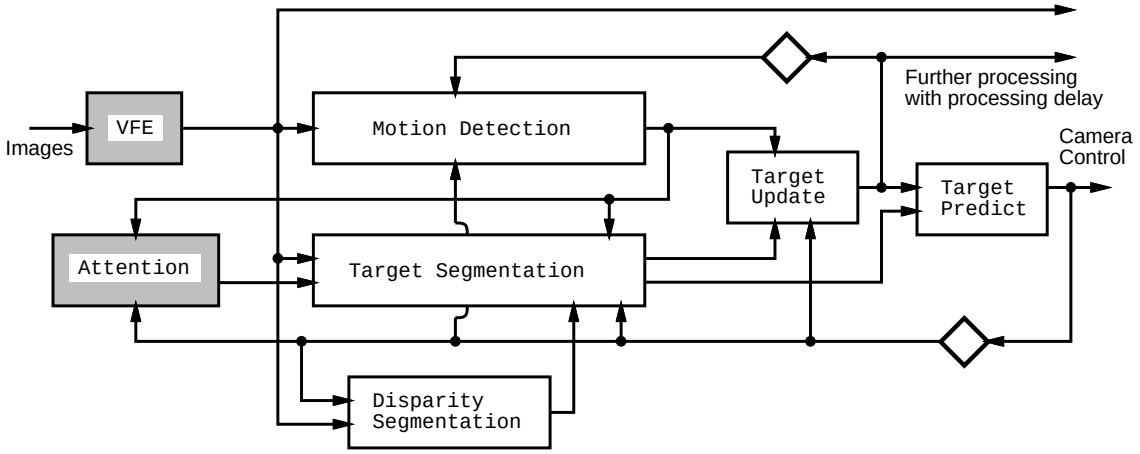


Figure 6.1: *The system implementation is shown schematically. (The diamonds indicate a one frame delay.)*

Chapter 7

Maintenance of Figure-Ground Segmentation

At this point it might be of importance to elaborate somewhat on what we can call maintenance of figure-ground segmentation.

There exist, as we have described, tracking methods that indeed perform a segmentation: rather than only providing a point or rough region location of the object, they provide some more precise outline of the object tracked. In the model-based case, such as e.g. (Lowe, 1991) this is an obvious effect. However, there are also methods dealing with the more general situation here, when no object model exists, or before the system has been able to determine which model applies.

If the object moves or sticks out in depth, we show in e.g. **(Uhlin et al., 1995a)** that it can be segmented out and represented by a mask. From this representation we can compute properties of color, texture and shape that may or may not be used during tracking. There are several reasons why this is important. The system by Toyama and Hager (1996) allow the tracker to switch between these properties, hence obtaining increased robustness. In our work we have utilized this possibility, existing only in systems relying on multiple cues, in a different way. By determining the color or the texture characteristics of the object we have information that allow us to segment out the object if it stops moving, or disappears from the visual field and then is seen again due to occlusions or gaze changes. It is worth noting that these object centered properties may not be distinguishing enough to detect the object if the entire scene is analyzed. It's color or texture may be similar to other areas of the scene .

In Figure 7.1 we show an example of maintenance of figure-ground segmentation: A person is segmented out for several frames, during that time information about the persons velocity and depth is recorded. When the person gets occluded a predict mode is entered. The system remembers the depth and predicts where the person should turn up again after the occlusion has ceased. See **(Nordlund and Eklundh, 1997)** for further details.

To summarize, an integrated system using multiple cues can maintain segmentation during varying circumstances by selecting the distinguishing properties in the particular situation. This provides an additional motivation for an approach such as ours.

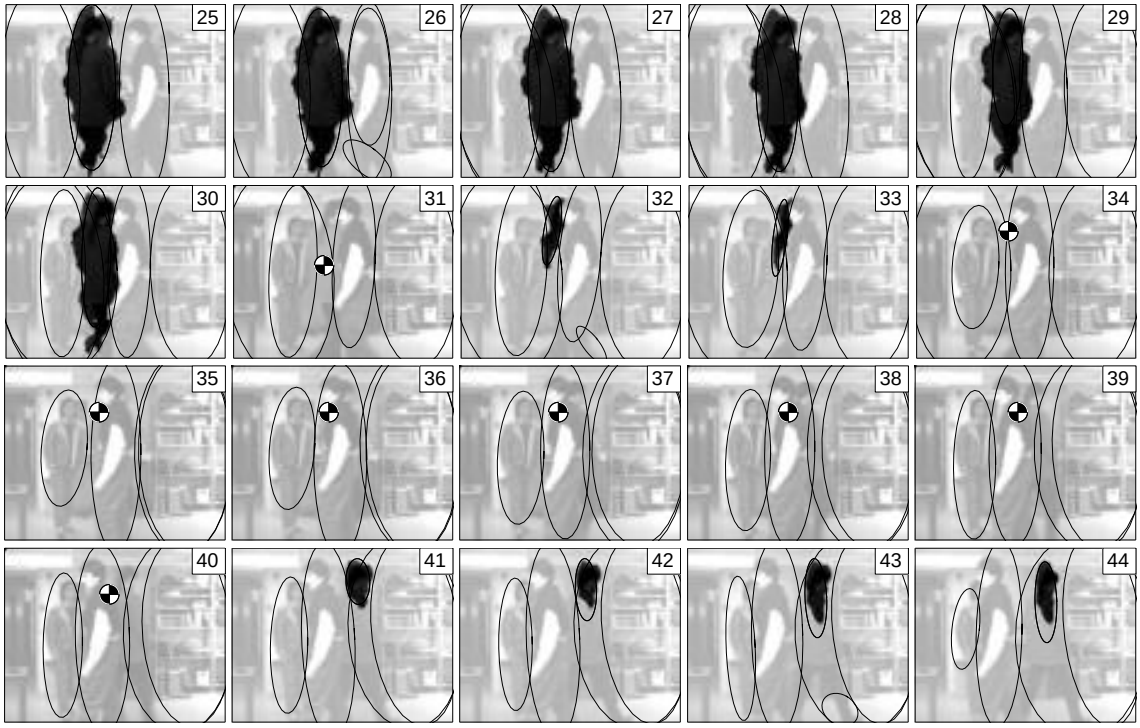


Figure 7.1: *Example of figure ground segmentation, from* **(Nordlund and Eklundh, 1997)**. *An accumulated tracking mask is shown in black. Ellipses shown represents motion masks extracted on a frame to frame basis. In some frames there is a black/white circle showing the predicted centroid of the tracking mask, when the tracking update fails. Frame number is indicated in upper right corner.*

Chapter 8

System Aspects

Studying seeing agents and vision and action from a general perspective is a far-reaching and difficult undertaking. The systems aspects become central, since all components should be regarded in the context of the tasks that the agent performs. Fermüller and Aloimonos (1995) propose a model consisting of "procedures for performing visual perceptions, physical actions, learning, and information retrieval, and purposive representations of the perceptual information, along with representations of information acquired over time and stored in memory", topics that cover most of what is being considered in cognitive science, AI and computer vision.

An important question then becomes what kind of approach can be used to investigate such a system in its entirety. Attempts to develop systems that close the loop between perception and action in real environments, but in limited contexts, certainly exist. One could mention the work on visual guidance of vehicles which has been quite successful, (see e.g. Masaki, 1995) for an overview. Another example is the work in the ESPRIT-project Vision-as-Process, reported in part in (Crowley and Christensen, 1995), aiming at seeing robots in indoor scenarios. However, none of these projects have considered the problems from the basic perspective that Fermüller and Aloimonos do. For instance they do not address systems capable of dealing with varying and complex environments and competent to perform a multiplicity of tasks.

In analogy to Brooks (1986), Fermüller and Aloimonos (1995) suggest a synthetic approach by which a complex system is built up step by step by adding operational models, thereby including more and more competences. Such a constructive approach is what many researchers in active vision have used, although with less ambitious goals. However, it is far from clear how such a methodology can be scientifically founded.

If the work aims at understanding biological vision, obviously the designed models can be tested against empirical data. If on the other hand, the goal is to design artificial vision systems, there is a need both to devise working systems and to provide some formal analysis of it. A formal approach to system design analysis is *discrete event systems* (Ramadge and Wonham, 1987; Ramadge and Wonham, 1989; Košecká, 1996) where tasks are modelled as finite state machines and in each state the underlying system is modelled as a continous system. Issues on scaling must also be addressed, which to date seems to be beyond the state-of-the-art. Systems for which there is an analysis of their functionality, such as the one proposed in (Košecká, 1996), have limited capabilities and address few of the basic issues in vision. Moreover, presented systems have few competences and evidence that they could scale up is missing.

Despite these unanswered questions we believe that a constructive approach is worth pursuing and that it involves design and analysis as well as implementation and experiments. With that in mind we now turn to the two aspects mentioned above.

8.1 Head-eye systems, fixation

The head-eye system used in our experiments was developed by Pahlavan (1993) (see Figure 1.2). For details about the control of the head see (Uhlin, 1996). This head is binocular and designed for

real-time control in a feedback loop. The head was used in a closed feedback loop in **(Nordlund and Uhlin, 1996)** although only one camera was used then. The binocular camera setup was exploited in **(Uhlin et al., 1995a; Maki et al., 1998; Nordlund and Eklundh, 1997)**, where we performed binocular stereo calculations with a phase based method by Maki (1996).

8.2 Continuous operation

A general point to make on the systems approach, is that the time aspect is crucial, (see e.g. Coombs, 1992, chap 6). A seeing system functions over time. Processing is 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 over time. We will here just note that they necessitate experimentation not only with algorithms but also with real complete systems. Some of the problems that arise with a continuously operating system are discussed in (Christensen et al., 1995).

8.3 Feedback Systems

The active vision approach to computer vision has been a trend in the last years (Aloimonos et al., 1988; Ballard, 1991; Bajcsy and Campos, 1992). Aloimonos et al. (1988) argue that an active observer can solve problems in a more efficient way than a passive one. Active vision implies that the system contains a (usually large) amount of feedback.

When making experiments with real-time systems with continuously incoming images it soon becomes obvious that some kind of feedback improves the system, even if the system does not contain any "active" parts, in the sense that some physical parameters have to be changed. It might e.g. be some image level gain or some image thresholding level. Feedback is needed if the system is expected to work in a changing environment.

8.4 Delays

The effect of delays in systems continuously getting visual input have been investigated for feedback loops and control theories for such problems exist, (see e.g. Coombs, 1992, chap 6). That is: How will the system behave if the delay is changed from Δt_1 to Δt_2 ? More difficult to evaluate is the effect of changing the delay in the system by changing the entire system itself, e.g. changing to a coarser resolution during image processing to shorten the delay. Another example would be to lower the number of iterations for an iterative method. Would the overall system perform better if a few iterations are skipped? Skipping a few iterations could give less accuracy in the computations, but on the other hand the delay could be shortened. Iterations could also be performed over time. I.e. instead of iterating over the same old data (data is getting older and older as time goes on, which will happen during iteration), the iteration could be modified to take new data as input. In this case we can get a feedback system (if we take into account the current state of the system). In this example it is difficult to find an optimal delay caused by the number of iterations. See also (Camus, 1997) for a discussion on this topic.

8.5 Reliability measures

Single algorithms based on specific cues will, as pointed out earlier, not suffice for a seeing robot. Such a system should have several redundant algorithms running in parallel to handle the variability and changes in its environment. It should in fact also be able to detect algorithm failures

and then switch between algorithms in a proper way. To integrate the algorithms, both by switching between different ones and by fusing their output the algorithms need a common representation. As the common representation for the different algorithms we have in **(Nordlund and Eklundh, 1997)** chosen a segmentation mask. This is a binary image where the figure-ground segmentation already has been performed to single out the interesting object.

Strategies for deciding whether an algorithm is applicable or not have so far attracted limited attention, both because it is a difficult problem and because few practical attempts to use computer vision algorithms in a system-context have been made. As noted earlier Toyama and Hager (1996) and Firby (1992) briefly touch upon this problem, see also (Bhanu, 1989; Courtney et al., 1997). Toyama and Hager (1996) states that "There are usually geometric constraints or thresholds which can be set so that a tracker will report failure when appropriate". In our opinion this problem must be considered already in the design stage of algorithms. How this should be done is an open question.

There are several reasons why we can't expect algorithms to always give proper results. First the required data may be difficult to observe due to noise or extreme imaging conditions. Then the algorithm will give uncertain results. A somewhat different case occurs when the assumptions that the algorithm builds upon are not valid at all, for instance when the required features are absent, or when the model assumption are violated. Many algorithms for computing disparities or optic flow and obviously also for shape-from-texture need texture to work (see e.g. Karu et al., 1996). Whether an area in the image is textured or not may easily be detected (up to some accuracy not defined here) by some gradient-based measure. This provides a way of introducing uncertainties. However, it is harder to handle violations of assumptions. A shape-from-texture algorithm that assumes that textures are isotropic can be fooled by anisotropic textures, for instance if a receding pattern is painted on a planar surface. Another concrete example of failures that are hard to detect is false matchings in a stereo-algorithm, e.g. when having repetitive patterns in the image.

A well-designed figure-ground segmentation system should calculate reliability measures at many different stages of the segmentation. An example of an early stage reliability measure is the saliency of a region based on the prominence of the corresponding peak in a multidimensional histogram. Reliability measures can also be calculated at much later stages. For examples of reliability measures see **(Nordlund and Eklundh, 1997)** and **(Nordlund and Eklundh, 1998)**.

8.6 Weakly coupled systems vs. strongly coupled systems

(Clark and Yuille, 1990, p 103–104), state: "Weakly coupled fusion is characterized by the combination of the outputs of independent information sources. Each module operates independently of the others" and "Strongly coupled data fusion involves alteration of the prior constraints (either the image formation model, prior model, or system model) used by a information processing module by the output of another module or sets of modules."

Using strong coupling when segmenting has both benefits and drawbacks. The information obtained from a segmentation step exploiting strong coupling between several cues is usually more informative compared to a segmentation using only one cue. On the other hand, if one of the cues turn out to be totally wrong due to some shortcoming of the algorithm, that whole segmentation step can be ruined if strong coupling is used.

Using weak coupling between several cues when segmenting gives less information from each included component, but if one algorithm fails (indicated by its low reliability measure), the system can more robustly recover by relying on backup from the other algorithms included.

8.7 Coarse Algorithms

Our aim is to build a working system having real-time performance by using equipment existing today. For this reason we focus on simple algorithms with low complexity and an algorithmic structure that suits existing high performance hardware. The main purpose of this work is not

to develop new and more accurate algorithms than those existing today. The main goal is to build a system with high performance, which is not necessarily the same as designing algorithms with high performance on single frames, having unlimited amount of time for computations. To take computational cost into consideration is especially important when dealing with systems involving control and feedback in a closed loop. Two things are worth noting, in conclusion. First, our experiments show that a set of coarse algorithms working inside an integrated system can do a surprisingly good job. Secondly, it may be a sensible conjecture that such a system is likely to be more robust than a system relying on a single although very sophisticated algorithm.

Chapter 9

Technical details

9.1 Features

When designing a system using multiple cues it is important to chose features that are not to computationally demanding to extract, preferably features that can be based on a visual front-end (Chapter 6). We have found a few ones that are quite easy to extract with real-time performance.

9.1.1 Corners

A corner detector detects and localizes isolated events in an image (see e.g. Kitchen and Rosenfeld, 1982; Harris and Stephens, 1988; Seeger and Seeger, 1994; Singh and Shneier, 1990; Schmid et al., 1998). It is an interesting alternative for real-time applications since data can be reduced dramatically by applying a corner detector. In some structure-from-motion algorithms, corner detection provides the fundamental data for 3-D reconstruction, (see e.g. Bhanu, Symosek, Ming, Burger, Nasr and Kim, 1989; Burger and Bhanu, 1990; Lee and Kay, 1991; Tomasi and Kanade, 1992; Beardsley et al., 1996).

Seeger and Seeger (1994) present a corner detection algorithm which they claim to have real-time performance, but they do not present any performance figures for their implementation. Singh and Shneier (1990) present a real-time implementation using special dedicated hardware. Wang and Brady (1995) report a frame-rate of 2-3 Hz for 256×256 images On a SUN Sparc-2. With special purpose hardware, a 300 MIPS Transputer machine, where some of the image processing takes place on a DataCube® image processor they report a frame-rate of 14 Hz.

We have made experiments using a corner detector proposed by Kitchen and Rosenfeld (1982). They define the "cornerity" κ as follows:

$$\kappa = \frac{g_{xx}g_y^2 + g_{yy}g_x^2 - 2g_{xy}g_xg_y}{g_x^2 + g_y^2}$$

where $g(x, y)$ is the image function and g_x and g_y denote the derivatives in respective direction.

We extracted corners and then we matched the corners from frame to frame, to obtain a sparse velocity field. Our work is promising but the results are not good enough to present yet. Work remains to finish this issue. In our experiments the k-means clustering algorithm (see e.g. Hartigan, 1975) was used for segmenting the velocity field, no explicit 3-D model of the motion was used. A drawback with the k-means algorithm is that it divides the data in a predefined number of classes. This drawback can be dealt with by dynamically changing the number of classes in a feedback loop.

9.1.2 Color

Extensive reviews/discussions of different 3-dimensional color spaces can be found in (Novak and Shafer, 1992; Perez and Koch, 1994). Perez and Koch (1994) argue that hue is a high level vari-

able due to electrophysiological experiments performed on monkeys and anthropological studies. They also state that the motivation for using hue in image segmentation is that material boundaries correlate more strongly with hue than intensity differences. They state that segmentation in the 1-D hue space is computationally less expensive than in the 3-D RGB space. Of course this is true.

But clearly some information is lost by not taking the saturation into account. By projecting on a plane perpendicularly to the diagonal of the RGB color cube the intensity dependency can be cancelled. This has been practiced by among others Shuster (1994).

In (Ennesser and Medioni, 1995) it is mentioned that the two most important factors affecting the efficiency of any algorithm using color space processing are the *color space* chosen and its *quantization*. They state that they have tested a number of different spaces and found that using the RGB space is "quite good". They also found that more buckets are needed in the case of using a 2-D color space compared to a 3-D space for equivalent results.

As a trade-off for computational speed, but also due to a hardware restriction of OpenGL® on Silicon Graphics "Infinite Reality Systems" we have chosen a 2-dimensional color space, a linear projection on a plane perpendicular to the diagonal of the color cube, (see also Shuster, 1994)

See **(Nordlund and Eklundh, 1997)** for an implementation of a color based image segmentation.

9.1.3 Texture

The only texture segmentation algorithm used in our experiments **(Nordlund and Eklundh, 1997)** is based on gray-scale blobs **(Nordlund and Eklundh, 1997)** and directional derivatives **(Nordlund and Eklundh, 1998)**. By appropriate use of VFE-computations more advanced methods seem feasible, but this has not been used yet.

9.2 Experimental setup

Thorough descriptions of experiments has been neglected in much work on computer vision throughout the years. Comparative studies is difficult, since the descriptions usually lack some important aspect. One particularly interesting aspect that seldom is discussed is the performance of the actual implementation at hand. This problem is discussed in (Christensen and Förstner, 1997, special issue on performance evaluation), see also (Bhanu, 1989; Courtney et al., 1997). Algorithms are often "straightforward to implement on a parallel architecture", but seldom is this done in practice, just to mention one example. In (Barron et al., 1994) a comparative study of optical flow algorithms is presented. Different techniques are compared for accuracy, but the computational complexity is not addressed at all. Other comparative studies have also appeared recently, see e.g. (Heath et al., 1997).

The importance of performing experiments in a closed loop in interaction with the environment must be stressed. When an algorithm is implemented to run continuously the robustness immediately will be tested much more than it would be by running experiments on pre-recorded data. There are several reasons why performing experiments "online" is preferred to "offline" experiments. To our experience the algorithms must be much more robustly designed to work online, since a lots of thresholds must be set adaptively. Having a pre-recorded sequence, there is plenty of time to run the algorithm over and over again, adjusting parameters until it works. This can not be done when experiments are performed online. Offline experiments are often limited by the amount of disk-space available. Running an experiment on say 10000 frames is possible offline, but most probably disk space limitations will not allow experiments on such an amount of images. There is also software quality aspects on this matter. When performing offline experiments quite often systems are put together by having shellscripts invoking other executables such as C/C++-programs. A program leaking memory is no problem if is restarted for each frame. If experiments are performed online for a long extent of time no memory leaks are acceptable, since all available memory will run out very soon. As it is today, results on tracking are presented

with experiments performed on say 10 frames when it would be most desirable to run a tracking algorithm on at least hundreds or thousands of frames. Structure from motion algorithms designated to work on image sequences may be presented with experimental results from a handful of images.

9.2.1 Experimental setup used here

The work presented in this thesis has mainly been performed using the tools described below.

Most prototyping has been performed in Matlab, which we have found to be an excellent tool for rapid image algorithm prototyping.

In **(Nordlund and Uhlin, 1996)** all real-time experiments were performed with the head-eye system constructed by Pahlavan (1993), see Figure 1.2. All images were acquired at a frame-rate of 25 Hz using a MaxVideo 200® as frame-grabber. From one of the cameras[1] only one field was used, this gave us images of the size 366×287 pixels, and a shutter speed of $1/50$ second. The lens used was a zoom lens[2], set to the widest viewing angle: 657 pixels focal length, or 31° and 24.6° viewing angle in the horizontal and vertical direction respectively. The motion detection algorithm was implemented in the C programming language to run on a MaxVideo 200®.

In **(Nordlund and Eklundh, 1997)** the software system was mainly built up in Matlab. Matlab also executed unix commands. Intermediate results were stored on disk and read back into Matlab. The reason for storing results on disk was mainly to shorten execution times during experimentation. Preprocessing such as disparity calculations could then be done offline.

In **(Nordlund and Eklundh, 1998)** all algorithms was implemented in C++[3]. The hardware used was a Silicon Graphics Octane[4] with 1 Gbyte RAM and a SSI graphics board. The camera used was a CCD camera[5]. The analog RGB-signal from the camera was converted to a digital CCIR601 signal by a analog component to serial digital converter[6]. All experiment were performed online. I.e. no intermediate results were stored on disk. Our system has a capacity to operate continuously on images coming directly from the camera. Experimental results presented here come from prerecorded sequences, since, at present, this is the only way to store image-result from the experiments on disk. The prerecorded sequences were grabbed using the program "mediarecorder" provided by SGI. mediarecorder was set to grab images with a resolution of a quarter of a frame, which gives us images of the size 320×243. The images were further cropped to a size of 176×112 pixels.

[1] The robot head is equipped with 2 CCIR cameras, Philips LDH 670.

[2] Camera lenses:Ernitec, model M12Z6. Manufacturers specifications: Focal length 12.5-75 mm, max aperture: F1.2, horizontal angle of view for 2/3" chip: $6^\circ 7' - 38^\circ 7'$, (Max angle of view we practically can use is 31° horizontally).

[3] We used Silicon Graphics mipspro 7.20 compiler.

[4] The machine has two 195 MHZ IP30 Processors, CPU: MIPS R10000, but we only use one of the processors.

[5] A Hitachi KP-D50 1 chip color camera equipped with a Cosmicar Television Lens 6 mm 1:1.2 lens.

[6] An Ensemble Designs, Serial Box III.

Chapter 10

Summary

In developing a seeing agent, such as a robot platform provided with a head-eye system, two issues become central: the importance of a *systems approach*, and the importance of utilizing *multiple cues*.

The systems approach has been utilized in our work when we have put together systems with a whole processing chain starting from images grabbed by the camera and ending with a resulting segmentation mask. We have also presented a system capable of pursuing a moving object controlling the camera in a closed loop with a frame-rate of 25 frames/s **(Nordlund and Uhlin, 1996)**.

After a general discussion on figure-ground segmentation and grouping we stressed the importance of using cues to 3-D, then we showed that it is possible to obtain figure-ground segmentation of objects by exploiting depth and motion. We have made experiments with different types of interactions between the depth and motion module, both in a serial and parallel manner **(Uhlin et al., 1995a; Maki et al., 1998; Nordlund and Eklundh, 1997)**.

We used a multi-scale approach for segmenting a 2-D histogram **(Nordlund and Eklundh, 1997)**. This approach was further simplified to suit our real-time performance restrictions. We show results on segmentation of color images with a frame-rate of up to 18 frames/s **(Nordlund and Eklundh, 1998)**, which supports our belief in the approach.

We argue for the importance of having a system working over time. By utilizing consistency over time, methods giving imprecise results on single frames are greatly improved. This is also shown in experiments **(Uhlin et al., 1995a)**.

The importance of an attentional scheme is stressed as well as approaches based on the idea of a visual front-end. These ideas have been implemented and demonstrated experimentally **(Uhlin et al., 1995a; Maki et al., 1998)**. By attending to subparts of the visual field, the subsequent analysis become simpler, since methods relying on image statistics that would fail if they were applied to the whole image, now succeeds. We have shown an example where a segmentation of an object can be maintained, although the cue that initially formed the segmentation disappears. This goal is reached by gathering characteristics about the pursued object and exploit this knowledge when needed **(Nordlund and Eklundh, 1997)**.

Using object properties such as color for segmentation, is a good approach since it is to a large extent invariant to image transformations such as translation, rotation and scaling. The advantage with segmentation methods not relying on image motion becomes apparent when the method is used together with control of the camera, since moving the camera greatly affect temporal derivatives, which are used in most motion algorithms. Thus color segmentation can be a good complement to methods computing 3-D structure **(Nordlund and Eklundh, 1998)**.

10.1 Open issues

An interesting issue is how the feedback for the system should be designed. There are various levels of feedback in a system, and how these feedback loops should be designed is an open

question. E.g. should disparity fields be fed back from the last frame when computing a new disparity field? The answer is probably yes, but if the disparity field is totally wrong, subsequent computations will be corrupted. The right amount of feedback should be used, not more than that the system manages to recover if calculations are getting wrong. Such problems can be coped with to some extent by having a layered architecture like the one proposed by (Toyama and Hager, 1996).

Temporal aspect of the algorithms could be further investigated. Today most motion-based algorithms, including ours, only use a few frames for computing temporal derivatives. Propagating segmentation information to the visual front-end-layer for doing anisotropic blurring is another interesting topic in the same spirit.

It would also be interesting to include texture based algorithms for segmentation, but so far we have not managed to get a real-time performance on such approaches. This holds also for shape cues in general.

An interesting topic to further investigate is how many different features that should be used in conjunction. Should features be integrated in parallel or in series. The answer is probably both. But much work remains to be done in this area, especially when it comes to having complete systems operating in interaction with the environment, since this puts much harder restrictions both concerning the computational cost and the general robustness of the included algorithms.

Finally it is still an open question how one should evaluate the efficiency of algorithms. How should we know how much time to spend on certain tasks? Should we chose high frame-rate or high resolution? These questions can probably be answered only by implementing complete systems where it is easier to judge the overall performance instead of looking on included algorithms in part. After all, the interesting question is the performance of the *complete system*.

Chapter 11

About the Papers

Common for all the included papers is that the layout have been changed to suit the format of the thesis. E.g. all two column journal and conference articles have been changed to one column style. The contents though are identical to that of the respective publication. One minor detail about references can differ: if a reference was not complete at the time when the article was written, e.g. page numbers were not known, it has now been updated to contain the latest information available.

11.1 Summary of contributions

We have in the papers enclosed in this thesis contributed to the following areas:

- *Fixation.* Uhlin (1996) made most of the work on a real-time environment implementation of a robot-head fixation mechanism. In **(Nordlund and Uhlin, 1996)** Nordlund did the main work on a more sophisticated fixation mechanism than the early implementation (Pahlavan et al., 1993; Pahlavan et al., 1996).

- *Phase based stereo.* Maki (1996) designed a simple and fast phase-based stereo algorithm suitable for real-time implementation which has been used in the experiments in **(Maki et al., 1998)**, **(Uhlin et al., 1995b)** and **(Nordlund and Eklundh, 1997)**. Nordlund's contribution was mainly to provide ideas for experimental setups, and performing experiments.

- *Color segmentation.* Nordlund developed and implemented a real-time color segmentation algorithm suitable for use in a cue-integration system **(Nordlund and Eklundh, 1998)**.

- *Visual front-end.* Nordlund, together with Mårten Björkman implemented a real-time visual front-end consisting of an image pyramid of low-order derivatives, which was used in the experiments in **(Nordlund and Eklundh, 1998)**.

- *Cue integration.* In **(Maki et al., 1998)** we integrate motion detection, motion disparities and binocular disparities to achieve figure-ground segmentation. Nordlund developed a 2-D histogram based method that is used to achieve figure-ground segmentation. The importance of reliability measures for algorithms is discussed **(Nordlund and Eklundh, 1997)**. The histogram method in **(Nordlund and Eklundh, 1998)** is a further development to achieve real-time performance. Moreover we show real-time results for a method also using corner features.

- *Figure-ground segmentation, maintenance of figure-ground segmentation.* In the joint work **(Maki et al., 1998)** we use motion detection, motion disparities and binocular disparities to achieve figure-ground segmentation. In **(Nordlund and Eklundh, 1997)** Nordlund used motion and binocular disparities to achieve figure-ground segmentation. Here is also demonstrated how an initially obtained figure-ground segmentation can be maintained by switching to

another cue when the initial cue disappears. In **(Nordlund and Eklundh, 1998)** Nordlund shows how a method exploiting motion parallax is applied on a static scene to indicate how an object stands out in 3-D. This object is further segmented, and the segmentation is maintained over time using color information.

- *Experimental work on image sequences.* In all of the enclosed papers we present results coming from real image sequences. Nordlund performed a substantial part of the experiments. In **(Nordlund and Uhlin, 1996)** we present an integrated system consisting of a Transputer network and a Datacube MaxVideo 200 pipeline processor which operates on continuously captured images. In **(Nordlund and Eklundh, 1998)** we present a system also operating on continuously captured images. This time we used a Silicon Graphics Octane with a SSI graphics board. We have also provided a thorough description of the experimental setup comparing with what is customary within the field of computer vision.

- *Systems approach.* In the spirit of earlier work by Pahlavan and Uhlin in our laboratory Nordlund has presented systems with a capacity of operating continuously on images coming directly from the camera, thus we prove that our systems consists of a complete processing chain, with no links missing[1]. For early work of a complete system where Nordlund contributed with programming and design, see (Bernoville et al., 1994). A robot with stereo cameras mounted on the arm was visually guided by an integrated model-based recognition and tracking system. The system was running on a SUN3 workstation and the tracking update had a frame-rate of a few Hz. The object recognition part of the system is described in (Andersson and Nordlund, 1993).

11.2 Paper A: Closing the Loop: Detection and Pursuit of a Moving Object by a Moving Observer

The journal version **(Nordlund and Uhlin, 1996)** of this paper is included in the thesis. There also exists a technical report with identical contents as the journal contribution, moreover there is a shortened conference contribution[2].

In the paper we present an integrated system, which is able to pursue a moving object and maintain the object centered in the image, by controlling a robot-head. The system contains three parts, independent motion detection, tracking, and control of a robot-head. The paper focuses on the detection mechanism and briefly discusses the tracking and control issues. The system runs continuously in time and updates the object localization at a frame-rate of 25 Hz. The moving object can be tracked although the observer performs an unknown independent motion, involving both translation and rotation. The major part of the image processing takes place on a MaxVideo200 pipeline processor. The robot head motor control runs on the Transputer network and involves both eye and neck movements, see (Pahlavan, 1993; Pahlavan et al., 1992; Pahlavan and Eklundh, 1994)

The real-time algorithm works on data coming directly from the cameras.

Contribution: Tomas Uhlin and I contributed equally to the theory that was used in this paper, I did most of the writing. The implementation was about equally divided between the two of us. The Head-Eye system was developed by Pahlavan (1993). The Head-Eye controller and the software for integrating the Transputer network and the MaxVideo200 board, both of which were used in this work was developed by Uhlin (1996).

[1] We do not claim that our systems are a complete *solution*, just a complete processing chain.
[2] For reference of the not included papers, see p 1.

11.3 Paper B, C: Towards an Active Visual Observer

Both a long version **(Uhlin et al., 1995b)** and a short version **(Uhlin et al., 1995a)** of the paper are included in the thesis.

In this paper we present a binocular active vision system that can attend to and fixate a moving target. The system 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, both stationary and moving, targets, and integrate three cues to obtain an overall robust behavior, ego-motion, target motion and target disparity.

The integration of many cues and concurrent modules, rather than sophisticated recovery mechanisms, forms a pervading theme. We stress that an active system must be equipped with means and clues to change its attention, while it otherwise is purely reactive. This system is therefore equipped with motion detection for changing attention and pursuit for maintaining attention, both of which run concurrently.

The presented algorithms were implemented in the C programming language. Experiments were performed partly in real-time. A pursuit algorithm with real-time performance was running during image capture, but the major part of the image processing was done after the image capture on the images stored on disk.

Contribution: Tomas Uhlin and I contributed to the theory on motion that was used in this paper. Atsuto Maki contributed with the work on binocular disparity. Tomas Uhlin did most of the writing. The implementation of motion algorithms was about equally divided between Tomas Uhlin and me.

11.4 Paper D: Attentional Scene Segmentation: Integrating Depth and Motion from Phase

An intended journal version "Attentional Scene Segmentation: Integrating Depth and Motion from Phase" **(Maki et al., 1998)** of the technical report "A Computational Model of Depth-Based Attention"is included in the thesis. This report also exists as a conference contribution[3].

In this paper a computational approach to attention is presented. It consists of an early parallel stage with preattentive cues followed by a later serial stage, where the cues are integrated. We base the approach on disparity, image flow and motion detection. We demonstrate a system integrating these cues, in such a way that the attention is maintained on the closest moving object. We show results from experiments in which a moving observer selectively masks out different moving objects in real scenes.

All algorithms were implemented in the C programming language. Modules were integrated in shellscripts. Intermediate results were stored on disk. No real-time performance was obtained.

Contribution: I contributed to the theory on motion that was used in this paper. Atsuto Maki contributed with the work on binocular disparity. Atsuto Maki and I both contributed to the work on integration of the algorithms. Atsuto Maki did most of the writing.

11.5 Paper E: Towards a Seeing Agent

The conference version "Towards a Seeing Agent" **(Nordlund and Eklundh, 1997)** of the technical report "Maintenance of Figure-Ground Segmentation by Cue-Selection"[4] is included in the thesis.

[3]For reference of the not included papers, see p 2.
[4]For reference of the not included report, see p 2.

Note that the title has been changed, apart from that the contents are identical. An approach to figure-ground segmentation based on a conjunction of motion and depth is presented. The main idea is to produce a 2-dimensional histogram with depth in one dimension and horizontal motion in the other. This histogram is then analyzed. The most significant peaks in the histogram are backprojected to the image to produce an object mask. This object mask is maintained over time.

A cue-selection algorithm using both the depth-motion algorithm and another feature-based algorithm is also demonstrated. This combined algorithm manages to maintain the object mask although the depth-motion algorithm alone fails.

The software system used was mainly built up in Matlab. Matlab calls to a few algorithms implemented in C/C++ were also used.

Contribution: I contributed to most of the ideas, much inspired by my supervisor Jan-Olof Eklundh. I performed all of the experiments. Atsuto Maki contributed with some experimental data. Tony Lindeberg provided us with his implementation of the scale-space primal sketch (Lindeberg, 1993) which was used in the experiments.

11.6 Paper F: Real-time Maintenance of Figure-Ground Segmentation

The technical report **(Nordlund and Eklundh, 1998)** is included in the thesis. It will be submitted to Int. Conf. on Vision Systems, Jan 99. An approach to figure-ground segmentation based on a 2-dimensional histogram of a transformation of the 3-dimensional color-cube is presented. The histogram is then analyzed with a peak-finding algorithm designed with real-time performance in mind. The most significant peaks in the histogram are backprojected to the image to produce an object mask. This algorithm replaces the one by Lindeberg mentioned above. A cue-selection algorithm using both the color algorithm and another image motion based algorithm is also demonstrated.

An integrated system with capabilities to operate continuously grabbing images directly from a CCD-camera with real-time performance having typical frame-rates of about 10 Hz is presented. The visual front-end has also been implemented.

Contribution: I contributed most of the ideas, much inspired by my supervisor Jan-Olof Eklundh. I did all the programming and performed all of the experiments.

Bibliography

Aloimonos, J. Y., Weiss, I. and Bandyopadhyay, A. (1988). Active vision, *Int. J. of Computer Vision* **1**(4): 333–356.

Andersson, M. and Nordlund, P. (1993). A model-based system for localization and tracking, *Proc. Workshop on Computer Vision for Space Applications*, Antibes, France.

Bajcsy, R. and Campos, M. (1992). Active and exploratory perception, *Computer Vision, Graphics, and Image Processing:Image Understanding* **56**(1): 31–40.

Ballard, D. H. (1991). Animate vision, *Artificial Intelligence* **48**: 57–86.

Barron, J. L., Fleet, D. J. and Beauchemin, S. S. (1994). Performance of optical flow techniques, *Int. J. of Computer Vision* **12**(1): 43–77.

Barrow, H. G. and Tenenbaum, J. M. (1978). Recovering intrinsic scene characteristics from images, *in* A. Hanson and E. Riseman (eds), *Computer Vision Systems*, Academic Press, New York, pp. 3–26.

Barrow, H. and Tenenbaum, J. (1993). Retrospective on "interpreting line drawings as three-dimensional surfaces", *Artificial Intelligence* **59**(1-2): 71–80.

Baumberg, A. and Hogg, D. (1994). Learning flexible models from image sequences, *in* J.-O. Eklundh (ed.), *Proc. 3rd European Conference on Computer Vision*, Vol. 800 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, Stockholm, Sweden, pp. 299–308.

Beardsley, P., Torres, P. and Zisserman, A. (1996). Acquisition from extended image sequences, *in* B. Buxton and R. Cipolla (eds), *Proc. 4th European Conference on Computer Vision*, Vol. 1065 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, Cambridge, UK, pp. 683–695.

Bernoville, J.-P., Dhome, M., Andersson, M., Nordlund, P., Berthod, M., Lamarre, H., Guilloux, Y. L., Monier, G., Lapreste, J.-T., Daucher, N., Rives, G., Lavest, J. M., Eklundh, J.-O. and Giraudon, G. (1994). Stereovision, final report, *Study report AS94–039*, ESTEC, Noordwijk, Holland.

Bhanu, B. (1989). Understanding scene dynamics, *Proc. Image Understanding Workshop 1989*, San Mateo, CA, pp. 147–164.

Bhanu, B., Roberts, B. and Ming, J. (1989). Inertial navigation sensor integrated motion analysis, *Proc. Image Understanding Workshop 1989*, San Mateo, CA, pp. 747–763.

Bhanu, B., Symosek, P., Ming, J., Burger, W., Nasr, H. and Kim, J. (1989). Qualitative target motion detection and tracking, *Proc. Image Understanding Workshop 1989*, San Mateo, CA, pp. 370–398.

Biederman, I. (1985). Human image understanding: recent research and a theory, *Computer Vision, Graphics, and Image Processing* **32**: 29–73.

Black, M. and Jepson, A. D. (1996). Estimating optical flow in segmented images using variable-order parametric models with local deformations, *IEEE Trans. Pattern Analysis and Machine Intell.* **18**(10): 972–986.

Blake, A. and Zisserman, A. (1987). *Visual Reconstruction*, MIT Press, Cambridge, Massachusetts.

Bobick, A. F. (1997). Movement, activity and action: the role of knowledge in the perception of motion, *Philosophical Transactions of the Royal Society of London* **B352**(1358): 1257–1265.

Brooks, R. (1986). A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation* **2**: 14–23.

Brunnström, K. (1993). *Active Exploration of Static Scenes*, Ph. D. dissertation, Dept. of Numerical Analysis and Computing Science, KTH, Stockholm, Sweden. ISRN KTH/NA/P--93/29--SE.

Brunnström, K., Eklundh, J.-O. and Uhlin, T. (1996). Active fixation for scene exploration, *Int. J. of Computer Vision* **17**: 137–162.

Bülthoff, H. H. (1991). Shape from X: Psychophysics and compution, *in* M. Landy and J. A. Movshon (eds), *Computational Models of Visual Processing*, MIT Press, Cambridge, Ma, pp. 305–330.

Bülthoff, H. H. and Mallot, H. A. (1987). Interaction of different modules in depth perception, *Proc. 1st International Conference on Computer Vision*, London, England, pp. 295–305.

Burger, W. and Bhanu, B. (1990). Estimating 3-d egomotion from perspective image sequences, *IEEE Trans. Pattern Analysis and Machine Intell.* **12**(11): 1040–1058.

Burt, P. J. (1988). Attention mechanisms for vision in a dynamic world, *Proc. 9th International Conference on Pattern Recognition*, IEEE Computer Society Press, Rome, Italy, pp. 977–987.

Camus, T. (1997). Real-time quantized optical flow, *Real-Time Imaging* **3**(2): 71–86.

Christensen, H. I. and Förstner, W. (1997). Performance characteristics of vision algorithms, *Machine Vision and Applications* **9**(5/6): 215–218. Special issue on: Performance Evaluation.

Christensen, H. I., Granum, E. and Crowley, J. L. (1995). System integration and control, *in* J. L. Crowley and H. I. Christensen (eds), *Vision-as-Process*, Basic Research Series, Springer Verlag, Berlin, Heidelberg, pp. 9–22. ISBN 3-540-58143-X.

Clark, J. J. and Ferrier, N. J. (1988). Modal control of an attentive vision system, *Proc. 2nd International Conference on Computer Vision*, IEEE Computer Society Press, pp. 514–523.

Clark, J. J. and Yuille, A. L. (1990). *Data Fusion for Sensory Information Processing Systems*, The Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, Dordrecht, Netherlands. ISBN 0-7923-9120-9.

Coombs, D. (1992). *Real-Time Gaze Holding in Binocular Robot Vision*, Ph. D. dissertation, Dept. of Computer Science, University of Rochester, New York.

Coombs, D. and Brown, C. (1992). Real-time smooth pursuit tracking for a moving binocular robot, *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, 1992*, Champaign, IL, pp. 23–28.

Courtney, P., Thacker, N. and Clark, A. F. (1997). Algorithmic modeling for performance evaluation, *Machine Vision and Applications* **9**(5/6): 219–228. Special issue on: Performance Evaluation.

Crowley, J. L. and Christensen, H. I. (eds) (1995). *Vision-as-Process*, Basic Research Series, Springer Verlag, Berlin, Heidelberg. ISBN 3-540-58143-X.

Crowley, J. and Parker, A. (1984). A representation for shape based on peaks and ridges in the difference of low-pass transform, *IEEE Trans. Pattern Analysis and Machine Intell.* **6**(2): 156–169.

Culhane, S. M. and Tsotsos, J. K. (1992). An attentional prototype for early vision, *in* G. Sandini (ed.), *Proc. 2nd European Conference on Computer Vision*, Vol. 588, Santa Margherita Ligure, Italy, pp. 551–560.

Darrel, T., Gordon, G., Woodfill, J., Baker, H. and Harville, M. (1998). Robust, real-time people tracking in open environments using integrated stereo, color, and face detection, *IEEE Workshop on Visual Surveillance*, IEEE Computer Society Press, Bombay, India, pp. 26–32.

Das, S. and Ahuja, N. (1995). Performance analysis of stereo, vergence, and focus as depth cues for active vision, *IEEE Trans. Pattern Analysis and Machine Intell.* **17**(12): 1213–1219.

Davis, J. W. and Bobick, A. F. (1997). The representation and recognition of action using temporal templates, *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, 1997*, IEEE Computer Society Press, San Juan, Puerto Rico, pp. 928–934.

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*, John Wiley and Sons, Chichester, England.

Eklundh, J.-O. (1996). Machine vision research at CVAP - an introduction, *Int. J. of Computer Vision* **17**: 107–112.

Eklundh, J. O., Yamamoto, H. and Rosenfeld, A. (1980). A relaxation method for multispectral pixel classification, *IEEE Trans. Pattern Analysis and Machine Intell.* **2**(1): 72–75.

Ennesser, F. and Medioni, G. (1995). Finding Waldoo, or focus of attention using local color information, *IEEE Trans. Pattern Analysis and Machine Intell.* **17**(8): 805–809.

Enroth-Cugell, C. and Robson, J. (1966). The contrast sensivity of the retinal ganglion cells of the cat, *J. of Physiology* **187**: 517–552.

Fermüller, C. and Aloimonos, Y. (1995). Vision and action, *Image and Vision Computing* **13**(10): 725–744.

Firby, R. J. (1992). Building symbolic primitives with continuous control routines, *First International Conference on AI Planning Systems*, College Park MD.

Fischler, M. A. and Bolles, R. C. (1986). Perceptual organization and curve partitioning, *IEEE Trans. Pattern Analysis and Machine Intell.* **8**(1): 100–105.

Florack, L. (1995). *The syntactical structure of scalar images*, Ph. D. dissertation, University of Utrecht, The Netherlands.

Gårding, J. and Lindeberg, T. (1996). Direct computation of shape cues using scale-adapted spatial derivative operators, *Int. J. of Computer Vision* **17**(2): 163–191.

Gilbert, A. L., Giles, M. K., Flachs, G. M., Rogers, R. B. and U, Y. H. (1978). A real-time video tracking system using image processing techniques, *IJCPR-78*, Kyoto, Japan, pp. 1111–1115.

Gilbert, A. L., Giles, M. K., Flachs, G. M., Rogers, R. B. and U, Y. H. (1980). A real-time video tracking system, *IEEE Trans. Pattern Analysis and Machine Intell.* **2**(1): 47–56.

Grossberg, S. (1993). A solution of the figure-ground problem for biological vision, *Neural Networks* **6**(4): 463–483.

Grossberg, S. and Wyse, L. (1991). A neural network architechture for figure-ground separation of connected scenic figures, *Neural Networks* **4**(6): 723–742.

Hager, G. D. (1990). *Task-Directed Sensor Fusion and Planning. A Computational Approach*, The Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, Dordrecht, Netherlands. ISBN 0-7923-9108-X.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector, *Proc. Fourth Alvey Vision Conference*, Manchester, UK, pp. 147–151.

Hartigan, J. A. (1975). *Clustering Algorithms*, John Wiley and Sons, New York.

Heath, M. D., Sarkar, S., Sanocki, T. and Bowyer, K. W. (1997). A robust visual method for assessing the relative performance of edge-detection algorithms, *IEEE Trans. Pattern Analysis and Machine Intell.* **19**(12): 1338–1359.

Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow, *Artificial Intelligence* **17**: 185–203.

Irani, M. and Peleg, S. (1993). Motion analysis for image enhancement: Resolution, occlusion, and transparency, *Journal of Visual Communication and Image Representation* **4**(4): 324–335.

Isard, M. and Blake, A. (1996). Contour tracking by stochastic propagation of conditional density, *in* B. Buxton and R. Cipolla (eds), *Proc. 4th European Conference on Computer Vision*, Vol. 1064 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, Cambridge, UK, pp. 343–356.

Isard, M. and Blake, A. (1998). A mixed-state condensation tracker with automatic model-switching, *Proc. 6th International Conference on Computer Vision*, Bombay, India, pp. 107–112.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, New Jersey.

Karu, K., Jain, A. K. and Bolle, R. M. (1996). Is there any texture in the image?, *Proc. 13th International Conference on Pattern Recognition*, Vol. II, IEEE Computer Society Press, Vienna, Austria, pp. 770–774.

Kass, M., Witkin, A. and Terzopoulos, D. (1987). Snakes: active contour models, *Proc. 1st International Conference on Computer Vision*, IEEE Computer Society Press, London, England, pp. 259–268.

Kitchen, L. and Rosenfeld, A. (1982). Gray-level corner detection, *Pattern Recognition Letters* **1**(2): 95–102.

Koenderink, J. J. and van Doorn, A. J. (1992). Generic neighborhood operators, *IEEE Trans. Pattern Analysis and Machine Intell.* **14**(6): 597–605.

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

Koffka, K. (1935). Principles of gestalt psychology, *Harcourt, Brace and World*, New York.

Košecká, J. (1996). *A Framework for Modeling and Verifying Visually Guided Agents, Analysis and Experiments*, Ph. D. dissertation, Dept of Computer and Information Science, Univ of Pennsylvania.

Leclerc, Y. G. (1989). Constructing simple stable descriptions for image partitioning, *Int. J. of Computer Vision* **3**(1): 73–102.

Lee, S. and Kay, Y. (1991). A kalman filter approach for accurate 3-d motion estimation from a sequence of stereo images, *Computer Vision, Graphics, and Image Processing:Image Understanding* **54**(2): 244–258.

Leung, T. and Malik, J. (1996). Detecting, localizing and grouping repeated scene elements from an image, *in* B. Buxton and R. Cipolla (eds), *Proc. 4th European Conference on Computer Vision*, Vol. 1064 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, Cambridge, UK, pp. 546–555.

Lindeberg, T. (1993). Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention, *Int. J. of Computer Vision* **11**(3): 283–318. Also ISRN KTH/NA/P--93/33--SE.

Little, J. J. (1997). Robot partners: Collaborative perceptual robotic systems, *First Int. Workshop on Cooperative Distributed Vision*, Kyoto, Japan, pp. 143–164.

Lowe, D. G. (1985). *Perceptual Organization and Visual Recognition*, Kluwer, Boston, chapter 7.

Lowe, D. G. (1991). Fitting parameterized three-dimensional models to images, *IEEE Trans. Pattern Analysis and Machine Intell.* **13**(5): 441–450.

Maki, A. (1996). *Stereo Vision in Attentive Scene Analysis*, Ph. D. dissertation, Dept. of Numerical Analysis and Computing Science, KTH, Stockholm, Sweden. ISRN KTH/NA/P--96/07--SE.

Maki, A., Nordlund, P. and Eklundh, J.-O. (1998). Integrating depth and motion from phase for attention. Extended version of tech report ISRN KTH/NA/P--96/05--SE submitted to Computer Vision and Image Understanding.

Malik, J. (1996). On binocular viewed occlusion junctions, *in* B. Buxton and R. Cipolla (eds), *Proc. 4th European Conference on Computer Vision*, Vol. 1064 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, Cambridge, UK, pp. 167–174.

Maloney, L. T. and Landy, M. S. (1989). A statistical framework for robust fusion of depth information, *in* W. A. Pearlman (ed.), *Proc. SPIE: Visual Communication and Image Processing IV*, Vol. 1199, Philadelphia, Pennsylvania, pp. 1154–1163.

Marr, D. (1976). Early processing of visual information, *Philosophical Transactions of the Royal Society of London* **B-275**: 483–524.

Marr, D. G. (1982). *Vision*, W.H. Freeman, San Francisco, CA.

Marr, D. and Hildreth, E. (1980). Theory of edge detection, *Proc. Royal Soc. London* **207**: 187–217.

Masaki, I. (ed.) (1995). *Proc. Int. Symp. on Intelligent Vehicles '95*, Detroit, Mi.

McLeod, P., Driver, J., Dienes, Z. and Crisp, J. (1991). Filtering by movement in visual search, *J. of Exp. Psychology: Human Perception and Performance* **17**(1): 55–64.

Milanese, R. (1993). *Detecting Salient Regions in an Image: From Biological Evidence to Computer Implementation*, Ph. D. dissertation, Dept. of Computer Science, Univ of Geneva, Switzerland.

Mumford, D. and Shah, J. (1985). Boundary detection by minimizing functionals, *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, 1985*, IEEE Computer Society Press, San Francisco, CA, pp. 22–26.

Murray, D. W., McLaughlin, P. F., Read, I. D. and Sharkey, P. M. (1993). Reactions to peripherial image motion using a head/eye platform, *Proc. 4th International Conference on Computer Vision*, Berlin, Germany, pp. 403–411.

Nakayama, K. and Silverman, G. H. (1986). Serial and parallel processing of visual feature conjunctions, *Nature* **320**(20): 264–265.

Nielsen, M. (1995). *From Paradigm to Algorithms in Computer Vision*, Ph. D. dissertation, Dept. of Computer Science, Univ of Copenhagen, Denmark. ISSN 0107-8283.

Nordlund, P. and Eklundh, J.-O. (1997). Towards a seeing agent, *First Int. Workshop on Cooperative Distributed Vision*, Kyoto, Japan, pp. 93–123. Also in tech report ISRN KTH/NA/P--97/05--SE.

Nordlund, P. and Eklundh, J.-O. (1998). Real-time figure-ground segmentation, *Technical Report ISRN KTH/NA/P--98/04--SE*, Dept. of Numerical Analysis and Computing Science, KTH, Stockholm, Sweden. Will be submitted to Int. Conf. on Vision Systems, Jan 99.

Nordlund, P. and Uhlin, T. (1996). Closing the loop: Detection and pursuit of a moving object by a moving observer, *Image and Vision Computing* **14**(4): 265–275.

Nordström, N. (1990a). Biased anisotropic diffusion-a unified regularization and diffusion approach to edge detection, *in* O. Faugeras (ed.), *Proc. 1st European Conference on Computer Vision*, Vol. 427 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, Antibes, France, pp. 18–27.

Nordström, N. (1990b). Biased anisotropic diffusion: A unified regularization and diffusion approach to edge detection, *Image and Vision Computing* **8**(4): 318–327.

Novak, C. L. and Shafer, S. A. (1992). Color vision, *Encyclopedia of Artificial Intelligence* pp. 192–202.

Ohlander, R. (1976). *Analysis of Natural Scenes*, PhD thesis, Carnigie-Mellon Univ. Pittsburgh, PA.

Okamoto, K. (1994). An attentional mechanism applied to moving objects, *Technical Report ISRN KTH/NA/P--94/14--SE*, Dept. of Numerical Analysis and Computing Science, KTH, Stockholm, Sweden.

Pahlavan, K. (1993). *Active Robot Vision and Primary Ocular Processes*, Ph. D. dissertation, Dept. of Numerical Analysis and Computing Science, KTH, Stockholm, Sweden. ISRN KTH/NA/P--93/16--SE.

Pahlavan, K. and Eklundh, J.-O. (1993). Head, eyes and head-eye systems, *Int J. of Pattern Recognition and Artificial Intelligence* **7**(1): 33–49.

Pahlavan, K. and Eklundh, J.-O. (1994). Mechatronics of active vision, *Mechatronics* **4**(2): 113–123.

Pahlavan, K., Uhlin, T. and Eklundh, J.-O. (1992). Integrating primary ocular processes, *in* G. Sandini (ed.), *Proc. 2nd European Conference on Computer Vision*, Vol. 588 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, Santa Margherita Ligure, Italy, pp. 526–541.

Pahlavan, K., Uhlin, T. and Eklundh, J.-O. (1993). Dynamic fixation, *Proc. 4th International Conference on Computer Vision*, IEEE Computer Society Press, Berlin, Germany, pp. 412–419.

Pahlavan, K., Uhlin, T. and Eklundh, J.-O. (1996). Dynamic fixation and active perception, *Int. J. of Computer Vision* **17**: 113–135.

Panerai, F. and Sandini, G. (1997). Role of visual and inertial information in gaze stabilization, *in* H. I. Christensen, C. Bräutigam and C. Ridderström (eds), *Proc. of the 5th International Symposium on Intelligent Robotic Systems 1997*, KTH, Stockholm, Sweden, pp. 111–120.

Perez, F. and Koch, C. (1994). Towards color image segmentation in analog VLSI: Algorithm and hardware, *Int. J. of Computer Vision* **12**(1): 17–42.

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Analysis and Machine Intell.* **12**(7): 629–639.

Porter, R. and Canagarajah, N. (1995). A robust automatic clustering scheme for image segmentation using wavelets, *in* E. K. Teo (ed.), *Proc. 2nd Asian Conference on Computer Vision*, Vol. 2, Singapore, pp. 76–80.

Ramachandran, V. S. (1988). Perception of shape from shading, *Nature* **331**(6152): 163–166.

Ramadge, P. J. and Wonham, W. M. (1989). The control of discerete event systems, *Proceedings of the IEEE* **77**(1): 81–97.

Ramadge, P. and Wonham, W. (1987). Supervisory control of a class of discrete event processes, *SIAM J. Contr. Optimization* **25**(1): 206–230.

Rock, I. (1997). *Indirect Perception*, MIT Press, Cambridge, Massachusetts.

Rock, I. and Brosgole, L. (1964). Grouping based on phenomenal proximity, *J. of Exp. Psychology* **67**: 531–538.

Schachter, B. J., Davis, L. S. and Rosenfeld, A. (1979). Some experiments in image segmentation by clustering of local feature values, *Pattern Recognition* **11**(1): 19–28.

Schmid, C., Mohr, R. and Bauckhage, C. (1998). Comparing and evaluating interest points, *Proc. 6th International Conference on Computer Vision*, Bombay, India, pp. 230–235.

Seeger, U. and Seeger, R. (1994). Fast corner detection in gray-level images, *Pattern Recognition Letters* **15**(7): 669–675.

Sha'ashua, A. and Ullman, S. (1988). Structural saliency: the detection of globally salient structures using a locally connected network, *Proc. 2nd International Conference on Computer Vision*, IEEE Computer Society Press, pp. 321–327.

Shen, X. Q. and Hogg, D. (1995). 3d shape recovery using a deformable model, *Image and Vision Computing* **13**(5): 377–383.

Shi, J. and Malik, J. (1998). Motion segmentation and tracking using normalized cuts, *Proc. 6th International Conference on Computer Vision*, Bombay, India, pp. 1154–1160.

Shulman, D. and Aloimonos, Y. (1988). (non-) rigid motion interpretation: A regularized approach, *Proceedings of Royal Society of London* **B-233**: 217–234.

Shuster, R. (1994). Color object tracking with adaptive modeling, *Proc. Worshop on Visual Behaviors*, Seattle, Washington, pp. 91–96.

Singh, A. and Shneier, M. (1990). Grey level corner detection: A generalized and a robust real time implementation, *Computer Vision, Graphics, and Image Processing* **51**(1): 54–69.

Spät, H. (1980). *Clustering Analysis Algorithms for Data Reduction and Classification of Objects*, Ellis Horwood Limited, Chichester, U.K.

Tolhurst, D. (1973). Separate channels for the analysis of the shape and the movement of a moving visual stimulus, *J. of Physiology* **231**: 385–402.

Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography: A factorization method, *Int. J. of Computer Vision* **9**(2): 137–154.

Toyama, K. and Hager, G. D. (1996). Incremental focus of attention for robust visual tracking, *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, 1996*, IEEE Computer Society Press, San Francisco, California, pp. 189–195.

Treisman, A. (1985). Preattentive processing in vision, *Computer Vision, Graphics, and Image Processing* **31**(2): 156–177.

Tsotsos, J. K., Culhane, S. M., Wai, W. Y. K., Lai, Y., Davis, N. and Nuflo, F. (1995). Modeling visual attention via selective tuning, *Artificial Intelligence* **78**(1–2): 507–545.

Uhlin, T. (1996). *Fixation and Seeing Systems*, Ph. D. dissertation, Dept. of Numerical Analysis and Computing Science, KTH, Stockholm, Sweden.

Uhlin, T., Nordlund, P., Maki, A. and Eklundh, J.-O. (1995a). Towards an active visual observer, *Proc. 5th International Conference on Computer Vision*, Cambridge, MA, pp. 679–686.

Uhlin, T., Nordlund, P., Maki, A. and Eklundh, J.-O. (1995b). Towards an active visual observer, *Technical Report ISRN KTH/NA/P--95/08--SE*, Dept. of Numerical Analysis and Computing Science, KTH, Stockholm, Sweden. Shortened version in Proc. 5th International Conference on Computer Vision pp 679–686.

Ullman, S. (1996). *High-level Vision*, MIT Press, Cambridge, Massachusetts.

Wang, H. and Brady, M. (1995). Real-time corner detection algorithm for motion estimation, *Image and Vision Computing* **13**(9): 695–703.

Wang, J. Y. A. and Adelson, E. H. (1994). Spatio-temporal segmentation of video data, *Proc. SPIE: Image and Video Processing II*, Vol. 2182, San Jose.

Wang, J. Y. A., Adelson, E. H. and Desai, U. (1994). Applying mid-level vision techniques for video data compression and manipulation, *Proc. of the SPIE: Digital Video Compression on Personal Computers: Algorithms and Technologies*, Vol. 2187, San Jose.

Westelius, C.-J. (1995). *Focus of Attention and Gaze Control for Robot Vision*, Ph. D. dissertation, Dept. of Electrical Engineering, Linköping University, Sweden.

Witkin, A. P. and Tenenbaum, J. M. (1986). On perceptual organization, *in* A. P. Pentland (ed.), *From Pixels to Predicates: Recent Advances in Computational and Robot Vision*, Ablex, Norwood, NJ, chapter 7, pp. 149–169.

Wolfe, J. M. and Cave, K. R. (1990). Deploying visual attention: The guided search model, *in* A. Blake and T. Troscianko (eds), *AI and the Eye*, John Wiley & Sons Ltd., chapter 4, pp. 79–103.

Yakimovsky, Y. (1974). On the recognition of complex structures: Computer software using artificial intelligence applied to pattern recognition, *Proc. 2nd International Joint Conference on Pattern Recognition*, Copenhagen, Denmark, pp. 345–353.

Yantis, S. and Jonides, J. (1984). Abrupt visual onset and selective attention: Evidence from visual search, *J. of Exp. Psychology: Human Perception and Performance* **10**: 601–621.

Zeki, S. A. (1993). *Vision of The Brain*, Blackwell, Oxford.

Zhong, Y., Jain, A. K. and Dubuisson-Jolly, M. P. (1998). Object tracking using deformable templates, *Proc. 6th International Conference on Computer Vision*, Bombay, India, pp. 440–445.