In Linde, P and Sparr, G, *Symposium on image analysis,* Lund University, 1996.

The XT-1 Vision Architecture

Christian Balkenius
Lars Kopp

Lund University Cognitive Science
Kungshuset, Lundagård
222 22 Lund, Sweden

christian.balkenius@fil.lu.se
lars.kopp@fil.lu.se

1 Introduction

A mobile robot navigating in an unstructured environment faces many difficult problems for which vision may potentially offer useful solutions. The XT-1 (eXpectation based Template matching) architecture was developed in an attempt to address many of these problems with similar constructions. The current system handles such diverse problems as landmark and place recognition, the generation of orienting and anticipatory saccades, smooth pursuit, as well as visual servoing during locomotion. Although all these tasks are highly interwoven, they can roughly be divided into subsystems for **navigation** and **target tracking**. The tracking system has been successfully implemented in a robot (figure 1). We are currently moving the navigational system from an experimental set-up to a real mobile robot (figure 2).

The emphasis of the architecture has been on the actual tasks that a mobile robot needs to perform rather than on the more theoretical aspects of computer vision. Although we view such work as important, the ultimate success for computer vision in robotics depends on its ability to generate useful information in a sufficiently short time (and at a sufficiently low cost, Horswill and Yamamoto 1995). In order for a robot to react to unexpected changes in the environment, the through-put of the system must be fairly high. As a result, the quality of the computed values often needs to be reduced. However, a rough localization immediately is usually better than a more exact one a few minutes later. Similarly, for target tracking, the important aspect is to keep the target in view, not to track it optimally.

To accomplish this feat, we have made heavy use of **expectations** which greatly reduces the amount of computations that need to be performed. For example, in the tracking subsystem, expectations of the target position constrain the region of the image that needs to be searched. In the navigational subsystem, place expectations are computed from earlier visual fixes together with path-integration during locomotion (Gallistel, 1990). These place representations constrain which landmarks can be expected and, consequently, which set of features that needs to be searched for in the image. A related aspect of the architecture is its use of a low-level attentional system which selects areas of the image where useful information is likely to be found.

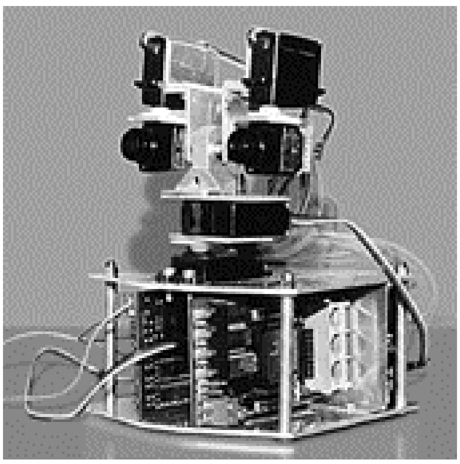

Figure 1. The LUCS Active Stereo Vision Head. The head has four degrees of freedom (pan, tilt, 2×vergence) and is controlled by the vision-architecture described in the text.

A further important feature of the architecture is that it takes biology seriously. An enormous body of data is available about the mammalian visual system and we it would be ignorant not to take it into account. This does not mean that one should necessarily try to model every detail of any real visual system. Many operations do have different computational realizations that are better suited for digital hardware. However, it appears that the overall task of a mobile robot is sufficiently close to that of an animal that the overall visual architectures should be very similar. From biology, we also borrow the idea that the visual system should be judged by how well it suits the need of the robot rather than on any other ground (McFarland, 1993).

2 Overview of the Architecture

The architecture can be divided into five conceptual levels: low-level processing, attentional processing, single feature processing, spatial relations, and place/object-recognition (See figure 3). At each higher level the representations becomes more complex, but the processing is fundamentally heterarchical: the information flow is both bottom-up and top-down, as well as lateral.

The first level is concerned with **low-level preprocessing** of the video-images. A scale-space pyramidal edge-detection constitutes the first stage at this level. In the second stage, the difference between successive edge images is used as a quick-and-dirty motion detection.

The second process level deals with **attentional processing** based on the input from the first level. A primitive attention module directs the attention of the tracking subsystem to sudden motion in the scene and triggers an orienting saccade toward it. When the navigational subsystem is disengaged, this primitive attention system is used to select targets for the tracking system. This module is inhibited while the camera-head is moving. A second parallel system directs attention to potentially good features in the image. These regions of the image are used as candidate landmarks at the higher levels.

Unlike the two previous levels which perform global computations, only local features are processed at the third level. The **single feature processing** is applied to regions of the image that have been selected, either by the attentional systems, or by top-down influences from higher levels. The feature-correlator is the central component of this level and is used both to compute optic flow and to locate landmark and target features in the image. A search-field module is used to control where in the image it is fruitful to compute local feature correlation. This module reduces the amount if computation required by the system.

At the fourth level, the **spatial relations** between individual features are used to represent landmarks in the navigational subsystem. Such collections of features can also be sent to the tracking system when the robot needs to pursue a goal. When the tracking system acts on its own, the optic flow calculated at the lower level controls a segmentation process where a region of homogeneous motion is selected as target.

Finally, at the fifth level, the angular relations between landmarks come together to form the representation of **places**. Such relations can be seen as second order-spatial relation, i. e., relations between collections of features which themselves are grouped with their spatial relations. Note that using this scheme, no object recognition or complicated segmentation is necessary to categorize a place. To a first approximation, it appears that also object recognition is a process at this level.

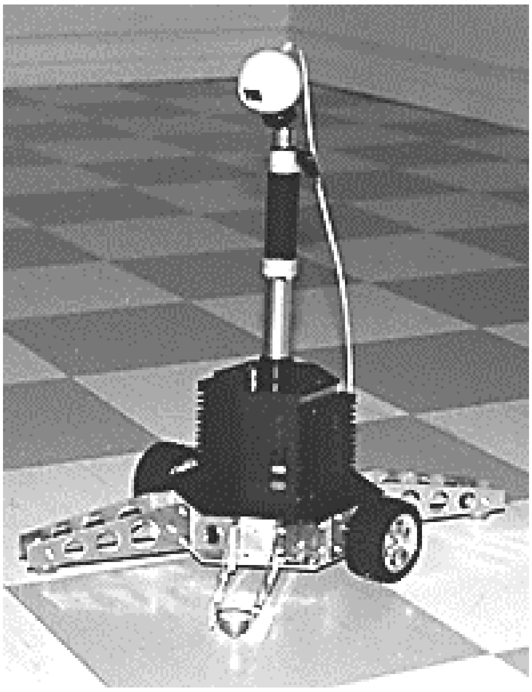

Figure 2. The mobile robot that will navigate using visual landmark recognition together with dead-reckoning.

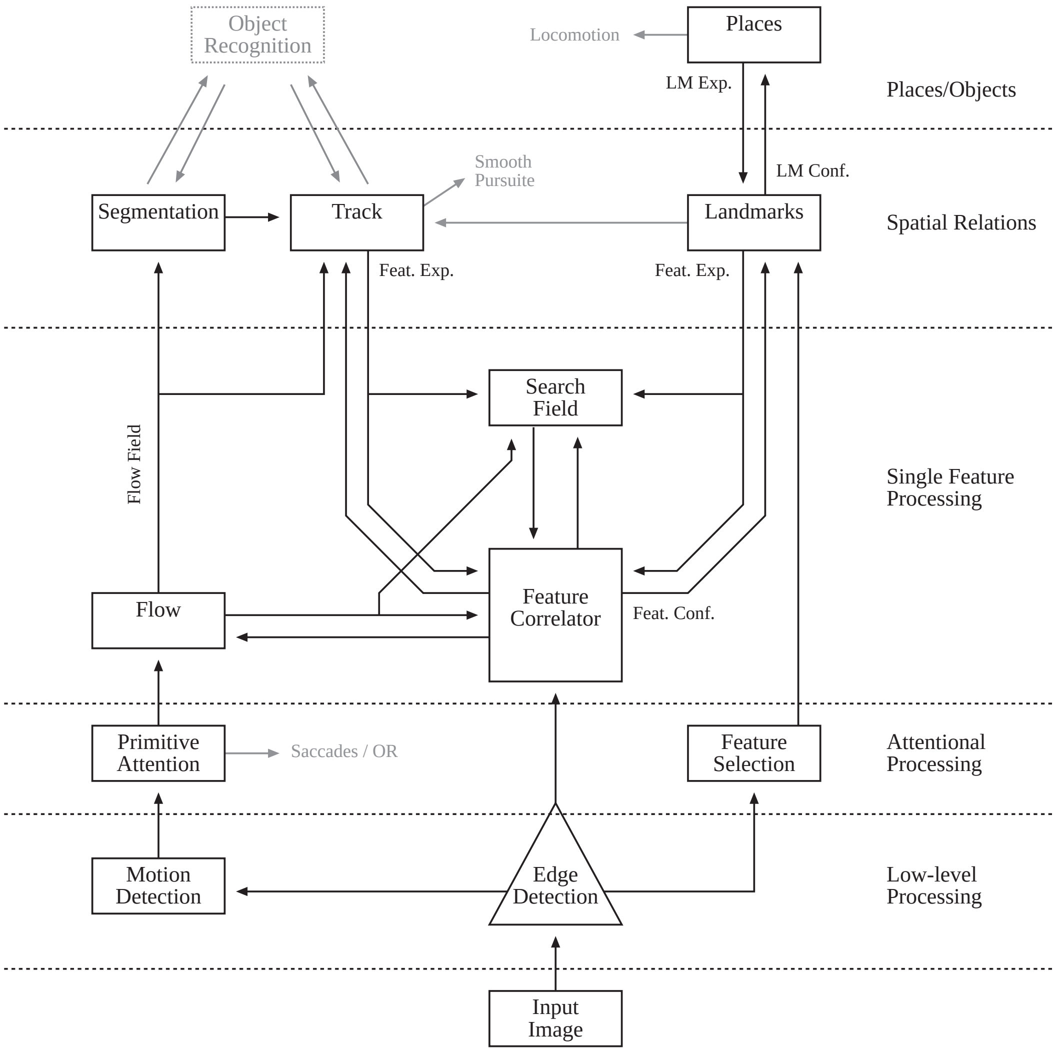

Figure 3. Overview of the XT-1 vision architecture.

3 The Use of Expectations

It is a formidable task to build a 3-dimensional model of an unknown object from a visual observation in a natural environment. No algorithm exists today that can use a bottom-up approach to do this for an arbitrary scene. It is much easier to compute the position and orientation of a specific shape in an image using a top-down approach and most applications of computer vision choose this latter method instead. However, with a top-down approach comes the problem of selecting the appropriate models to test against the environment.

The remedy to this problem is, of course, to already know which model to apply. In this case, only a single model needs to be matched against the image. While this may appear to beg the question, we want to argue that, at least for a mobile robot, this is true most of the time. For example, once the robot has managed to determine its position the first time, it can use dead-reckoning to update its position and orientation. As a result, it will always have a fairly accurate estimate of its location even before the visual system is engaged. This, in turn, implies that it can select the correct landmarks to look for in the scene most of the time. There exists much evidence for the use of such mechanisms in animal navigation (Gallistel, 1990).

Our approach can be called **expectation-based vision** since it emphasizes the top-down influences on visual processes. It differs from other model-based processing in one important aspect: it does not put any formal requirements on the top-level representations except that they should facilitate the processing at lower levels. The expectations we use are in the form of **elastic templates**, that is, collections of features together with their approximate spatial relations.

In the XT-1 architecture, such expectations are used to control almost all processing. The expected place is used to select the landmarks that will be visible from the current location. These, in turn, select the features to be searched for in the image. For example, if you know you are in the kitchen, there is no need to apply templates for showers and beds (cf. Rumelhart et al., 1986). This does not mean that one is unable to recognize a shower in the kitchen, but it requires that the system gives up the hypothesis of being in a normal kitchen.

Expectations are also used in other ways, for example to generate anticipatory saccades toward landmarks that are not currently in view, and outside the visual system for spatial navigation (Balkenius, 1995a). Below, we describe in more detail how the idea of expectations are implemented in the two major subsystems.

4 Target Tracking

One of the most significant visual processing stages in the complex vision machinery is the object tracking module. Since in any realistic setting, the visual system will continuously be in motion, it must be able to track any object of interest before it can be categorized or recognized. With no such module, the visual processes would not be able to account for complex visual tasks such as: smooth pursuit, landmark and object recognition.

The tracking module in the XT-1 architecture is based on the search-light metaphor (Crick, 1984), since it puts the selected object in focus of attention. Let us consider the interactions between modules participating in the tracking process. The modules that are involved are: **primitive attention**, **optic-flow**, **segmentation** from motion, the **feature correlator**, the **search field** and the **tracking** module (See figure 3).

First, the optic-flow is calculated only in image-areas where something is in motion. This results from the optic-flow computations being data-driven by inputs from a low-level primitive attention system. The optic-flow computations are based on a correlation method where features are correlated in a restricted search area, the search field, between two successive images. The search field is intimately connected with expectations since expectations of the target location and movement govern the shape of the search field. For example, when an object moves fast in a certain direction, the search field enlarges in that direction. This is an adaptive regulation which makes it possible for the tracking process to follow fast moving objects.

Second, in the segmentation module, a winner-take-all principle decides which local motion-direction is selected as a target. In the next stage optic-flow is used to make figure-from-ground separation. This process integrates optic-flow information and categorize regions of the image with coherent motion. To do this, a neural network classifies motion-directions into eight categories. Neighbouring regions with the same direction preference are evaluated as a group and the largest group is selected as the object of interest.

Finally, the tracking module computes the target position and controls the movement of the camera head. Another function of the tracking module is to select features that could belong to the current target and handle them to the feature correlator. In this way, expectations of the target are used in a top-down fashion to control the recognition process.

5 Navigation

The central task of the navigational system is to recognize places. Like the tracking subsystem, place recognition is performed on a number of levels. The modules used are: **feature selection**, **feature correlator**, the **search field**, a **landmark** module and a **place** module. The search field and the feature correlator are shared with the tracking subsystem. The other modules are specific to the navigational system.

The feature selection unit picks out features where high contrasts or edges are salient. A stochastic process chooses between features to pick out the sixteen most suitable features. The selected features are glued together by their spatial relations which are learned by the landmark module. A landmark is represented as a set of features at both a fine and a coarse scale, together with the spatial relations among them. The fine scale is a 256×256 pixels edge-filtered image, and the coarse scale is a 32×32 pixels edge-filtered image. The two different scales emphasize different aspects of the image. We have found that different environments put varying demands on such representations. To make the landmark module to work properly, more than one scale is necessary.

The landmark module recognizes landmarks and is used to generate angles to target objects. As input, it takes expected landmarks from the place-module. Since the relations between features have previously been learned, the landmark module can use expectations of where features are to be found to speed up the recognition process. At the start of the recognition process, the search field is enlarged to its maximum, but for each feature that is found, the search field can be shrunk.

In detail, the landmark module consist of a modified ART 2 network (Carpenter and Grossberg, 1987), that learns to recognize features. When the landmark module has to learn new landmarks, sixteen new features are selected that are chosen by the feature selection unit. Since the spatial relations between features are allowed to vary slightly in the recognition phase, landmarks are tolerant to slight variations in size and orientation. Moreover, since we allow some of the features to be missing, the landmarks are stable against partial occlusion.

The place system trades memory for computational power. The heavy use of expectations makes the computational requirements much smaller than for most vision systems. On the other hand, a lot of template data needs to be stored. In the current implementation, the memory requirements are approximately 16kByte/m^2 in a normal office environment. However, hard-disk memory is cheap, processor speed is expensive.

6 Results and Further Research

All the modules in the architecture have been implemented and tested with real video-input. However, all the modules in the architecture have not yet been run simultaneously. Today, both major subsystems operate successfully in real time using fairly modest computational resources.

We believe that the architecture already contains sufficiently many levels for most robot tasks. The further development of the architecture will be toward including more modules at each level rather than extending it upwards. For example, it will be necessary to include modules for obstacle avoidance and stereo processing.

In the future, we will also investigate how vision should interact with motor control in the different behavior systems of a mobile robot (cf. Balkenius 1995a). It will also be necessary to further study how vision should interact with other sensory systems (Balkenius 1995b).

Acknowledgments

This research was supported in part by NUTEK.

References

Balkenius, C. (1995a). *Natural Intelligence in Artifical Creatures*. Lund University Cognitive Studies 37.

Balkenius, C. (1995b). “Multi-modal sensing for robot control”. In Niklasson, L. F., Bodén, M. B. ed. *Current trends in connectionism*. Hillsdale, NJ: Lawrence Erlbaum. 203-216.

Carpenter, G., Grossberg, S. (1987). “ART2: Self-organization of stable category recognition codes for analog input patterns”. *Applied Optics, 26* , 4919-4930.

Crick, F. (1984). “Function of the thalamic reticular complex: the searchlight hypothesis”. *Proceedings of the National Academy of Sciences,* (81), 3088-3092.

Gallistel, C. R. (1990). *The organization of learning*. Cambridge, MA: MIT Press.

Horswill, I., Yamamoto, M. (1995). *A $1000 active stereo vision system*, manuscript, MIT AI Lab.

McFarland, D., Bösser, T. (1993). *Intelligent behavior in animals and robots*. Cambridge, MA: MIT Press.

Rumelhart, D. E., Smolensky, P., McClelland J. L., Hinton G. E. (1986). “Schemata and sequential thought processes in PDP models”. In Rumelhart, D. E., McClelland, J. L. ed. *Parallel distributed processing*. Cambridge, MA: MIT Press. 7-57.