Control of Perception in Dynamic Vision

Henrik I. Christensen, Claus S. Andersen & Erik Granum

Laboratory of Image Analysis, Institute of Electronic Systems,
Aalborg University, Fr. Bajers Vej 7, Bldg D1,
DK-9220 Aalborg Ø, Denmark

Index terms: Computer Vision, Robotics, Artificial Intelligence.

ABSTRACT

In order to achieve continuous operation and thus facilitate use of vision in a dynamic scenario it is necessary to introduce a purpose for the visual processing in order to provide information that may control the visual processing and thus limit the amount of resources needed to obtain the required results. A proposed architecture for vision systems is presented together with an architecture for visual modules. This architecture enables both goal and data driven processing with a potentially changing balance between the two modes. To illustrate the potential of the proposed architecture an example system for recovery of scene depth is presented together with experimental results which demonstrates a scalable performance.

1. INTRODUCTION

In order to use vision for monitoring of the real world it is necessary for the sensor to be able to operate continuously and thus in real time. Real-time is here taken to imply response within a fixed predefined time interval. Real time may thus be processing at a speed which is below video rate, and it will typically be defined in consequence of the temporal characteristics of phenomena to be monitored.
In order to direct provessing and achieve continuous operation it has been suggested[1,2,4] that processing should be goal directed and the problem of vision should not be studied in isolation but rather in the context of a user (which has goals). Even with goals available most of the algorithms and techniques available today does not operate with an upper bounded time complexity unless the input to the system is constrained somehow. Active control of the resources, in reaction to goals, in order to ensure continuous operation is often referred to as *control of perception*.
In control of perception several kinds of resources may be controlled. I.e.:

- The sensor system
- *Where* and *What* to process? (i.e. use of regions of interest)
- Storage resources

In particular in model based processing where the items in the image/scene are compared to a number of predefined models the search for potential matches is typically NP-complete and to ensure completion within a fixed time interval one must limit the size of such models to a minimum. I.e., anything not directly related to the task at hand (a perceptual goal) should be thrown away, or at least not included in the search. When a system operates continuously the potential amount of information which may be extracted from each of the images is enormous and it is thus necessary to have facilities for reduction of this information and strategies for 'intelligent' forgetting must be developed.
In order to facilitate processing with an upper bounded time complexity it is also possible to select the data which will be processed, i.e., one may select a region of interest for processing. I.e., based on contextual knowledge and present goals one may predict where features may be expected or one may determine an optimal viewing angle for analysis of a particular object. The selection of regions of interests provides a convenient mechanism for focusing of attention.
The problem of vision is highly complex and the desire to achieve continuous operation to facilitate practical use of computer vision in a variety of domains has a long history. It has now been recognized for more that a decade that control is needed in order to enable real world use of computer vision in a large

range of domains[2]. There has recently been an increasing interest in control and system integration as the research is moving from studies of individual techniques towards construction of system. Aloimonos[1] and Ballard[4] have recently proposed a task driven approach to control where only the most critical features for a given task are extracted and used for satisfaction of goals. The proposed approaches are non-modular and to a certain extend an opportunistic approach to control. Crowley et al.[8] have proposed a long term plan for reconsideration of the vision problem in the context of continuous operation and goal directed processing. Preliminary studies have resulted in a suggested system architecture[5] and a centralized controller for decomposition of goal requests from a user[6]. A fundamental hypothesis in much of recent research is active use of the sensor system. I.e., it has been conjectured that many vision problems may be simplified substantially if they are studied in the context of an active sensor system where both intrinsic and extrinsic parameters may be changed. Preliminary results such as those reported by Clark & Ferrier[7] and Rimey & Brown[9] are very promising and indicative that a study of control of perception must include considerations related to the sensor system.

In this paper a general framework for interaction between visual modules is proposed is section 2. The proposed architecture for system control contains facilities for several different strategies to control as it is considered essential that several competing approach may co-exist in a single system. In section 3 a general architecture for continuously operating visual modules is developed and it is described how temporal context may be utilized for local interaction between modules to ensure maintenance of the available representations. In section 4 it is described how the system and module architectures proposed may be used in a system for robust extraction of sparse depth cues. For each of the system modules the implications of the system architecture and control it is described. In section 5 the possible use of a controllable sensor system is briefly outlined. The described system has been implemented in experimental software and tested on both synthetic and natural images. Experimental results which demonstrates a scalable performance and techniques for detection of unexpected scene events are reported in section 6. Finally section 8 provides a summary and an outline of some issues for future research.

2. A SYSTEM APPROACH TO CONTROL

In a goal directed approach to vision several different strategies to control may be used. Control may be divided into two main categories[5]:

- Hierarchical
- Heterarchical

In a hierarchical approach the modules are ranked according to abstraction or responsibility and the flow of information (data and control) is *top-down* and/or *bottom-up* . In a typical scenario a specified goal provides a focus for the upper most module (i.e., interpretation) which in turn specifies what information the module must acquire from "lower" level modules. In consequence of "local" goal specifications a module provides (to the extend possible) the desired data or a specification that goal satisfaction is impossible (possibly with a confidence factor associated with the response). A typical architecture for hierarchical control is shown in figure 1.

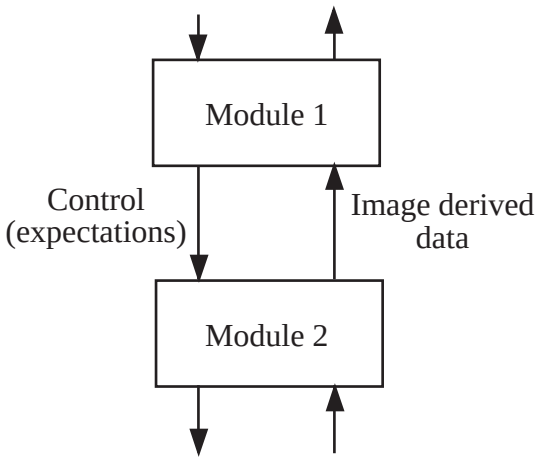

Figure 1: System architecture for hierarchical control

It is important to note that data provided by a module may arise due to two phenomena:

1. verification/rejection of goal requests
2. unexpected / unanticipated data (scene events)

The data in category one has already been outlined, but for control it is equally important to consider those in category two. Scene/image events provides important cues for bootstrapping and model invocation, and in a continuously operating system it provides also cues for change of processing modes; i.e., temporal events may be related to initiation/termination of motion (at a given scale) or a change in motion pattern (constant motion $\leftrightarrow$ accelerating, collision, ...). Such events may require a change of higher level goals; i.e., from a "description"

of an item it may be necessary to direct processing towards "tracking".
In the heterarchical control strategy there is no strict ordering of modules. In such a system the use of both data and control information is opportunistic. An example of systems which exploits such a strategy is the well known blackboard systems used in AI. In this strategy one of two approaches to control of individual modules may be adopted.

1. explicit requests to modules
2. posting of requests

In the explicit approach the modules have well defined strategies which specifies the primitives/data needed for satisfaction of individual goal requests and a specification of the modules to activate to obtain such data. The alternative approach uses a blackboard or posting mechanism for announcement of goal/subgoal requests.
Each module or group of modules has access to these requests and they have internal facilities for evaluation of their ability to satisfy such requests. The opportunism, mentioned earlier, is thus related to the self activation of modules in consequence of announced requests. In such a strategy a module might respond to a request with a new request. I.e., a tracking module might provide information about the focus of expansion given that a sequence of images or tokens is provided by another module.
The activation of modules can also be data driven so that modules self activate when/if the needed data for processing are available. Such an activation strategy facilitates event driven or bottom-up driven processing. A typical architecture for heterarchical control is shown in figure 2.
Goal requests may have many different formulations, but they may typically be seen as expectations. I.e., what is the point in a request like "where is the box?" unless there is an expectation that a box can be found in the scene.
In a temporal context (given continuous operation) where the scale of analysis is sufficient most of the "features" in the image/scene will be well behaved and obey some kind of path coherence which allow prediction of their location/parameters in subsequent images. Such predictions may be thought of as expectations which specifies what and where to 'look' (process). In robust estimation of features or derived parameters this mechanism facilitates accumulation of evidence from multiple images or views. In event detection the expectations provides a context for ignorance. I.e., what features or behaviors are the system/module already aware of.

The use of expectation is particularly suited in a hierarchical control strategy where a single sequential set of modules cooperate to maintain a description of the scene to facilitate goal satisfaction and continuous operation. The local interaction between modules provides an efficient tool for small scale changes in parameters and regions of interest etc.

For requests related to items not previously seen the distribution and nature of the items determines what an optimal strategy to control is.

Generic objects (i.e. object classes) can typically be described in a feature hierarchy where the object is decomposed into parts. I.e. a cup is composed of a body (cylinder with container functionality) and a handle (generalized cylinder with grabable functionality), each of these items may be broken down into other primitives etc. Given such a break down each module may have strategies for transformation of input parts (data made available to the module) into output parts/features (data provided to other modules). There is little point in centralizing such knowledge and given distributed strategies the hierarchical approach to control seems a good choice. For goals related to specific objects (or instances) where unique features can characterize the object (or a temporal phenomena) it is more efficient to exploit a heterarchical approach. I.e., if it is known that a particular object is the only yellow item which can occur in the scene, a "simple" color segmentation module may be invoked for goal satisfaction. Control should here perform a direct invocation of the color segmentation module (with a specification of any contextual knowledge which may guide the processing) as a top-down invocation may be slow and inappropriate. For system level control both kinds of control are thus desirable and a system should preferably have facilities for use of both kinds of control.

In the following section a standard module architecture for modules which facilitates both strategies will be presented.

3. CONTROL OF VISUAL MODULES

In continuous processing of input data, where some kind of continuity for the motion in a dynamic scene may be assumed, it is convenient to exploit the temporal context for reduction of ambiguity. I.e., from tracking it is well known that maintenance of labels / matching may be achieved through adaptation of a hypothesize – match – update cycle of processing. In such an approach the temporal context is used for prediction of the appearance of features in the subsequent sample (image), and the matching is then

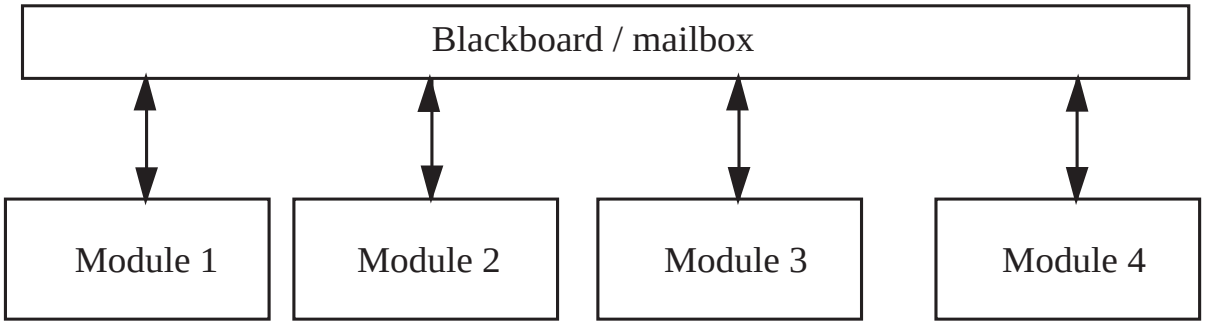

Figure 2: System architecture for heterarchical control

performed between the predicted and the image derived features. The module architecture is shown in figure 3. This approach has proven to be highly robust and to simplify matching substantially. In order to make such a module operate in a goal directed context it is necessary to extend the module architecture so that processing may be guided.

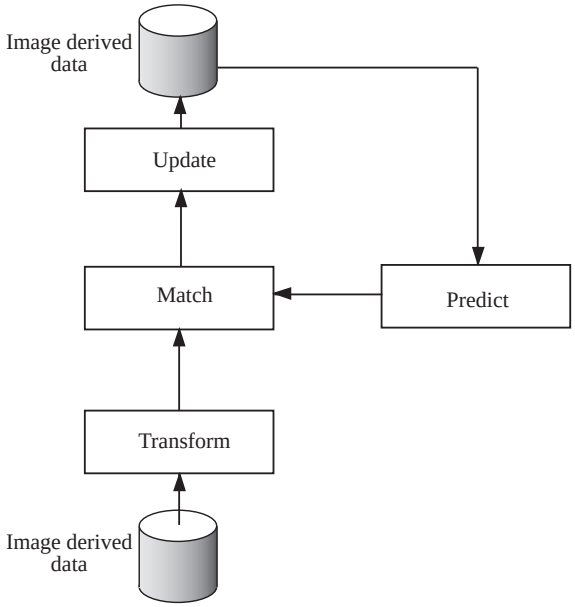

Figure 3: An architecture for predict–match–update

The use of top-down expectations may be incorporated through introduction of an additional model (expectation model) in the module. The prediction can then take both the expectation and data derived models into account when the features in the next image are estimated. I.e., the prediction will estimate the occurrence of features which has been seen earlier and features which are expected by other modules. Through change of the weight associated with the primitives in each of the two models it is possible to change the balance between top-down (goal-directed) and bottom-up (data/event-driven) processing.

The prediction of new primitives may be used mot only in matching but the primitives can also be converted into the vocabulary of module input primitives or "lower" level primitives. Such converted primitives may be used for control of other modules as they provide an expectation which specifies primitives that should be present in order for this module to generate needed output primitives. Through introduction of the conversion function (inverse transform) a control loop between neighboring modules has been closed and a convenient mechanism for hierarchical control has been provided. The modified module architecture is shown in figure 4.

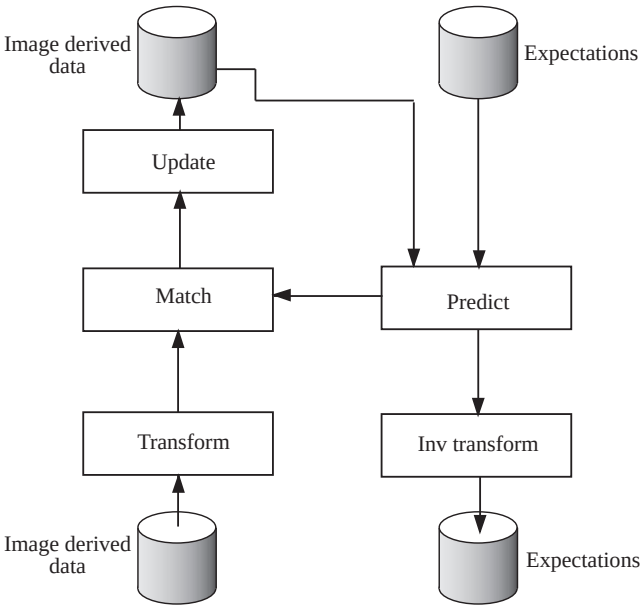

Figure 4: A module architecture which facilitates use of hierarchical control

For the heterarchical control each module is equipped

with a local controller which may perform one of four different operations.

1. Determine if a given goal request received from a common system control channel (i.e., a blackboard) is relevant to the module and if so introduce primitives which enables goal verification / rejection.
2. Determine if the data available at the input may be fused (transformed) into a sensible set of features which can be forwarded to "higher" level modules.
3. Control the balance between goal-directed and data driven processing based on information related to quality of available primitives, the set of present goals, and the available resources.
4. Determine local status information and communicate changes to other modules or a centralized controller. Based on the ratio

$$\frac{\#new\ primitives\ seen}{\#items\ in\ local\ model} \quad (1)$$

and

$$\frac{\#primitives\ lost}{\#items\ in\ local\ model} \quad (2)$$

it may be determined if the module is operating in a stationary/stable mode or an event detection mode (potentially for each region of interest). If a large number of new primitives are introduced and other set of primitives are lost between each image there is excessive noise in the data or the tracking processing is failing due to non-optimal parameter settings. If a large number of primitives are being introduced it suggests an event which has lead to introduction of new structure within the field of view, while loss of primitives suggests removal of structure.

The total module architecture, which includes the structure shown in figure 4 and a local controller, has thus facilities for both hierarchical and heterarchical control. Given an "intelligent" local controller it is possible to dynamically shift the balance between the two modes of control.
In the next section an example of the use of such a module architecture is outlined.

4. AN EXAMPLE SYSTEM

To demonstrate some of the principles described in the previous sections a vision system is presently under construction at Laboratory of Image Analysis, Aalborg University. For recovery of sparse depth cues an iterative binocular feature based stereo approach is used. The images from each of the cameras are subjected to the following processing steps:

a) Image acquisition

b) Edge detection

c) Line extraction

d) 2–D tracking

e) 3–D recovery and tracking.

The 2-D lines which are maintained by the tracking process are then fused in a matching process which provides a set of 3-D lines (or rather a disparity description along 2-D lines). The 3-D lines are subsequently tracked in 3-D. The architecture for the recovery is shown in figure 5.

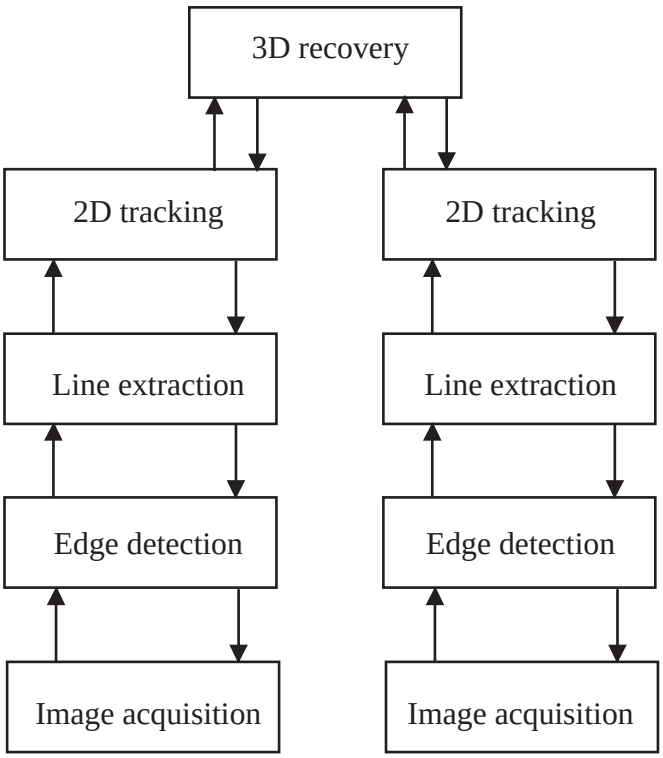

Figure 5: Architecture for recovery of sparse depth cues

Each of the modules are interconnected according to a hierarchical strategy. The implications in terms of control are outlined below for each of the modules.

3-D Recovery and Tracking The 3D recovery is based on a "simple" matching which utilizes minimum distance between measured primitives.

The matching allows one $\leftrightarrow$ many matching to occur through integration of multiple observation by averaging. The extracted 3-D lines are subsequently tracked using Kalman filtering.

The 3-D tracking enables prediction of line position and orientation in 3-D. Given knowledge about intrinsic camera parameters (obtained through through calibration) it is possible to backproject the lines into 2-D (in a local coordinate system for each of the cameras). The back projected lines may subsequently guide the 2-D tracking process.

As the tracking processes in both 2-D and 3-D are based on Kalman updating each line has an associated covariance matrix for the estimated parameters (position, orientation, and velocity). The covariance is large for new or changing parameters and small for stationary parameters. This corresponds directly to the size of the search regions which should be used for matching with lines in new images. The covariance information is thus made available as part of the back projection.

2-D Tracking In the 2-D tracking both projected and data derived lines are used in the prediction and matching. There is presently no balancing between the two kinds of primitives and they are associated equal weight. The 3-D lines have associated id's for the 2-D lines which were matched as part of the 3-D recovery. These id's are used for simple matching of projected and data derived lines, and such lines have a higher confidence than those with no 3-D counterpart. I.e., the lines with support in both models have both temporal and spatial contextual support and they are thus considered reliable.

The predicted 2-D lines are also back-projected to lower level modules as search regions for the analysis of single images. These search regions have associated size information which is directly proportional to the covariance related to position of line mid-points.

2D Line Extraction The extraction of straight line segments may be based on a variety of different techniques. We have here chosen to use a Hough based technique which has been optimized for detection of finite length line segments.

The Hough detection algorithm performs a scan through the image to construct values in the accumulator space which subsequently is segmented to provide a linked list of lines. The actual scanning of the image is by far the step which uses the least resources and it has consequently not been optimized for use of search regions. Anything received from the edge detection module is simply processed. The Hough module does, however, relay the expectation model downwards to lower level modules.

Edge Detection In order to optimize line extraction it is desirable to have an edge detection process which calculates local gradients and orientation. This is here achieved through use of a 7×7 Prewitt operator, where the gradients initially are stored in a polar representation (magnitude and orientation). The edge image is subsequently processed by a hysteresis based segmentation and thinning operator which provides edge segments of width equal to one. The thinned image is now converted into a single image where pixel with a value different from zero are coded with edge orientation for use in the Hough algorithm.

In this module the input image is only processed within the specified search regions (i.e., processing is driven by top-down expectations and it does not have facilities for detection of events, a facility which will be added later). All the pixels outside search regions are assigned a value of zero. In order to facilitate simple use of the search regions the expectation model form the basis for construction of a mask image where the search regions are projected onto and subsequently filled. When the edge detection and thinning is performed the mask image is checked and only at the positions where the mask image has a non-zero value is the operator applied. The width of the search regions is equal to $s_\perp = k * \sigma_\perp$ where $\sigma_\perp$ is the variance perpendicular to the line, while k is a control parameter that enables scaling of search regions, a similar formula is used for calculation of the length of the search region (i.e., $s_\parallel = k * \sigma_\parallel$).

The adaptation of control information is expected to facilitate speed-up of processing and demonstration of a scalable performance for most of the modules. The actual results obtained with the system is described in section 6.

5. USE OF SENSOR SYSTEM

At Aalborg University a binocular robot camera head has been build. The head includes facilities for control of focus, zoom, aperture, vergence angle, and baselength (distance between optical centers).

In stereo matching it is well known that matching is more difficult when the baselength is large. In order to reduce this problem, the described system uses an iterative method, where the initial baselength is small ($d < 10cm$), and once an initial correspondence (with a high uncertainty on the depth estimates) has been obtained the baselength is gradually increased improving the disparity estimates while maintaining (reliable) correspondence. The strategy for change of baselength is based on the uncertainty associated with the extracted 3-D line segments. The covariance for all the line segments are averaged and the resulting *average covariance* is used for control of baselength. Initially the number of 3-D line segments determines when the control algorithm is activated. I.e., a certain number of lines must be present before the improvement algorithm is initiated. The baselength is then increased until a pre-specified value of the covariance is achieved.

The camera head can also controlled by other modules in the system; i.e., geometric scene modeling, but that in another issue which is still under investigation.

6. EXPERIMENTAL RESULTS

The system presented in section 4 has been tested on a number of synthetic and natural image sequences. For the natural images a polyhedral world is used. Most of the test images has been acquired for a model of a small town build from wooden blocks. An example images from one of our towns is shown in figure 6.

The images acquired was processed by the system presented earlier and an example image which shows the search regions for the image shown in figure 6 is illustrated below in figure 7.

In a sequence of experiments we have tried to vary the control parameters (k) associated with the search regions in order to demonstrate that these parameters allow control of use of resources. The test reported here are based on the sequence from which the image in figure 6 was taken. In an experiment where no control is imposed the system extracts on the average $n_{2\perp D} = 40$ 2-D lines which results in an average of $n_{3\perp D} = 38$ 3-D lines.

In the experiments the control parameters k was varied from $k = 0.25$ to $k = 1.5$. The results obtained are shown below in table 1.

Figure 6: Example of natural image of one of the model towns used

Setup	$\overline{n_{3\perp D}}$	relative exec. time
No control	38	1.00
k=0.25	38	0.25
k=0.50	33	0.35
k=0.75	33	0.43
k=1.00	33	0.49
k=1.50	33	0.56

Table 1: Results obtained when the control parameters are changed.

The reported timing results does not include the time needed for construction of the mask image in edge extraction as this time is highly dependent on the efficiency of the used filling algorithm. It should, however, be noted that our filling algorithm is fairly slow (on the order of 2 sec. for a 512×512 image) and we need thus a small control parameter to obtain good results, but our filling algorithm has not been optimized in any way. A more elaborate study of filling algorithms may potentially give much better results.

From the results shown in tabel 1 it is evident that the execution time may be controlled through variation of the control parameters k. It is also evident that the results obtained with feedback from other modules are more robust.

7. SUMMARY

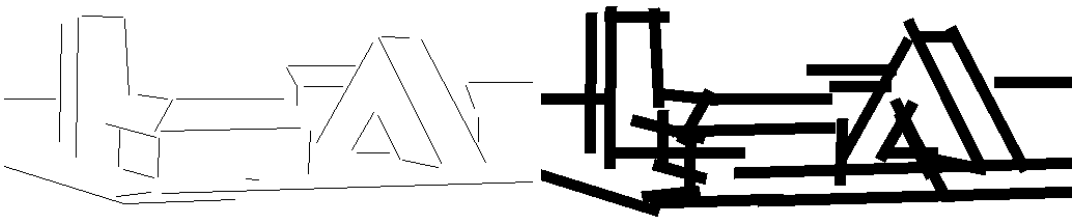

Figure 7: Lines and corresponding search regions for processing of the image shown in figure 6.

It has been argued that vision processing must be goal directed in order to enable continuous operation, and an architecture which facilitates both hierarchical and heterarchical propagation of control information has been presented. A module architecture which is suited for incorporation into this system structure was also presented. The proposed architectures was demonstrated on an example system for recovery of scene depth, and experimental results demonstrated that the system has a scaleable performance.

Future research will be aimed at introduction of similar structures at other levels in a vision system in order to demonstrate scaleable performance for a fully integrated vision system.

8. ACKNOWLEDGMENTS

The work presented in this manuscript has benefitted substantially from discussions with our partners in ESPRIT Basic Research Action BR3038 "Vision as Process" and our colleagues at Laboratory of Image Analysis, Aalborg University in particular the members of the VINE group.

For the work reported here Henrik I Christensen was funded by ESPRIT BR3038 "Vision as Process" while Claus S. Andersen was funded by the Danish Technical Research Council under the MOBS framework programme and the Faculty of Science and Technology at Aalborg University. This funding is gratefully acknowledged.

9. REFERENCES

1. J.Y. Aloimonos, Purposive and Qualitative Active Vision, Proc. DARPA IUW-90, pp. 816–828, Morgan Kauffmann, Los Angeles, CA., 1990.

2. R. Bajcsy & D. Rosenthal, "Visual and Conceptual Focus of Attention", in Structured Computer Vision, (Eds.) S. Tanimoto & A. Klinger, pp. 133 – 150, Academic Press, New York, NY., 1980.

3. R. Bajcsy, "Active Perception", IEEE Proceedings special Issue on Computer Vision, Vol. 76, No. 8, pp. 996 – 1005, August 1988.

4. D. Ballard, "Animate Vision", Artificial Intelligence, Vol. 48, No. 1, pp. 57 – 86, February 1991.

5. H.I. Christensen & E. Granum, "Specification of Skeleton Control Structure", VAP report DR.E.1.2, Aalborg University, Aalborg, June 1990.

6. H.I. Christensen, "Process Supervisor for Skeleton System", VAP report DR.E.2.3, Aalborg University, Aalborg, August 1991.

7. J.J. Clark & N.J. Ferrier, "Modal Control of an Attentive Vision System", Proc. 2nd Intl. Conf. on Computer Vision, (Eds.) R. Bajcsy & S. Ullman, pp. 514 – 523, IEEE Press, Tarpon Springs, Fla., 1988.

8. J. Crowley, A. Chehikian, J.Kittler, J.Illingworth, J.O.

Eklundh, G. Granlund, J. Wiklund, E. Granum, & H.I. Christensen, Vision as Process, ESPRIT Basic Research Proposal, Aalborg, 1988.

9. R. Rimey & C.M. Brown, "Controlling Eye Movements with Hidden Markov Models", Intl. Journal on Computer Vision, Submitted 1990.