A STRUCTURED PATTERN MATCHING APPROACH TO INTERMEDIATE LEVEL VISION

Robert B Fisher

D.A.I. RESEARCH PAPER NO. 177

This paper is being presented as a discussion paper
at the 1982 European Conference on Artificial Intelligence,
Paris, July 12-14, 1982

A STRUCTURED PATTERN MATCHING APPROACH TO INTERMEDIATE LEVEL VISION

Robert B Fisher

Abstract:

This paper presents an approach to extracting the intermediate level information from an image - the description of plausible objects and their structure and location. Our system (IMAGINE) is based on a pattern matcher which operates using a structured vocabulary describing specific objects to be searched for in the scene. The matcher depends upon a set of rules (over the vocabulary) to suggest potential matches, and a set of semantic routines to select the correct ones. The approach is interesting because of its explicit symbolic aspects, its rule based partitioning and its parallelism potentials. An example of the system using a table recognition rule base is given.

Acknowledgements:

The author was appreciatively supported during the course of this work by a postgraduate studentship from the University Of Edinburgh. Special thanks also go to R. Beattie, W. Caplinger, J. Hallam, J.A.M. Howe, and D. Wyse for their assistance with both the content and exposition of this paper.

A Structured Pattern Matching Approach to Intermediate Level Vision

Robert B. Fisher
Department of Artificial Intelligence
University of Edinburgh

Abstract: This paper presents an approach to extracting the intermediate level information from an image - the description of plausible objects and their structure and location. Our system (IMAGINE) is based on a pattern matcher which operates using a structured vocabulary describing specific objects to be searched for in the scene. The matcher depends upon a set of rules (over the vocabulary) to suggest potential matches, and a set of semantic routines to select the correct ones. The approach is interesting because of its explicit symbolic aspects, its rule based partitioning and its parallelism potentials. An example of the system using a table recognition rule base is given.

Keywords: model based object recognition, intermediate level vision, image matching, scene grammars

I. Introduction

This paper presents a vision system which uses some of the semantics of intermediate level scene analysis to describe the objects present in a scene. It starts from a set of primitive descriptions of the low level features (straight edges). Then, using a set of pattern matching rules which describe how these features relate to more complex features, other (higher and lower) descriptions of the image are deduced. The process iterates until all possible matches have been made.

The main body of this paper outlines the definition of a ***vocabulary***, a set of ***matching rules***, and the ***associated semantics*** for recognizing a table in a real scene. (A table is thought to be a typical instance of an object identifiable by this method. It has structural properties that hold over a broad class of actual tables and is generally composed of components with regular geometric structures.)

The question arises as to why bother with another formalism and reasoning structure. We feel that the approach described below has several strong points. The first is that all entities reasoned about are symbolic, discrete and explicitly labelled. Secondly, both the syntactic and semantic elements can be formally described. Thirdly, it places more emphasis upon relational and structural properties, and only uses geometrical reasoning at the lower levels of semantics. Fourthly, though the example uses only bottom up reasoning, top down rules are equally usable. Lastly, the reasoning structure suggests a simple parallel network structure, where each rule is implemented in a separate processor with the matches as the information flow.

II. Background Discussion

We used three levels of vocabulary to represent our intermediate level understanding of an image. The lowest level vocabulary is a set of descriptions for primitive two dimensional image structures. The middle level describes three dimensional scene structures, and the highest level is the commonly used (human) vocabulary for objects and their components. For example, in a blocks world, the three levels would have terms for the structures of line, parallelepiped and arch.

The matching rules are specified in terms of this vocabulary and the associated semantic routines implement the allowable underlying relationships (such as adjacency and shared boundaries). The rules suggest possible matches between structures, and the semantics determine which are valid. The hierarchy arises from the structure of the domain: 2D observables to 3D structures to objects. In a bottom-up recognition scheme, this hierarchy reflects the process of structures hypothesized and verified at one level become the inputs into matches at the next. This form of structuring rests upon the assumption that there exists a partitioning of the object, at various levels of description, into a set of largely independent substructures.

A major assumption of this process is that the interpretation of structures at any level may be ambiguous (though more so at the lower levels). For example, a line segment could arise from different causes, such as object boundaries, shadows or changes in surface reflectance. In the absence of global information (i.e., information from sources distant to the line), it may be hard to interpret the line. So, we take the approach of considering all possible interpretations feasible, implementing alternative interpretations by alternative rules. We then assume that the invalid ones will be either eliminated due to incorrect semantic relationships at the current level, or will not match at higher levels. This depends heavily upon the assumption that the scene is globally consistent, and that this property (at the various levels) will in practice eliminate invalid hypotheses. This means that the matching process should be suggestive enough to propose at least all valid interpretations, and that the semantics should be strong enough to allow only the consistent ones.

Model driven vision has been receiving more recent attention because of the ACRONYM system (Brooks (1981)). This system combines sophisticated models with a constraint-based geometrical reasoner to attempt to recognize objects in three dimensional images. This program (development in progress) has demonstrated some success with largely two dimensional images. This author believes that the initial stages of scene analysis should depend more upon relational and topological reasoning, and only rely as strongly upon geometrical reasoning in the verification stages. ACRONYM's models are represented in a declarative manner; IMAGINE's are partially declarative (the rules) and partially procedural (the semantics). Unless the semantic routines can be

generated automatically from declarative, process-independent models, there will be some long term limitations to our approach.

Freuder's (1977) SEER program is a system which actively reasons about what it knows of the scene, in order to hypothesize structures and where to look for them. Our work is similar, but considers all reasonable hypotheses (conceptually in parallel) rather than having a priority ordered queue.

III. Image Matching

We define a table structurally. Our definition requires a table to have a top and four legs connected at the corners. The top must be rectangular, but may be thin (seen as a parallelogram) or thick (seen as a parallelepiped); in any case, the topmost surface must be seen as a parallelogram. The legs may be thin (seen as lines) or thick (seen, in profile, as pairs of lines). The legs must be attached at the corners of the table top, but only three need to be directly seen to do so. This definition is restrictive about what are acceptable tables, but this is not important - we could broaden this generic class of acceptable tables by adding alternative definitions (implemented as alternative rules), or by making the definition more general. Outside of coincidental alignments, the definitions are designed for recognition of tables from any angle, including upside down. Notice that we combine two elements - a constructive description of how elements relate to each other (i.e., the top and legs), and an analytic description of acceptable shapes for objects (i.e., the top seen as a parallelogram or parallelepiped). The second element also includes our assumptions of viewpoint, geometry, projection, etc.

The most important part of recognizing objects in an image is to define a vocabulary for those objects. For our table example, we define terms such as:

high level: complete table(TB), table top(TT), table leg(TL)

intermediate level: parallelepiped(PL), rectangular parallelepiped(RP)

low level: line segment(LS), quadrilateral(QD)

There are other terms which represent partial structures not commonly given distinct names by people. Altogether, these terms become labels attached to the structures which the program finds in the image, much as a human would use the word "table" to describe a particular table. Associated with each term is a frame-like semantic structure (node) which records the information relevant to a particular instance of the term.

The matching process is driven by a set of rules. These rules designate the syntactic basis for generating new structures from existing components. A rule (usually) has the form "LHS <- RHS1 RHS2", which means: if two nodes of types RHS1 and RHS2 are found, then consider making a node of type LHS. The matching process is driven by these rules: Given a rule, the matcher selects all RHS nodes according to spatial constraints determined

by the structure being searched for (LHS). In essence, it does a spatially-constrained exhaustive bottom-up matching according to the set of rules. It was decided to allow at most two nodes on the right hand side of a rule in order to limit the combinatorial aspects of the matching. This is effective because the rule acts as a filter and only allows a subset of the actual matches to succeed. The output of a match, being another node, may be used in subsequent matches. The rules can specify both bottom-up and top-down reasoning. Four example rules are:

HT <- TT TL (hypothesized table from a table top & table leg)

TT <- RP (table top from rectangular parallelepiped)

PP <- PG PG (partial parallelepiped from two parallelograms)

TL <- HT (hypothesized table suggests table leg)

Associated with each syntactic rule is a set of semantic routines (Thompson & Dostert (1974)). A rule merely specifies the categories of acceptable components, and the potential results, whereas the semantic routines perform the detailed comparisons needed to ensure a valid match. There are four (generally small) routines associated with each rule:

search routine - specifies where to find potential RHS2s to match a given RHS1.

check routine - ensures the potential match is valid by performing specific tests, using the semantic values of the input nodes. For the table model base, the types of semantic checking were for proximity, orientation, similarity and relative sizes of features.

set routine - calculates the semantic values of the output node. (This is invoked when a match passes the check routine.)

subsumption routine - ensures that the newly created node is unique. (This is needed because: 1. alternate derivations of the same construct arise, and 2. real data gives multiple slightly different interpretations.)

The matcher input data base is generated by preprocessing a grey level picture. The output of a edge detection and tracking process (Beattie (1982)) is used to produce the straight line segments (LS nodes). A second routine produces all reasonable quadrilaterals (QD nodes) from the lines. The output of the matching process is a database of nodes, each of which represents one instance of a particular named structure (e.g., TB, TL) found in the image. Associated with each node is a tree of the subcomponent nodes used in its derivation.

IV. Results on a Real Picture and Discussion

In figure 1, we show the raw image. After running the preprocessing programs mentioned above, we had produced 298 straight line segments (596 LS nodes) and 51 quadrilaterals (51 QD nodes) for the initial database. The matcher was then run to completion, looking for all instances of all structures in the image. It had 13 terms in its vocabulary and used a set of 14 pattern matching

rules (and semantics) to reason about them. The number of instances of each structure deduced were:

line segment:	596	table top:	13
quadrilateral:	51	table legs:	110
parallelogram:	11	hyp. table (1 leg):	130
partial p.piped:	16	hyp. table (2 legs):	124
parallelepiped:	2	hyp. table (3 legs):	59
rectangular p.piped:	2	hyp. table (4 legs):	20
		table:	4

We see that there were 4 instances of tables found. These were multiple recognitions of the actual table, and resulted from the combinations of two rectangular parallelepipeds found for the table tops, with two possible legs at one corner. Figure 2 shows the line segments from one of the successful table nodes.

All told, the system (using the table rule set) attempted 97072 matches, of which 2185 were successful and 491 remained after subsumption. It took 5 minutes of running time on a PDP 11/60 to produce this result. This time is excessive, and is largely due to inefficiencies resulting from the 11/60's small address space.

So, this approach seems to be usable. Notice that the table was identified even though edge data was missing, including boundary connections. Again we stress IMAGINE's strong points are its explicit symbolic elements, its granular knowledge/alternative reasoning capabilities, its heterarchical reasoning structure and its potential for parallel implementation.

Countering its strong points are its lack of formal declarative object and image formation models (unlike ACRONYM), the uncertain process of choosing a vocabulary and set of rules for describing objects and the anticipated difficulties with recognizing non-rectangular or non-manmade objects.

This research was supported by a studentship from the University of Edinburgh.

Bibliography

Beattie, R.J. (1982), **Edge Detection for Semantically Based Early Visual Processing**, Proc. 1982 European Conference on Artificial Intelligence

Brooks, R.A., (1981), **Symbolic Reasoning Among 3-D Models and 2-D Images**, Artificial Intelligence, Vol 17, pp285-348

Freuder, E.C., (1977), **A Computer System For Visual Recognition Using Active Knowledge**, Proc. 5th IJCAI, pp671-677

Thompson, F.B., Dostert, B.H., (1974) **Practical Natural Language Processing - The REL System as Prototype**, REL Project Report 13, Computer Science, California Institute of Technology

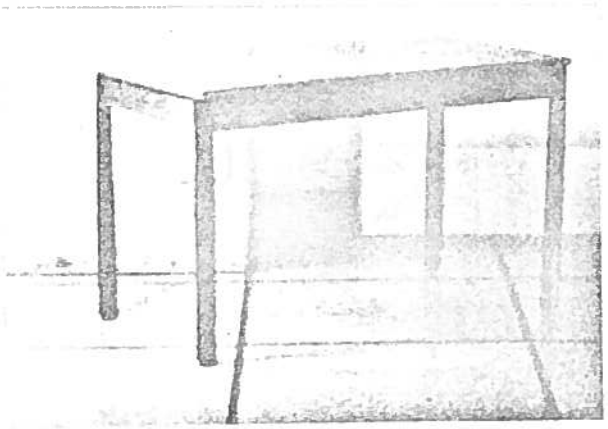

Figure 1 - Raw Picture

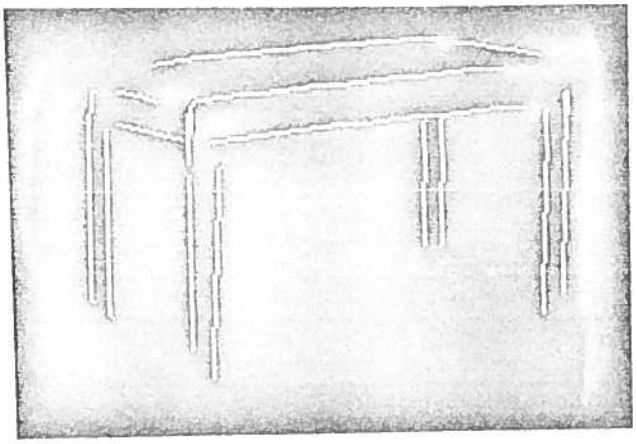

Figure 2 - Recognized Table