

A PROCEDURAL MODEL OF RECOGNITION

William S. Havens *

Department of Computer Science
University of British Columbia
Vancouver, B.C., Canada

1. Introduction

This work is based on the belief that a computational theory of perception must encompass dual theories of representation and recognition. We propose a procedural model of recognition which 1) provides a mechanism for flexibly integrating top-down and bottom-up control structures, 2) supports the use of a hierarchy of model driven methods, and 3) defines a deductive process scheduling mechanism.

2. The Chicken and Egg Problem

The top-down recognition model proposed by Minsky(1974), championed by Kuipers(1975), and used by many others exhibits a number of short-comings:

1. A frame must explicitly be made the current hypothesis before its expertise is available to the recognition process.
2. An ordering must be placed on alternative subgoals. The top-down paradigm forces the choice of a single hypothesis at a time. The mechanism for choosing a next hypothesis is completely failure driven.
3. Identical subgoals must be carried out independently. The same subgoal may be attempted and achieved numerous times by the system.

To avoid this thrashing behaviour, Kuipers(1975) has advocated the use of a similarity network to recommend a replacement candidate frame on failure. This scheme assumes first that a mapping exists between the failing frame and each next candidate frame and secondly that the network is in some sense "complete". Any "surprises" cause the system to revert to blind automatic backtracking.

These difficulties are three manifestations of what Mackworth(1975) has called "the chicken and egg problem". It seems clear that to avoid these problems, an integration of both top-down and bottom-up techniques are needed. Kaplan(1973) and more recently Bobrow & Winograd(1977) have advocated the use of multiprocessing with priority queue scheduling. We argue that in order to effectively integrate top-down and bottom-up recognition methods more sophisticated mechanisms than priority queues are needed.

3. Implementation

We define a procedural recognition model which utilizes multiprocessing and a deductive scheduling mechanism to realize an integration of top-down and bottom-up search. The model operates in two mutually recursive modes. The first mode is called matching and incorporates essentially the top-down recognition paradigm. The expectations of a frame are matched by direct observation by recursive use of matching i.e. subgoaling, and by recursive calls to the second mode described below.

The second mode implements bottom-up search and consists of two cyclical phases called expectation and completion.

Each frame may have associated with a number of processes called supergoals. Instead of relying entirely on matching to recognize an instance, a frame may suspend its recognition by computing a set of current expectations and suspending supergoal processes to patterns representing those expectations. Each such process remains suspended

until evidence is discovered which satisfies its expectations. The process is then resumed and the recognition of its associated frame continues.

Whereas a subgoal has an explicit caller and must eventually return a success or failure to that caller, a supergoal has no explicit caller. Instead, when a supergoal succeeds it begins the completion phase by resuming in turn one or more suspended supergoals of those higher frames of which its frame may be part. This provides a mechanism for handling non-determinism without resorting to failure driven methods. A particular frame instance need be recognized only once.

Completion also provides a deductive process scheduling mechanism. Processes are activated when the expectations that they are bound to have been satisfied. The model, as well, supports the use of a hierarchy of model driven methods to guide the recognition process. A frame's method computes a set of expectations, binds supergoals to those expectations, and may relinquish control to a higher method. Alternatively, the method may retain control. This permits a frame to pre-empt a higher but less specific method and impose its own more powerful method to encourage the discovery of evidence satisfying its expectations and propagating its own recognition.

This recognition model has been implemented as an experimental programming language, a dialect of LISP called MAYA (Havens,1976). The language provides primitives for constructing and accessing frames and semantic networks. MAYA provides control structure mechanisms for implementing the recognition model described. Multiple processes associated with particular frames may be created, suspended to patterns within frames, and resumed again by matching those patterns. Specific control primitives are defined to permit subgoal and supergoal interaction and to support the realization of method hierarchies. We are presently experimenting with the recognition model by applying MAYA to a computer vision task (Havens,1977).

Bibliography

- Bobrow, D.G. & Winograd, T.(1977) An Overview of KRL: A Knowledge Representation Language, Cognitive Science, vol 1, #1, Jan'77.
- Havens, W.S.(1976) MAYA Language Reference Manual, TM-13, Dept.of Comp.Sci., UBC,Vancouver, Canada.
- Havens, W.S.(1977) A Computational Model of Recognition for Perception, Ph.D. thesis, Dept. of Comp. Science, Univ.of British Columbia, Vancouver Canada, in preparation.
- Kuipers, B.J.(1975) A Frame for Frames: Representing Knowledge for Recognition, in Representation and Understanding, D.G.Bobrow & A.Collins (eds.) Academic Press, New York.
- Minsky, M.(1974) A Framework for Representing Knowledge, AI Memo #306, MIT AI Lab, Cambridge, Mass.
- Mackworth, A.K.(1975) How to See a Simple World, in Machine Intelligence 8, E.W.Elcock & D.Michie (eds.) in press.
- Kaplan, R.M.(1973) A Multiprocessing Approach to Natural Language, Proc. 1973 National Computer Conference, AFIPS Press.

*Author's new address: Department of Computer Science
University of Tennessee, Knoxville, TENN. 37916