Reprinted from O.D.Faugeras (ed.) Fundamentals in Computer Vision

Printed in Great Britain

SYNTACTIC AND STRUCTURAL METHODS II

Thomas C. Henderson
Department of Computer Science
The University of Utah
Salt Lake City, Utah 84112

The emphasis up to now has been on the use of string grammar techniques for the description of shape. However, it is sometimes preferable to trade parsing efficiency for descriptive power, and to this end a much richer class of geometrical grammars and parsing techniques for those grammars have been developed. Alternatively, one can forego grammars altogether and describe shape as a relational structure; this implies a graph matching approach to shape analysis. A side effect of this approach is the construction of a structural description of the shape being analyzed; in the grammatical approach, this description depends directly on the grammar, whereas for relational models, the description depends on the defined relations.

GEOMETRICAL SHAPE GRAMMARS

The notions of formal language theory can be modified so as to allow for a more direct description of the geometrical relations between the pieces of a shape. Thus, geometrical shape grammars must not only describe how the primitve pieces of the shape are joined together (syntactic coincidence), but also how well various relations between geometrical properties of the primitives hold (semantic consistency).

An example of this approach is the stratified context-free shape grammar, or SCFSG, of Henderson & Davis (1981c), which is a quadruple (N,T,P,S) just like for a string grammar, but with a much richer structure. Let V = N U T, then for every v in V,

v = <name part> {attachment part} ⌊semantic part⌋

where <name part> is the unique name of the symbol, {attachment part} is a set of places where the symbol can be attached to other symbols, and ⌊semantic part⌋ is a set of properties, usually geometric, of the symbol.

Each vocabulary symbol represents a piece of the boundary of a shape.

This formulation differs significantly from grammars such as PDL or tree grammars in that a terminal symbol no longer represents an oriented line segment or physically defined piece of the shape, but rather a logical piece of the shape. For example, the terminal symbol of a tree grammar might represent a horizontal line segment of a specific length, while for a SCFSG the terminal symbols would represent wall, floor, roof, etc.

Productions in the grammar are also quite different from ordinary productions. Every p in P is of the form $(v{:}{=}a,A,C,G_a,G_s)$, where the rewrite part $v{:}{=}a$ means that the symbol v in N is composed of the string of symbols $a = v_1v_2 \ldots v_k$ in V^+. Stratification of the grammar is achieved by assigning a level number to each symbol, indicated by $l(v)$; $l(S) = n$, for t in T, $l(t) = 0$, and in any rewrite rule $v{:}{=}a$, if $l(v) = k$, then for every v_i in a, $l(v_i) = k-1$. Furthermore, in order to apply a rewrite rule, it is necessary to check for appropriate syntactic coincidence and semantic consistency which are described by the A and C parts, respectively, of the production. Finally, if the rewrite rule is to be applied, then the G_a and G_s parts of the production describe how to form the attachment part and semantic part, respectively, of the new symbol.

Such a grammar allows an anthropomorphic layout of the semantics of the shape, i.e., productions can be described graphically, and relations such as parallel, equal length, etc. can be used in describing shape. Other approaches to the use of attributes and geometrical shape grammars are given in Vamos & Vassy (1973), Fu (1974) and Gonzalez and Thomason (1978).

PARSING TECHNIQUES

Parsing techniques for shape grammars are usually nondeterministic in nature, and classical top-down and bottom-up techniques have been applied to shape analysis. However, a start has been made toward a theory of shape grammar compilers, see Henderson (1981a) and Henderson & Davis (1981b). The approach advocated there involves producing a parsing mechanism based on both syntactic and semantic constraints between the pieces of the shape. A similar method has been proposed by Masini and Mohr (1978).

A constraint between two symbols can be represented in complete generality by a relation between two sets, see Montanari (1974). If X and Y are sets, then the characteristic function F can be used to represent relation R:

$$F : X \times Y \rightarrow \{0,1\}; \quad F(x,y) = 1 \text{ iff } (x,y) \text{ in } R.$$

If X has m elements and Y has n elements, then 2^{mn} different relations exist between X and Y. An m by n matrix whose entries are just F(x,y) can be used to represent a relation.

Negation, union, intersection and a partial ordering relation of set inclusion can be defined in a straightforward way. Relations form a complete lattice with greatest element the matrix of all ones and least element the zero matrix. Union and intersection act as sup and inf, respectively. A composition of relations is defined as:

$$R_{13} = R_{12}\, R_{23}$$

$$\text{iff} \quad R_{13}(r,s) = \vee[R_{12}(r,t) \wedge R_{23}(t,s)]$$

This is just Boolean matrix multiplication when relations are represented as matrices. A relation is total if every element of X and Y are in relation to some other element.

We would like to consider constraints between more than two variables at a time, but the amount of information grows exponentially with the number of variables. This poses practical difficulties since an n-ary relation R is any subset of $X = X_1 x X_2 x \ldots x X_n$, and if each X_i has m elements, then there are a total of 2^p n-ary relations, where $p = m^n$. However, projections of n-ary relations to networks of binary relations present a useful alternative. A network of binary relations is a set of sets $X = \{X_1, X_2, \ldots, X_n\}$, and a relation R_{ij} from every set X_i to every set X_j, for i,j = 1,n. The nodes of the network are the elements of X, and the relations R_{ij} can be viewed as labeling the edges between nodes. Given X as above, there are 2^p networks of binary relations, assuming $R_{ij} = R_{ji}$ and $p = m^2 * [n * (n+1)/2]$.

N-ary relations can be defined in terms of networks of binary relations. The projection formula proposed by Montanari is:

$$R'_{ij}(i_1,i_2) = \vee\, a(i_1,i_2,i_3,\ldots,i_n)$$
$$\text{where } i_3,i_4,\ldots,i_n = 1,m.$$

Not all n-ary relations are representable by a network of binary constraints. Moreover, an n-ary relation may have many distinct network representations. Montanari shows that the projection formula provides a minimal (with respect to inclusion) network to represent an n-ary relation.

The central problem in using networks of constraints to represent an n-ary relation is to find the minimal network equivalent to the given network. No algorithm is known other than complete enumeration.

We now discuss the procedures for deriving local constraints from the shape grammar. Two types of constraints, syntactic and semantic, are described. The semantic attributes of a vocabulary symbol are computed from the attributes of the symbols which produce it (see Knuth (1968) for a discussion of defining semantics for context-free languages using both synthesized and inherited attributes; we use only synthesized attributes here). Consider a vocabulary symbol as representing a piece of the boundary of a shape. If a hypothesized vocabulary symbol is part of a complete shape, then it is adjacent to pieces of the shape which can combine with it to produce a higher level vocabulary symbol. Therefore, if the set of all possible neighbors of a vocabulary symbol is known, and at one of its attachment points no hypothesis for any of these symbols exists, then that hypothesis can be eliminated. This type of constraint is called a syntactic constraint. The other type of constraint involves some geometric relation between the semantic features of two vocabulary symbols, e.g., the main axis of an airplane is parallel to the axis of its engines.

Let G = (N,T,P,S) be a SCFSG, let v,w,x be in V, let at(v) denote the attachment points of v, and let av be in at(v). We define:

1. v $\underline{\text{Ancestor}}_{av,aw}$ w iff there exists p in P such that the rewrite rule of p is v:=...w... and there exists an aw in at(w) such that aw is identified with av in G_a of p. That is, the attachment point of the left hand side symbol, v, is associated with endpoint aw of the right hand side symbol w.

2. w $\underline{\text{Descendent}}_{aw,av}$ v iff v $\text{Ancestor}_{av,aw}$ w.

3. v $\underline{\text{Neighbor}}_{av,aw}$ w iff there exists p in P such that the rewrite rule of p is x:=...v...w... and aw is specified as being joined to av in A of p, or there exists x in V with ax in at(x) and y in V with ay in at(y) such that x $\text{Ancestor}_{ax,av}$ v, and y $\text{Neighbor}_{ay,ax}$ x, and w $\text{Descendent}_{aw,ay}$ y.

Using matrix representations for these relations, the descendents and neighbors of a symbol at a particular attachment point can be computed, see Henderson & Davis (1981c) or Davis & Henderson (1981) for details.

Semantic constraints can be generated in exactly the same way as syntactic constraints, i.e., by defining binary relations and compiling their transitive closure. For example, the axes of two symbols are parallel if a production states this explicitly, or if each symbol has an ancestor parallel to itself and these ancestors are parallel. Such constraints also permit global constraints to be accounted for, e.g., the orientation of the main axis of an airplane could be fixed, and this certain information propagated throughout the system.

The result of such a parse of an unknown shape is a description in terms of the grammar used. This approach to shape analysis can be viewed as a type of pattern-directed translation in that the graph structure of relations on the primitives is mapped into a structural description based on the underlying grammar.

AMBIGUITY AND NOISE

One of the major problems facing any shape analysis method is that of the noise in the data. For example, extraneous primitives may be generated by spurious edges, or actual primitives of a shape may go undetected due to the poor quality of the image. Another major concern is the ambiguity of the primitives that are extracted to represent the patterns. For example, if the terminal symbols of a grammar represent the line segments oriented at 0, 45, 90 and 135 degrees, then what symbol is assigned an edge oriented at 22.5 degrees? How can this ambiguity be accounted for in the model or the analysis?

The problem of noise can be overcome to a great extent by smoothing the data. Many syntactic shape modeling methods use line segments as the primitive shape elements. Piecewise linear approximations provide an efficient means of data compression and noise suppression. However, the noise elimination techniques are usually directly related to the type of image analyzed and the type of shape model. e.g., string, tree or geometrical.

Elimination of ambiguity in the data is also closely related to the specific shape model used. A common method is to quantize the "primitive space" fine enough so that ambiguous primtives have little effect or are easily accounted for in the model. Alternatively, ambiguity can be accounted for in the grammar by allowing some leeway in the application of productions.

STOCHASTIC GRAMMARS

Stochastic grammars provide a mechanism for describing the effect of random events on a pattern. As such, stochastic grammars are interesting in their own right and need not be restricted to shape grammars. When noise and ambiguity affect the terminal symbols of a grammar, more terminal symbols may be introduced, i.e., new symbols account for deformed versions or perhaps parts of the original terminal symbols. These new symbols have a lower a priori likelihood of being in the data. We consider how this can be accounted for in a string grammar.

Given a language L, every string x in L can have a probability p(x) from (0,1] associated with it such that the sum of the probabilities of all the strings in the language is one. A <u>stochastic grammar</u> is a quadruple (N,T,P,S), where N and T are the non-terminals and terminals, respectively, S in N is the start symbol, and P is a finite set of stochastic production triples (a_i,b_{ij},p_{ij}), $j = 1,\ldots,n_i$, and $i = 1,\ldots,k$, where a_i is in $(N\ U\ T)^*N(N\ U\ T)^*$, b_{ij} is in $(N\ U\ T)^*$, and p_{ij} is the probability associated with the application of the production, p_{ij} in (0,1] and the sum of the p_{ij} is one for $j = 1$ to n_i. Let (a_i,b_{ij},p_{ij}) be in P. Then $c = da_ie$ may be replaced by $f = db_{ij}e$ with probability p_{ij}, that is, $c \Rightarrow f$ with probability p_{ij}. If a sequence of strings $w_1,w_2,\ldots,w_{n+1}$ exist such that $c = w_1$, $f = w_{n+1}$ and $w_i \Rightarrow w_{i+1}$ for $i = 1$ to n, then c generates f with probability $p = \ \ p_i$, and $c \Rightarrow^* f$.

Let L(G) denote the stachastic language generated by G:

$$L(G) = \{(x,p(x)|x \text{ in } T^*,\ S\Rightarrow^* x,$$

$$\text{and } p(x) = \ \ p_j,\ j = 1,k\}$$

where k is the total number of distinct derivations of x from S, and p_j is the j^{th} derivation of x. See Fu (1974) for a complete description of stochastic grammars and recognizers.

GRAMMATICAL INFERENCE

A major inconvenience of grammatical methods is the necessity of designing a shape grammar to model the desired class of shapes. Some effort has been expended to directly infer the grammar from a set of examples. Grammars generated in such a way can be assigned a measure of goodness in terms of their complexity, that is, the number of non-terminal symbols and productions. The two major approaches to grammatical inference are enumeration and induction. Both of these methods will be briefly presented here, and their application to shape grammars discussed; see also Fu (1974) and Fu and Booth (1975a, 1975b).

Gold (1967) has formulated the theoretical study of grammatical inference by enumeration. Basically, given a set of strings in the target language and a set of strings not in the target language, then a grammar must be found. Different types of languages are produced depending on the method of presentation of information: text or informant presentation. The former provides both the set of strings in the grammar and those not in the grammar, while the latter gives only the set of strings in the grammar. An enumeration method to infer finite-state grammars has been developed by Pao (1969). A finite-state grammar is constructed for the set of strings; then, with the help of the informant, the grammar is generalized. Crespi-Reghizzi (1971) developed a method for inferring an operator precedence grammar. The search for a grammar can be made more efficient if the form of the grammar is known; this allows the elimination of a large class of grammars. The notion of one grammar covering another also allows elimination of all those grammars covered by an unsuccessful grammar.

Inductive inference methods proceed by discovering the recursive structure of a grammar. Given a valid string, delete substrings from it and determine if the resulting string is acceptable. If so, then substitute repetitions of the deleted substring and determine if the resulting string is acceptable. If it is, then a recursive structure of the language has been found.

Grammatical inference of shape is, like parsing, somewhat more complicated than its string grammar counterpart. Given a set of patterns, the problem is first to describe the patterns in terms of the chosen shape primitives and relations between them, and then to determine a shape grammar that generates that set. Thus, three things must be determined:

the primitives, the relations and the productions. Evans (1971) has produced a heuristic inference method. After generating a set of possible grammars, a measure of goodness determines which one is chosen. For example, such a measure could be proportional to the number of primitives and relations in the grammar and inversely proportional to the number of productions squared. Lee & Fu (1972) have extended string grammar inference techniques to an interactive shape grammar inference system.

A method for inferring tree grammars is described in Gonzalez & Thomason (1978). Given a sample set of trees, productions are formed for each tree separately. Productions are checked for embedding. Next, equivalent non-terminals are merged. After this step, non-terminals of the appropriate degree are combined. Finally, productions are added so that all the samples derive from the start symbol, and a tree grammar is formed. Examples of 2-D pattern grammar inference are given; however, the number of non-terminals and productions seems inordinately high. Techniques also exist for stochastic grammar inference, and Gonzalez and Thomason (1978) and Fu (1974) dicuss these.

RELATIONAL MODELS

Several high-level relational models have been described in the literature, see Fischler & Elschlager (1973), Davis (1979), Shapiro & Haralick (1979) and Shapiro (1980). Such models use convex parts or line segments as shape primitives. Once a decomposition of the shape in terms of the primitives is achieved, a relational description of the shape is constructed in terms of the relations. For example, Shapiro (1980) uses ternary relations between primitives which form intrusions and protrusions. A search is then performed to find mappings from a prototype to a test shape. This is equivalent to a subgraph isomorphism problem, and consequently is NP-complete. However, special look-ahead operators are used to reduce the amount of search required, see Haralick & Shapiro (1979). This approach to shape modeling allows for simple, yet sufficient models, which permit inexact and partial matches. However, the construction of the relations is quite expensive, being of order n^3 complexity.

Semantic networks offer a similar approach to shape description, e.g., see Ballard et al (1978). Each node of the net represents a primitive, and nodes are connected by one or more arcs which indicate the relation between those nodes. Operations defined on a semantic net include: pattern-directed access, updating (deletion, addition and modification of nodes or arcs) and inference. Semantic nets are very limitied in scope unless they allow quantification and the use of logical connectives. Moreover, in most systems, encoding and decoding are done by hand.

REFERENCES

Ballard, D., Brown, C. & Feldman, J. (1978). An Approach to Knowledge-Directed Image Analysis. In Computer Vision Systems, eds. A. Hanson and E. Riseman, New York: Academic Press.

Crespi-Reghizzi, S. (1971). An Effective Model for Grammar Inference. IFIP Congress, Yugoslavia.

Davis, L. (1979). Shape Matching Using Relaxation Techniques. IEEE T. on Patt. Anal. Mach. Intell., PAMI-1, no. 1, 60-72.

Davis, L. & Henderson, T. (1981). Hierarchical Constraint Processes for Shape Analysis. IEEE T. on Patt. Anal. Mach. Intell., PAMI-3, no. 3, 265-277.

Evans, T.G. (1971). Grammatical Inference Techniques in Pattern Analysis, In Software Eng., ed. J.T. Tou, Vol. 2, New York: Academic Press.

Fischler, M. & Elschlager, R. (1973). The Representation and Matching of Pictorial Structures. IEEE T. on Computers, C-22, 67-92.

Fu, K.S. (1974). Syntactic Methods in Pattern Recognition. New York: Academic Press.

Fu, K.S. & Booth, T.L. (1975a). Grammatical Inference: Introduction and Survey - Part I. IEEE T. on Systems, Man, and Cybernetics, SMC-5, no. 1, 95-111.

Fu, K.S. & Booth, T.L. (1975b). Grammatical Inference: Introduction and Survey - Part II. IEEE T. on Systems, Man, and Cybernetics, SMC-5, no. 4, 409-423.

Gold, E.M. (1967). Language Identification in the Limit. Inf. and Control, 10, 447-474.

Gonzalez, R.C. & Thomason, M.G. (1978). Syntactic Pattern Recognition. Reading, Ma: Addison-Wesley.

Haralick, R.M. & Shapiro, L. (1979). The Consistent Labeling Problem. IEEE T. on Patt. Anal. Mach. Intell., PAMI-1, no. 2.

Henderson, T.C. (1981a). Shape Grammar Compilers. In Digital Image Processing, eds. J.C. Simon and R.M. Haralick, pp. 327-336, Dordrecht, Holland: D. Reidel.

Henderson, T.C. & Davis, L. (1981b). Compilateurs de Grammaires de Formes. Proc. 3rd AFCET Conf. on Patt. Rec. and Art. Intell., Nancy, France, 415-424.

Henderson, T.C. & Davis, L. (1981c). Hierarchical Models and Analysis of Shape. Pattern Recognition, 14, nos. 1-6, 197-204.

Knuth, D. (1968). Semantics of Context-Free Languages. Math. Systems Theory, 2, no. 2, 127-145.

Lee, H.C. & Fu, K.S. (1972). A Syntactic Pattern Recognition System with Learning Capability. Proc. Int. Symp. Comput. and Inf. Sci. (COINS-72), Miami Beach, Fl, 14-16.

Masini, G. & Mohr, R. (1978). Un Systeme de Reconnaissance des Formes. Proc. AFCET Symp. on Patt. Rec., 107-114.

Montanari, U. (1974). Networks of Constraints: Fundamental Properties and Applications to Picture Processing. Inf. Sci., 7, 95-132.

Pao, T.W. (1969). A Solution of the Syntactical Induction-Inference Problem for a Non-Trivial Subset of Context-Free Languages. Interim TR 69-19, U. of Pennsylvania.

Vamos, T. & Vassy, Z. (1973). Industrial Pattern Recognition Experiment - A Syntax Aided Approach. IJCPR, Wash., 445-452.