LEARNING SHAPE DESCRIPTIONS

Jonathan H. Conn ell and Michael Brady

Massachusetts Institute of Technology, Artificial Intelligence Laboratory
545 Technology Square, Cambridge MA 02139

ABSTRACT

We report on initial experiments with an implemented learning system whose inputs are images of two-dimensional shapes. The system first builds semantic network shape descriptions based on Brady's *smoothed local symmetry* representation. It learns shape models from them using a modified version of Winston's *ANALOGY program.* The learning program uses only positive examples, and is capable of learning disjunctive concepts. We discuss the lcarnability of shape descriptions.

1. Introduction

We report on initial experiments with an implemented system that learns two-dimensional shapes from images. The system first builds semantic network descriptions of the imaged shape based on Brady's *smoothed local symmetry* representation [Brady and Asada 1984, Ileide 1984]. It learns shape models from the descriptions using a modified version of Winston's *ANA LOG Y* program [Winston 1980, 1981, J982; Winston, Binford, Katz, and Lowry 1984]. The inputs to the program are grey-scale images of real objects, such as tools, model airplanes, and model animals. The outputs of the program are production rules that constitute a procedure for recognising subsequent instances of a taught concept.

Figure la shows the gray-scale image of (a model of) a Boeing 747, Figure lb shows the results of Brady's smoothed local symmetries program, and Figure lc shows a portion of the semantic network that is computed from them by our program. The semantic network is transformed into a set of associative triples [Doyle and Katz 1985] and input to our learning program. The 747 generates 239 associative triples. Similarly, Figure 2a shows the subshapes found from the smoothed local symmetries of a tack hammer and Figure 2b shows the full semantic net for this image. The tack hammer generates 51 associative triples.

The learning program is a modification of Winston's *ANALOGY* [Council 1985]. It is capable of learning concepts containing disjunctions. The program learns shape models using positive examples only. Figure 3b shows the concept *hammer* that is learned from the three positive instances shown in Figure 3a.

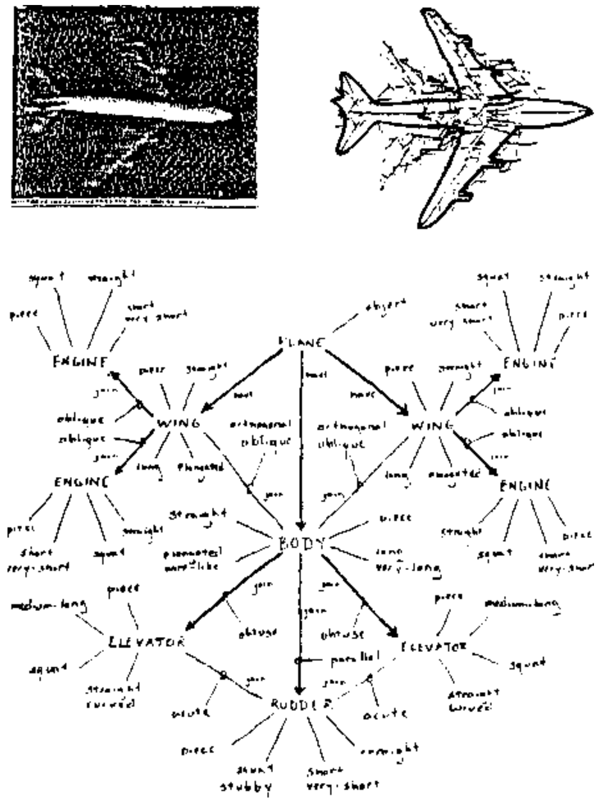

Figure 1. a. The input image, b. The smoothed local symmetries *of* the plane c. A portion of the hierarchical semantic network that is computed from the information m b. The full network generates 239 associative triples.

The novelty of our work is the ability to learn visual shape representations from real visual data. Previous work has not been based on real data because such data was unavailable or too complex and unstructured for existing learning algorithms. However, recent developments in edge-detection [Canny 1983] and middle-level vision [Brady and Asada 1984] have provided a solid base on which to build a robust vision system. Using this system we can generate shape descriptions in a form amenable to learning. Furthermore, although the descriptions typically comprise between fifty and three hundred assertions, various forms of abstraction keep this volume of data manageable.

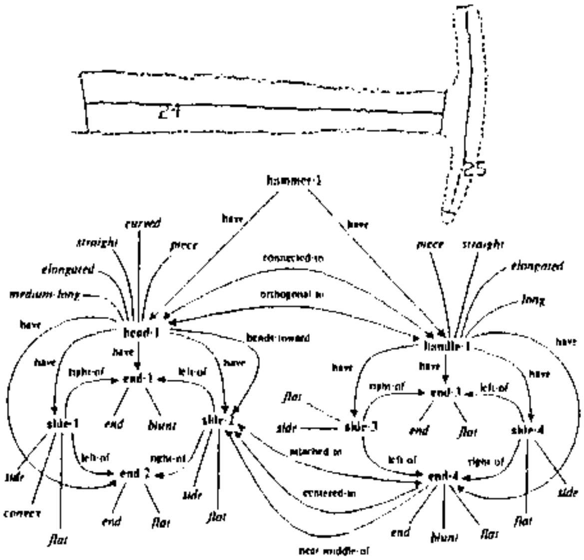

Figure 2. a. The main nmoothed local symmetries computed from the results of Brady's program, h. The semantic network that is computed from the information in a.

2. Representing Shape

To describe an object it is necessary to first segment it into separate subshapes. In terms of the mathematical analysis in Brady and Asada 1984 i, a subshape is defined as maximal with respect to smooth variations in the defining parameters. For example, the portions of fuselage in front of and behind the wings of the B747 in Figure 2 are joined, but the handle and blade of a screwdriver arc perceived as separate pieces. Once a part has been found, its shape is specified by three numbers: the aspect ratio, the curvature of the axis, and the change in width along the axis.

Joins between subshapes are determined by examining the spines of the regions and the adjacency of the contour segments. A join is specified by the relative angle and sizes of the pieces, and the location of join with respect to each piece. Few previous representations of shape have described subshape joins. For example, *ACRONYM* (Brooks 1981, Brooks and Binford 1980] specified the coordinate transformation between two joined pieces, but did not explicitly describe the join.

Once we break the image into pieces and find the joins we must somehow represent this information. Images are noisy, so it is necessary to develop representations that are stable, in the sense of being invariant under localized changes such as image noise. However, tasks involving visual representations, for example inspection, often require that programs be sensitive to fine detail. A variety of techniques for simultaneously achieving stability and sensitivity have been proposed, each expressing some aspect of hierarchical description. The underlying idea is that gross levels of a hierarchy provide a stable base for the representation, while finer levels increase sensitivity.

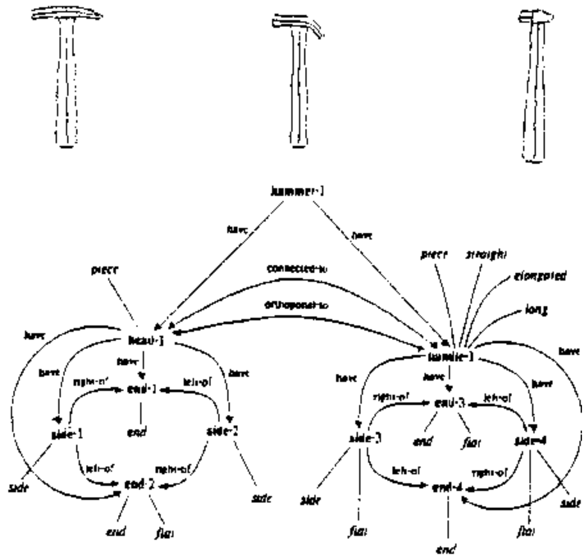

Figure 3. The concept *hammer* that is learned from the three positive instances shown above.

A vision program needs to maintain several different representational hierarchies, including the following:

• Numeric values and symbolic descriptors

Specifying a shape parameter of interest, say a measure of the elongation of a shape, by a numerical value is sensitive, but highly unstable. Symbolic names that correspond to an interval of numeric values are (usually) more stable but less sensitive. Our representation employs symbolic descriptors that have overlapping ranges. For example, an end which is determined to be on the borderline between blunt and sharp is declared to be both blunt and sharp. Overlaps like this help to combat the quantization error introduced by encoding a continuous range as a set of discrete symbolic values. A small change in value leads to a small change in the representation.

• Structural approximations to shapes

Marr and Nishihara [1978] proposed summarizing the lesser subparts of an object, leaving them unspecified until they are needed. For example, all airplanes have a fuselage, with pairs of symmetrically attached wings and elevators. Upon closer examination, a wing of a B747 has two attached engine pods, a DC10 has one, and an L1011 none. Suppressing mention of the engine subshapes, as well as summarizing the parameters that describe the shapes of the wings and fuselage, enables the descriptions of the three airplanes to closely match each other.

In general, larger subshapes tend to determine gross categorization, and so they tend to appear higher in the structural hierarchy. Conversely, smaller subshapes tend to allow finer discrimination and occur lower in the hierarchy. The smaller subparts of a tool typically determine the specific function of the tool. For example, deciding whether a tool is an awl, a gimlet, or a Phillips screw driver involves looking closely at the end of the blade; the relatively localized *context* of the business end of the blade is established by the grosser levels of the hierarchy, where it is recognized (for example) that the tool is not a hammer or wrench. In this way, the Marr-Nishihara proposal tends (hcuristically) to relate large scale geometric structure to gross functional use.

- A-kind-of hierarchies

Family hierarchies are ubiquitous, and apply as much to visual shape representations as to the more cognitive situations in which they were developed in Artificial Intelligence. *ACRONYM* represents the fact that the sets of B747-SPs, B747s, wide-bodied jets, jets, and aircraft, are ordered by subset inclusion. Similarly, a claw hammer is a-kind-of framing hammer, which is a-kind-of hammer. In general, a subset hierarchy is a partially-ordered set, but not a tree. From the domain of tools, for example, a shingle ax is both a-kind-of ax, and a-kind-of hammer.

3. Learning

The commonest form of inductive generalization used to learn concepts from positive examples is the *drop condition* heuristic IDietterich and Michalski 1981, Winston 1984, page 398). This is the method used in our program. Through careful design of the representation the method has been extended to allow generalizations of intervals and structural graphs.

The idea behind the heuristic is that if two things belong to the same class then the differences between them must be irrelevant. Accordingly, when we have a partial model of a concept and receive a new example, we modify the model by deleting all the differences between it and the example. This can be seen by comparing Figure 2b with Figure 3b. Notice that the network in Figure 3 puts very little constraint on the size or shape of the head. This is because the shapes of the heads in the examples vary widely. For instance, the heads of the first and third hammer are straight while the head of the second hammer is curved. Note also that the manner in which the handle joins the head.is only loosely specified. This is because the handle is joined to the side of the head in the first two examples but to the end of the head in the third example.

This is a simplified explanation of the learning algorithm. The matching involved is not graph isomorphism nor is it, merely counting the number of required features an object has. Rather it is a complex local matching scheme. Consider using the semantic net shown in Figure 1 as the model for the *airplane* concept. For an object to match this model, at the top level it must have three pieces which look similar to the three in the model. A piece of the example is similar to the wing model if, first of all, it has the shape specified in the network and, second, it has two things which look like engines attached to it. Suppose that a certain piece has the right shape for a wing but has only one engine attached to it. At the level of the wing model the program notices that there is a discrepancy yet judges that the piece is still close enough to the description to be called a wing. When the top level of the matcher asks if the piece in question looks like a wing the answer is "yes". No mention is made of the fact that the wing is missing an engine. The difference only matters locally and is isolated from the higher levels of matching.

Another important concern is limiting the scope of generalizations made. Imagine that the program is shown a positive example that is substantially different from its current model. Altering the model by the usual induction heuristics typically leads to gross over-generalization. This, in turn, runs counter to what Winston [1984, page 401] has dubbed *Martin's law,* namely: learning should proceed in small steps. Therefore our program creates a new, separate model based on the new example, splitting the concept being taught into a disjunction.

In some cases, the disjunction will be replaced by a single model as positive examples are taught that are intermediate to the disjuncts. For example, suppose that the first example of a hammer shown to the program is a claw hammer, and that the second is a sledge hammer. The program will create a disjunction as its concept of hammer, but it will be consolidated into a single model once it has seen such examples as a mallet and ballpein hammer.

Even though the program only generalizes a concept using an example that is structurally similar, it is sometimes deceived and must recover from over-generalization. We follow Winston [1984] and provide censors that override the offending rule. Censors can be generalized and there can be disjunctive censors; in fact this is the usual case. Since censors can be generalized they also have the possibility of being over-generalized. This is countered by putting censors on the censors. In general, a concept is not represented by a single model but by a group of models. There can be several positive models corresponding to the disjuncts as well as several negative non-models summarizing the exceptions to the other models.

4. Current Work

The goals of our research are not limited to learning. The work reported here forms part of the *Mechanic's Mate* project [Brady, Agre, Braunegg, and Conneil 1984], which is intended to assist a handyman in generic assembly and construction tasks. The primary goal of that project is to understand the interplay between reasoning that involves tools and fasteners and representations of their shape.

For example, instead of learning that a certain geometric structure is called a hammer, we learn that something which has a graspable portion and a striking surface can be used as a hammer. These two functional concepts are then defined geometrically in terms of the shape representation. Reasoning from function as well as from form allows more flexibility. For instance, faced with a hammering task, but no hammer, one might try mapping the hammer structure onto that of any available tool. A screw driver provides a good match, identifying the blade of a screw driver with the handle of the hammer, and the (assumed flat) side of the screw driver handle with the striking surface of the head of the hammer. In this way, the Mechanic's Mate can suggest improvisations, like using a screw driver as a hammer.

Our initial goal was to learn shape models cast in the representation described previously. Eventually, the *Mechanic's Mate* will have to learn about the non-geometric properties of objects: weight, material type, and the processes that use them. Currently we are using Katz's English interface |Katz and Winston 1983] to tell our program such things. This is not satisfactory. Instead, we hope to teach dynamic information using a robot arm and hand.

Another area of interest is inducing structural subclasses from examples. Since the subclasses that form the a-kind-of hierarchy are an important part of the shape representation, they should be learnable. However, in learning subclasses there is a danger of combinatorial explosion. Learning subclasses requires a suitable similarity metric. Feature-based pattern recognition systems learn subclasses as clusters in feature space, and clusters are sets that are dense with respect to the Euclidean metric. Part of our research in learning shape descriptions has been to determine what makes objects look similar. This suggests using the metric employed in the learning procedure to form subclasses through a process analogous to feature space clustering. This is the focus of our current work.

5. Acknowledgements

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's Artificial Intelligence research is provided in part by the the System Development Foundation, the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-80-O-0505, and the Office of Naval Research under contract number N00014-77-C-0389. We thank the people who have commented on the ideas presented in this paper, particularly Phil Agre, Steve Bagley, Hob Berwick, Hen DuBoulay, Alan Bundy, Margaret Fleck, Scott Heide, Boris Katz, Tomas Lozano-Perez, John Mallery, Tom Mitchell, Sharon Salveter, Dan Weld, and Patrick Winston.

References

Brady, Michael, and Asada, Haruo, [1984], "Smoothed Local Symmetries and their Implementation," *Int. J. Robotics Research,* 3 (3) .

Brady, Michael, Agre, Philip, Braunegg, David J., and Connell, Jonathan H., [1984], The Mechanic's Mate, ECAI *84:* Advances in Artificial Intelligence, O'Shea, T. (ed.), Elsevier Science Publishers B.V. (North-Holland).

Brooks, Rodney A., [1981], "Symbolic Reasoning Among 3-D Models and 2-D Images," *Artif. Jntell,* 17, 285 - 348 .

Brooks, Rodney A., and Binford, Thomas O., [1980), Representing and Reasoning About Partially Specified Scenes, *Proceedings, DARPA Image Understanding Workshop,* Baumann, Lee S. (ed.), Science Applications Inc., 150 156.

Canny, John Francis, [1983], Finding Edges and Lines in Images, MIT Artificial Intelligence Laboratory, Cambridge Mass., AI-TR-720.

Connell, Jonathan H., [1985], forthcoming SM thesis, MIT Department of Electrical Engineering.

Dietterich, T. G., and Michalski, R. S, [1981], "Inductive Learning of Structural Descriptions," *Artif. Intell,* 16 .

Doyle, Richard J., and Katz, B, [1985], Exploring the Boundary Between Natural Language and Knowledge Representation, MIT Artificial Intelligence Laboratory, Forthcoming AI Memo.

Heide, S, [1984], A Hierarchical Representation of Shape, SM thesis, MIT Department of Mechanical Engineering.

Katz, Boris, and Winston, Patrick H, [1983], A Two-way Natural Language Interface, *Integrated Interactive Computing Systems* P. Degano and Erik Sandewall (eds.), North-Holland, Amsterdam.

Marr, D., and Nishihara, H. K, [1978], "Representation and Recognition of the Spatial Organisation of Three Dimensional Shapes," *Proc. Roy. Soc. Lond.* ii, 200 , 269 - 294.

Winston, Patrick II., [1980], "Learning and Reasoning by Analogy," *Comm. ACM,* 23 , 689 - 703.

Winston, Patrick H., [1981], "Learning New Principles from Precedents and Exercises," *Artif. IntelL,* 19 , 321 - 350.

Winston, Patrick H., [1982], Learning by Augmenting Rules and Accumulating Censors, MIT Artificial Intelligence Laboratory, AIM-678.

Winston, Patrick H., [1984], *Artificial Intelligence, 2nd. Ed.,* Addison-Wesley, Reading, Ma..

Winston, Patrick H., Binford, Thomas 0. , Katz, B., and Lowry, M., [1984], Learning Physical Descriptions from Functional Definitions, examples, and precedents, Robotics Research, Michael Brady and Richard Paul (eds.), MIT Press, Cambridge, 117 - 135.