THE REPRESENTATION OF SHAPE

Andrew Latto
David Mumford
Jayant Shah

Reprinted from IEEE 1984 PROCEEDINGS OF THE WORKSHOP ON COMPUTER VISION REPRESENTATION AND CONTROL

IEEE COMPUTER SOCIETY
1109 Spring Street, Suite 300
Silver Spring, MD 20910

IEEE COMPUTER SOCIETY PRESS

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC

THE REPRESENTATION OF SHAPE

Andrew Latto, David Mumford

Department of Mathematics
Harvard University
Cambridge, MA, 02138

Jayant Shah

Department of Mathematics
Northeastern University
Boston, MA, 02115

1. Introduction

A computer vision system is a program that accepts as input some class of 2-dimensional intensity arrays (or images for short) and outputs a description of certain of the objects present in the image, i.e. it picks out certain regions and points in the image, organizes these in various ways depending on the type of object and gives them semantic labels (e.g. chair, rib, wrench, airplane, etc.). To accomplish this, it is often supposed that modules of something of the following type will have to be present:

a) Some bottom-up low-level processing modules producing something like Marr's 2 1/2-D sketch (see Marr (1982)).

b) A knowledge base of prototype objects or object parts (usually stored as a database, i.e. declaratively, but can be stored either procedurally or via strength of connections in connectionist models).

c) Some process of bringing the two together, which can be largely bottom-up or top-down, can be local or global, may involve constraint analysis, can be one step or an iterative relaxation process, etc.

The subject of this paper is the nature of the stored description of the prototype objects. Even if the prototype is stored procedurally or in connection strengths, the procedure or the connections can usually be thought of as embodying some collection of features or properties of the object being identified. The most low-level possibility, for instance, is to store a template, a reference 2-dimensional array, which would be matched 1-1 with some part of the image. At the other extreme, one could imagine the prototype objects being lists of high-level properties: e.g. the letter "A" would be a list (made-up-of-thin-lines, sharp-point-at-top, two-feet-at-bottom,...), a German shepherd would be (furry, four-footed, pointy-eared,...). Both of these seem clearly inadequate for the task of visual identification. The purpose of this paper is to propose a new knowledge representation scheme for describing a restricted class of shapes, namely 2-dimensional, black and white shapes with boundaries made up of curves that are smooth, except possibly at corners. Such shapes arise whenever some part of the image is singled out by a surrounding contour, by homogeneous texture or by thresholding. Namely, a new image is formed with this part black, the rest white, and the surrounding contour is smoothed if necessary. We will call our data structures shape descriptors. These shape descriptors can be used to represent prototype objects in the database and as an intermediate-level, pre-semantic representation of an image which can be matched against shape descriptors in the knowledge base. We want our shape descriptors to have the following properties insofar as possible:

a) The shape descriptor should contain enough data to enable a program to reconstruct an image which looks to a person generally similar to the original (different degrees of detail may be called for in different contexts so the shape descriptor has to allow refinements or simplifications and omissions when necessary).

b) The shape descriptor should contain only information which is invariant under all perturbations of the image leaving the shape looking generally the same to a person. These perturbations include nonlinear distortions and even ones which do not preserve topology.

These seem to us to be essential features of a knowledge representation scheme if it is to be efficient for the purposes of shape identification. Note that these properties imply that the shape descriptor is not an information preserving representation of the image. The amount of information present will vary from context to context, but it seems unlikely that any context will require the shape descriptor to be so precise as to allow complete reconstruction of the image. We are designing a program to use the shape descriptors described below to actually identify simple shapes of several different categories: handwritten characters, leaves of trees, and silhouettes of common tools and of animals. This will be reported on elsewhere.

It might be objected that such 2-dimensional black and white shapes with boundaries made up of smooth curves constitute a wholly artificial toy world (like the blocks world) which is irrelevant to object identification in the real world. We do not think so. First of all, convolving an image with a Gaussian and thresholding gives you such images on various scales from a real image. Secondly, the essence of the complexity of the perceptual quality "shape" already seems to be displayed by these 2-dimensional images. Thirdly, in a real-life setting where a multiplicity of scales and textures are present, and three dimensional objects are rotated and partially obscured, it seems to us unrealistic to make identifications by a one-stage feature extraction plus database comparison process. We think the bigger problem is broken up into several stages or levels and that our shape descriptors can be used at several stages at each scale for different purposes.

In §2, we will review previous work on the description of shape. In §3, we will describe and motivate, partly on psychological grounds, the ingredients of our shape descriptors. In three appendices, we will illustrate the theory with three nearly complete examples.

2. Survey of Shape Representation Schemes

One can classify all shape representation schemes that we have seen into the following groups:

a) Heavily numerical schemes
b) Chain coding schemes
c) Feature lists
d) Structural analyses

(a) and (b) are extensively used in the image processing and pattern recognition community, (c) by psychologists, and (d) by the AI community. Our proposal will be of type (d).

Let's consider the numerical schemes. It is clear that 'shape' is a global property of the image so one approach is to take a double integral over x and y to extract some global invariants. Two principle integrals have been used: the spatial Fourier transform and the moments of the image (i.e. $\iint I(x,y)x^n y^m dxdy$, cf. Hu (1962), Duda & Hart (1973), §9.3.6.2). There are several difficulties with these techniques. Although the numerical description of the image these integrals give changes in a simple way under rotations, translations and dilations of the image, they do not change in any simple way under non-linear distortions of the image, even small ones which don't affect the subjective shape of the image. Another problem is that the topology of the image (e.g. a circle vs. an open 'U' shape) is not mirrored in the Fourier transform or the moments. For instance, consider a normally drawn letter and one drawn to imitate a broad-nibbed pen in which certain lines are thick, others thin (figure 1a,1b). As far as double integrals are concerned, figure 1b will be much closer to figure 1c than to figure 1a! We think there is a possibility that certain global shape invariants can be detected from such integrals, but this seems unproven. (Spatial band pass filters are another matter: although these can be calculated by taking a spatial Fourier transform, truncating, and taking the inverse transform, they then provide modified input to further processing and cannot be considered ways of representing shapes or extracting features of shapes by themselves.)

Another numerical approach to shapes is by approximating elementary curves. Thus one can find points of maximum curvature on the boundary of a 2-dimensional shape and use these to form a polygonal approximation. Or if intermediate values of the curvature of the boundary are important, one can use splines to give a better approximation to the boundary. Both of these have 3-dimensional versions used to approximate surfaces. All representations of this type have the following problem: they make explicit the relationship between adjacent points along the contour but do not make explicit where and in what ways the contour doubles back on itself. This doubling back is immediately apparent to the human eye and strongly affects our idea of the shape. Moreover, calculation of the geometry of the region enclosed by a contour doubling back on itself from such a representation is not numerically stable. Thus the two shapes in figures 1d and 1e can each be put together out of a dozen line segments with almost identical slopes, i.e. locally corresponding pieces of the two contours are very similar. But small changes of the slopes greatly affect the area and shape of the whole, due to the curve doubling back nearly perfectly as well as the 'jaws' nearly closing in figure 1e. One figure is an island with a large bay, the other a ring with a small gap. This problem arrises from the use of a one-dimensional data-structure to represent shape, an inherently two-dimensional attribute. A second problem is how to choose points on the

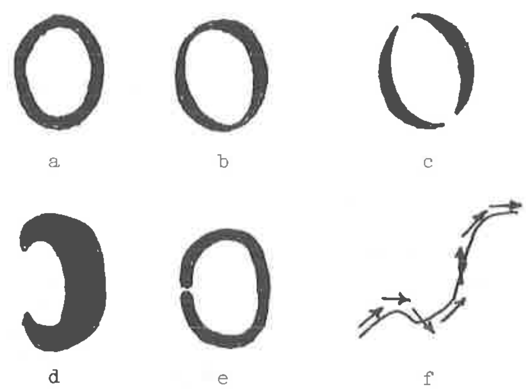


FIGURE 1

contour to break it up. If the contour has small features as well as large ones, the mesh size will have to be varied along the contour or else a very large number of points will have to be used. If a large number is used, the representation contains many redundant bits; if a smaller number is used, the problem is how to choose the 'right' ones. If the mesh size is varied, then one has to have a theory for it.

Chain codes are fundamentally a discrete version of the idea of approximation by elementary curves. The idea goes back to Freeman (1961) of approximating a boundary contour by sequences of horizontal, vertical and diagonal steps (figure 1f). This gives an 8-letter alphabet which is then processed by techniques from formal language theory (see Ledley (1965), Shaw (1969), Pavlidis & Ali (1979), Fu (1982)). Related ideas have been proposed in the psychological literature, with special attention to the explicit representation of whole or partial symmetries or repeated patterns (see Leeuwenberg (1971), Simon (1972)). The objections raised against other curve approximation schemes apply here too; since the representation is one-dimensional, the shape of the interior region is not made explicit. While it seems a good idea on the basis of economy to use discrete representations rather than continuous ones, it is not clear that discretizing the curve so totally at such an early stage is a good idea. We are very sensitive to the information in the curvature of a contour and it seems better to compute this while a continuous representation is available, before discretizing. Another objection to the use of chain codes as a shape representation is that if edges are present that form a graph with nodes (e.g. the skeleton the the letter 'A'), chain coding such a graph requires one to pick out a maximal subtree by some arbitrary choice. A final objection, not to chain-codes themselves but to their use as 'surface representations' of a grammar, is that we have seen no convincing example where a traditional context-free grammar seems to be the most natural way to capture the constraints present in semantically significant classes of images.

A representation by a feature list is the classical psychological representation of a concept or category. One is presented with an exemplar and one checks, one by one, a list of features. If all are present, or enough are present to bring some weighted sum over a threshold, then the exemplar is declared to belong to the category (See for example Rosch and Lloyd (1978), Smith and Medin (1981)). In the realm of visual categories, this is the type of 'deep representation' hypothesized by Kosslyn (1980). Neurophysiologically the analog of this idea is that the firing of each neuron detects the presence or absence of a specific feature in the stimulus. This approach is often joined to more numerical methods via cluster analysis, i.e. if half a dozen numerical invariants of a figure are chosen, then one can map images to points in Euclidean 6-space and look for clusters of these points and the simplest combinations of these features that distinguish these clusters. However, we feel that the features that images exhibit should not be just listed, they should be tightly organized in a structure that incorporates their interdependence. Thus features along a curve or an axis come in a definite order. Compare figure 2a vs. 2b or figure 2c vs. 2d. Likewise proximity groups subsets of features together and repetition and parallelism provide structural links between features. It might be suggested that one should incorporate interdependence relationships as features in their own right. However, it is not clear how one would then avoid combinatorial explosion in the dimension of the feature space.

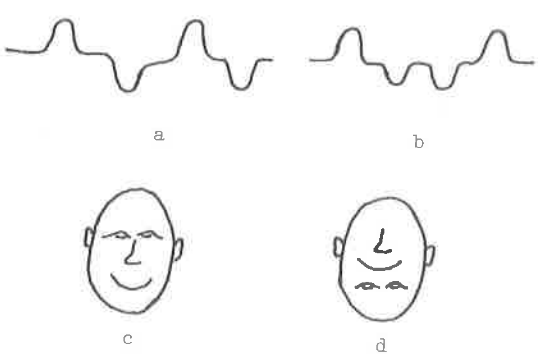


FIGURE 2

The last type of representation scheme is the one that we favor. The essential idea is to decompose the object into components or parts, each with various salient features, and to build up a description of the whole by a frame or net that expresses the links between these pieces. This idea has many roots but it is made explicit in two important papers: Marr & Nishihara (1978) and Minsky (1975). The origin of our project was the attempt to make more explicit and apply the ideas in Marr & Nishihara. Many of our ideas parallel ideas introduced in the recent paper of Davis (1983). We discuss our proposal in detail in the next section.

3. What goes into a shape descriptor

Rather than writing out formally what the most general shape descriptor is, we will introduce its aspects one at a time, motivating each one in turn. Some of the references and psychological data that we give as motivation refer to 3-dimensional shapes, but, we repeat, our shape descriptors apply only to 2-dimensional black and white shapes.

A. Components: The idea that general shapes should be broken up into more 'components' is an old one. Pavlidis (1968) proposed describing an arbitrary polygon in terms of the maximal convex sub-polygons whose sides were all part of lines extending the sides of the given polygon. Binford (1971) proposed describing a 3-dimensional object as a union of more elementary shapes called generalized cylinders. Alternately, a spine or skeleton can be sought in a general shape by the medial axis transform (Blum (1973)) or other thinning algorithm. Then the shape can be represented as the union of the pieces which thin to or surround the segments of the skeleton. These ideas have been developed by many people, especially Agin (1981), Brady (1983), Shapiro (1980) and Shapiro et al (1980). Our descriptors are based on this idea too, with one key extension: we distinguish 3 different types of components which play very different roles. The first we

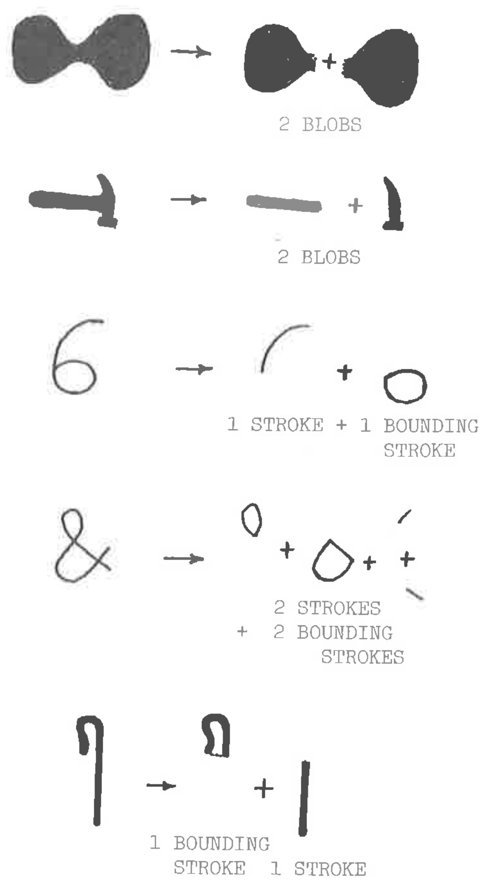


FIGURE 3

call a blob and it is to be a roughly convex region (i.e. convex except for details on its boundary and possibly small holes in its interior) which is not extremely elongated. (We will not give numbers to define exactly 'nearly' or 'extremely': this is a task which can only be done in the context of a working program.) The second type of component we call a stroke. It is a thin region bounded by nearly parallel lines without sharp changes in direction and which does not enclose or nearly enclose another region. The third we call a bounding stroke. It is a thin region which may have corners and which bounds or nearly bounds a blob.

It is our contention that any 2-dimensional shape with boundary made up of smooth curves has at least one and at most a small number of (not necessarily disjoint) decompositions into such components (with the proviso that most of the boundary of each component is part of the boundary of the original shape). The argument for these distinctions is that natural forms tend, on a given scale, to be seen sometimes as truly 2-dimensional shapes (thick regions) and sometimes as essentially 1-dimensional (thin, extremely elongated regions). Likewise, necks are natural break points for 2-dimensional objects, and sharp corners are natural breakpoints for 1-dimensional objects (unless it is the boundary of an indivisible 2-dimensional shape). A 2-dimensional shape without a narrow neck must be nearly convex. Finally, strokes which double back and cross themselves get extremely complicated and a natural way to control this seems to be to organize them in that case in terms of the regions they bound.

B.Hierarchy: Marr & Nishihara (1978) stressed that a decomposition as above should be hierarchical. Their famous picture (figure 4) of a man as a hierarchical union of generalized cylinders makes their point immediately clear. We also incorporate such a hierarchy based on physical scale (similar to the 'part-of' hierarchy used in the theory of semantic nets) with the extra remark that components lower on the hierarchy may vary from thick to thin and vice versa, and may also be both positive and negative, i.e. holes in a bigger shape are shapes in their own right after reversing figure and ground.

Putting together the hierarchy and the decomposition into components, we produce a concept of a shape descriptor which is a tree. Its nodes represent parts of the shape which are readily perceived as whole shapes in their own right. The top or root node is the whole shape. Each node is either (a) of compound type in which case the immediately lower nodes are its components; or (b) it is a blob, stroke, or bounding stroke, in which case the immediately lower nodes are either detailed parts of the blob or stroke, or else they are islands, bounded pieces of the complement of the shape. Several examples are found in the appendices.

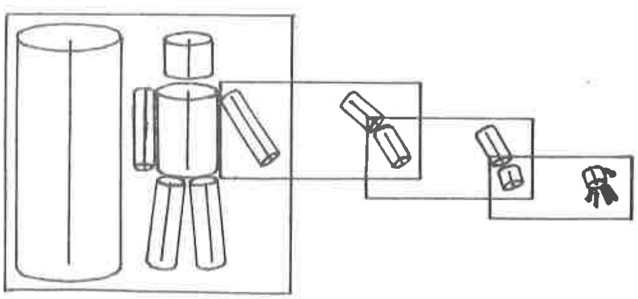


FIGURE 4

C.Numbers and Coordinates: We do not feel that highly precise coordinates have any place in the shape descriptor. For instance, given two prominent points in a figure, it is almost always unimportant whether one is 2.1 or 1.9 times as far from the x-axis as the other. The theory of 'naive physics' (Hayes (1979), deKleer (1983)) has shown how intuitive physical reasoning seems to be well captured in a world in which all physical quantities take values in a set with only three elements (-,0,+)! For this surprising conclusion to work, it is essential to know sometimes not only the value of a quantity to this precision, but also the value of the difference of 2 quantities if they are given in the same units, and the value of the first and second derivatives of a quantity. The analog in naive geometry seems to be to take 3x3 coordinates like a tic-tac-toe board for position, a chain-code approach to direction (i.e. N,NE,E,SE,..) and a 3-valued set of curvatures (-,0,+) (see Figure 5). At first, it sounds like this is throwing away too much information to hope to be able to reconstruct a similar shape. But suppose we supplement these charts as follows:

a) Lay out a separate 3x3 coordinate chart for each node, small or large.

b) Relate the charts for lower nodes to the one above them by a location, a scale factor, and an orientation (this is indicated in the appendices by notations such as (loc SE, scale 2, dir W).)

c) Orient each chart along the axis of the component in question and allow the x- and the y-scales to differ by an 'aspect ratio' so as to fit the component. This aspect ratio and the scale factor in (b) will have a higher degree of precision.

In (c), the axis of a component will be the direction in which it is elongated (calculated for instance via the moments), or the direction of a very prominent straight line, or the axis of a complete or nearly complete symmetry of the component.

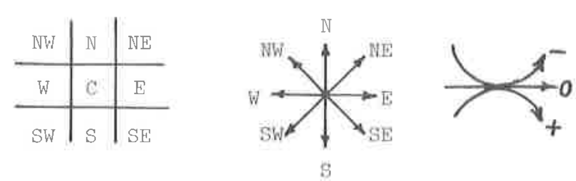


FIGURE 5

Now if two points of a shape are nearby and have the same 3x3 coordinates, you can find their relative position by joining them by a virtual line and noting its direction. (In some settings, such as the perception of faces, human beings seem very sensitive to degrees of curvature in edges. One possible way to incorporate this is to imagine a virtual circle osculating the given edge: the relative position of the center of this circle is a rather accurate way of describing the curvature of the edge.)

D. Tours of edges: Coming down to specifics of how individual components are described, the main data structure is an analog of a chain code. For blobs, we make a clockwise tour of the boundary. Strokes, we consider as one-dimensional objects and make a lengthwise tour essentially along their spine (not a circular tour, forward on one side and back on the other side). To do this, we assign arbitrarily a direction to each stroke. For bounding strokes, we consider the region they bound and make a clockwise tour of the boundary, as for blobs. Along these tours, we do not give a 'turtle-graphic' way of tracing the curve but rather we break the curve up at curvature maxima, points of inflexion, singularities such as cusps, and T-, X-, fork and arrow junctions with other strokes. The pieces of the curve between such distinguished points we call segments which are also assigned types such as line, arc, loop, etc. In addition, we have a segment type called a virtual line which either marks the location at which a component is connected to one of its neighbors in a compound shape or marks a gap between the terminators of a bounding stroke which must be filled to complete the boundary. Segments are also given locations which may be one or more of the 9 coordinates on the chart, directions which may be one or more of the 8 cardinal directions and curvature from the set (+,0,-).

E. Primitive Types: At the base of the whole data structure, note that there is a small set of undefined types for points, segments and components. These receive a procedural definition through the programs that 'parse' images into shape descriptors, that reproduce an image from a shape descriptor, that merge two overlapping shape descriptors and that transform a shape descriptor to allow for changes caused by arbitrary cut-offs and thresholds in the parsing program. (The last is central part of the matching process to be described in a later article.) There are many non-trivial choices to make in deciding on a particular list of primitive types. For example, hidden lines, axes of symmetry or special alignments of features are incorporated as special types of virtual lines. As another example, motivated by optical illusions which produce the appearance of spirals that are not actually present, we decided to parse a spiral by inserting a virtual line that makes its outer loop into a bounding stroke, and makes the inner loops into lower nodes (see Figure 6). The examples in the appendices will illustrate the primitive types that seem most correct to us now: but this part of the structure may well be modified by experience.

F. Links: After all this structure, there still seem to be further perceptually prominent aspects of shapes that have to be incorporated. One of these is symmetry. A part or whole of the shape may be bilaterally or centrally symmetric, or symmetric under a rotation through an angle $2\pi/n$, for some $n \geq 3$. Or a partial symmetry may exist which interchanges the parts of the figure given by nodes (e.g. a detail in one part of the shape is the mirror-image of a detail elsewhere in the shape). Another situation is the presence of prominent parallel lines in different parts of the shape. A more mundane type of link is the one that indicates the identification of points and segments in different nodes, thereby telling us how to glue components together or glue details into the whole. Many of these links are illustrated in the examples in the appendices.

POSSIBILITY I
PRIMITIVE STROKE FROM
x TO y

POSSIBILITY II
HIGHER NODE = BOUNDING STROKE
FROM x TO z
(INCLUDING VIRTUAL LINE m);
LOWER NODE = ARC FROM z TO y

FIGURE 6

References

Agin, G., Hierarchical representation of three-dimensional objects using verbal models, IEEE Trans. on Pattern Anal. Mach. Intell., 3, pp.197-204, (1981).

Binford, T., Visual perception by computer, Proc. IEEE Conf. on Systems and Control, (1971).

Blum, H., Biological shape and visual science, J. Theoretical Bio., 38, pp.205-287 (1973).

Brady, M., Criteria for representation of shape, in **Human and Machine Vision** (ed. J. Beck et al), Academic Press, pp.39-84 (1983).

Davis, E., The Mercator representation of spatial knowledge, IJCAI-1983, pp.295-301.

deKleer, J. & Brown, J.S., The origin, form and logic of qualitative physical laws, IJCAI-1983, pp.1158-1169 (1983).

Duda, R. & Hart, P., **Pattern classification and scene analysis**, Wiley (1973).

Freeman, H., On the encoding of arbitrary geometric configurations, IEEE Trans. Electronic Computing, 10, pp.260-268 (1961).

Fu, K.S., **Syntactic pattern recognition and applications**, Prentice-Hall (1982)

Hayes, P., The naive physics manifesto, in **Expert systems in the electronic age** (ed. D. Mitchie), Edinburgh Univ. Press, pp.242-270 (1979).

Hu, M., Visual pattern recognition by moment invariants, IRE Trans. on Inform. Theory, IT-8, pp.179-187 (1962).

Kosslyn, S., **Image and mind**, Harvard Univ. Press (1980)

Ledley, R.S., Film input to computer and associated syntax-directed pattern recognition, in **Optical and Electro-optical information processing**, (ed. J.L.Tippet et al) MIT Press, pp.591-613 (1965).

Leeuwenberg, E., A perceptual coding language for auditory and visual patterns, Am.J.of Psych., 84, pp.307-349,(1971).

Marr, D., **Vision: A Computational Investigation into the Human Representation and Processing of Visual Information**, W.H. Freeman and Company (1982).

Marr, D. & Nishihara, K., Representation and recognition of the spatial organization of three-dimensional shapes, Proc. Royal Soc. London, B200, pp.269-294 (1978).

Minsky, M., A framework for representing knowledge, in **Psychology of vision**, (ed. P. Winston), McGraw-Hill (1975).

Pavlidis, T., Analysis of set patterns, Pattern recognition, 1, pp.165-178 (1968).

Pavlidis, T. & Ali, A hierarchical syntactic shape analyzer, IEEE Trans. on Pattern Anal. Mach. Intell., pp. 2-9 (1979).

Rosch, E., Principles of categorization, in **Cognition and Categorization**, (ed. E.Rosch and B.Lloyd), L.Erlbaum, pp.28-49,(1978).

Shapiro, L., IEEE Trans. on Pattern Anal. Mach. Intell., 2, (1980).

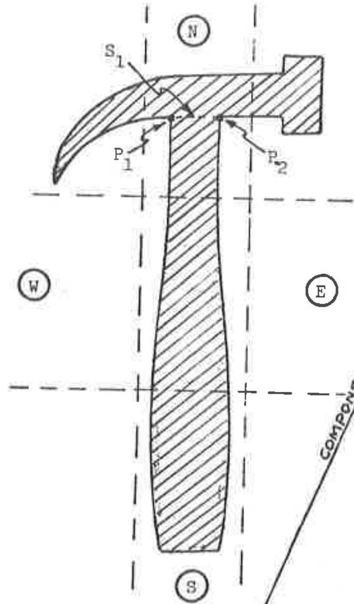
Shapiro, L., Mulgaonkar, P., Moriarty, J., & Haralick, R., A generalized blob model for 3-dimensional object description, 2nd IEEE workshop on picture description, (1980).

Shaw, A.C., The formal picture description scheme as a basis for picture recognition, Information and Control, 14, pp.9-52 (1969).

Simon, H., Complexity and the representation of patterned sequences of symbols, Psych. Review, 79, pp.369-382 (1972).

Smith, E. & Medin, D., **Categories and Concepts**, Harvard Univ. Press (1981).

APPENDIX A: THE SHAPE DESCRIPTOR OF A PARTICULAR HAMMAR



NODE 1: COMPOUND, 2:1

POINTS:

P_1:CONCAVE CORNER,LOC N

P_2:CONCAVE CORNER,LOC N

SEGMENTS:

S_1:INTERNAL VIRTUAL LINE,LOC N,DIR E

COMPONENTS:

NODE 2:

(LOC N,SCALE 1:2,DIR E)

LINKS:$P_1,S_1,P_2 \rightarrow P_1,S_4,P_5$

NODE 4:

(LOC C,SCALE 1:1,DIR N)

LINKS:$P_1,S_1,P_2 \rightarrow P_1,S_1,P_2$

COMPONENT

COMPONENT

NODE 2: BLOB, 2:1

POINTS:

P_1:STRAIGHT,LOC C

P_2:CUSP,LOC SE

P_3:CONCAVE CORNER,LOC NW

ETC.

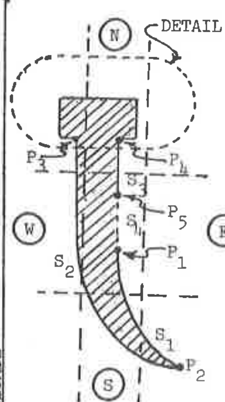
DETAIL:

NODE 3:

(LOC N,SCALE 1:4,DIR N)

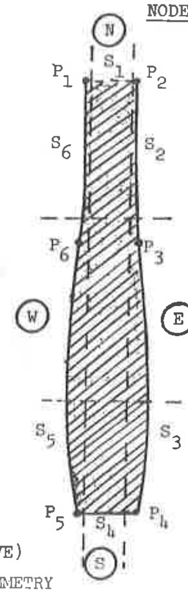
LINKS:$P_3,P_4 \rightarrow P_5,P_4$

(SEGMENTS AS ABOVE)



DETAIL

NODE 4: BLOB, 4:1



POINTS:

P_1:CONVEX CORNER,LOC NW

P_2:CONVEX CORNER,LOC NE

ETC.

SEGMENTS:

S_1:INT. VIRT. LINE,LOC N,DIR E

S_2:ARC,LOC NE,DIR S,CURV -

S_3:ARC,LOC (E,SE),DIR S,CURV +

ETC.

TOUR:

$(P_1,S_1,P_2,\ldots,P_6,S_6,P_1)$

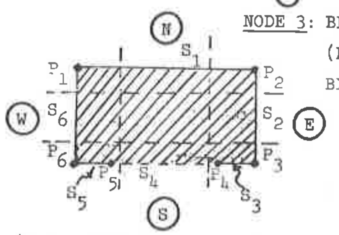
BILATERAL SYMMETRY:

(P_1,P_2)

(S_2,S_6)

(P_3,P_6)

ETC.



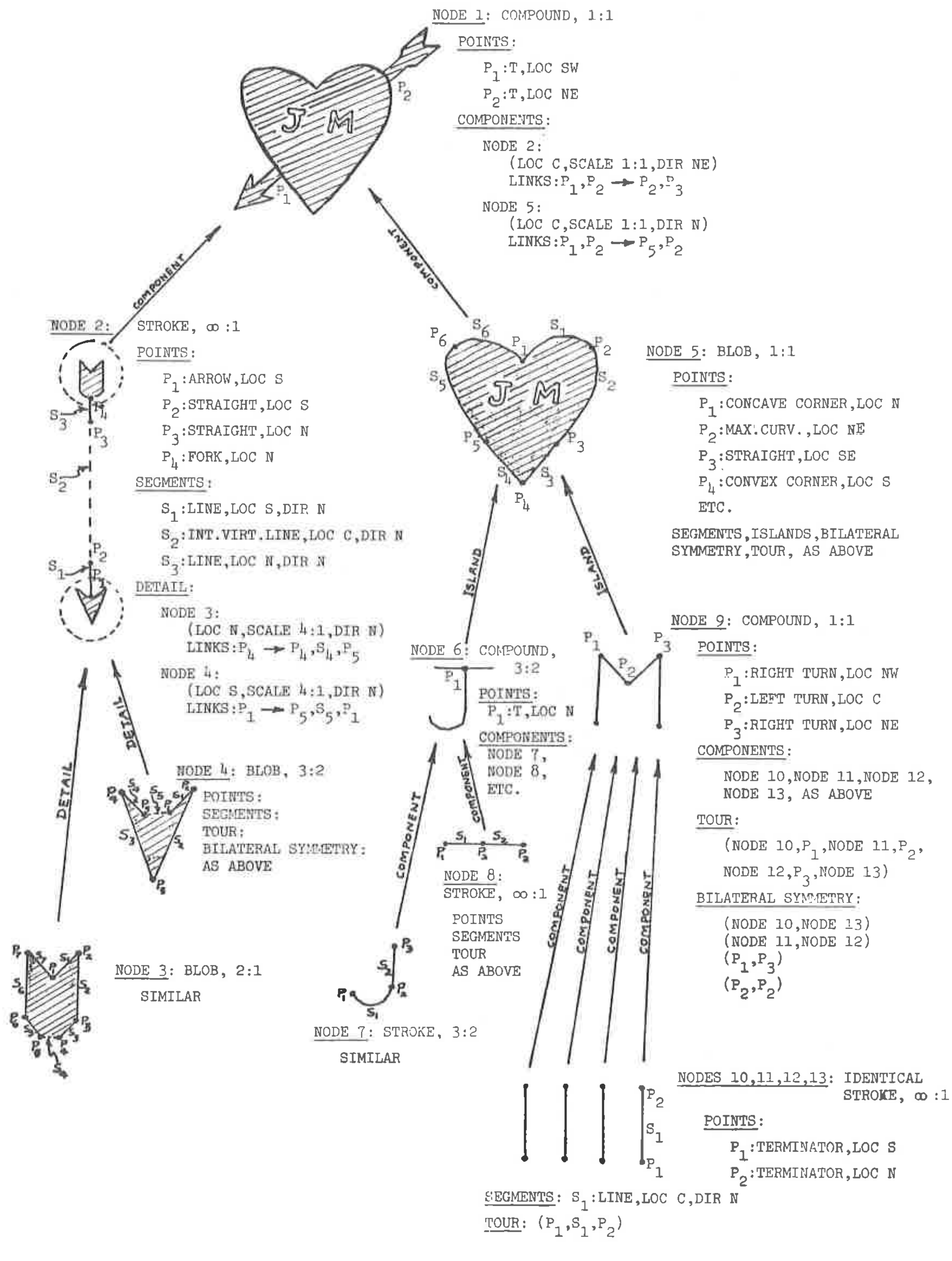
NODE 3: BLOB, 1:2

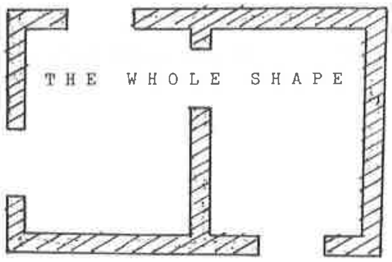
(POINTS, SEGMENTS AS ABOVE)

BILATERAL AND CENTRAL SYMMETRY

(N.B. The pictures are for human use only! They are not part of data structure)

APPENDIX B: OUTLINE SHAPE DESCRIPTOR OF CARVING IN TREE





APPENDIX C: THE SHAPE DESCRIPTOR OF A HOUSE-PLAN

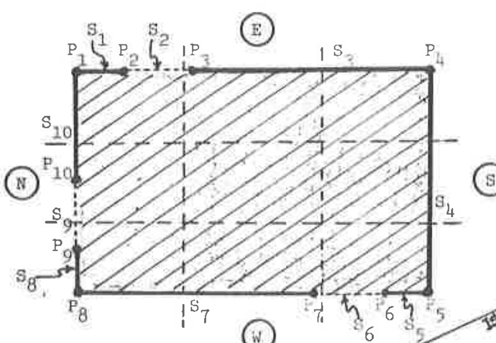
NODE 1: BLOB, 3:2

POINTS:

P_1:RIGHT CONVEX CORNER,LOC NE
P_2:STRAIGHT,LOC NE
P_3:STRAIGHT,LOC E
P_4:RIGHT CONVEX CORNER,LOC SE
P_5:RIGHT CONVEX CORNER,LOC SW
P_6:STRAIGHT,LOC SW
P_7:STRAIGHT,LOC W
P_8:RIGHT CONVEX CORNER,LOC NW
P_9:STRAIGHT,LOC NW
P_{10}:STRAIGHT,LOC N

SEGMENTS:

S_1:LINE,LOC NE,DIR S
S_2:EXT.VIRT.LINE,LOC NE,DIR S
S_3:LINE,LOC (E,SE),DIR S
S_4:LINE,LOC S,DIR W
S_5:LINE,LOC SW,DIR N
S_6:EXT.VIRT.LINE,LOC SW,DIR N
S_7:LINE,LOC (W,NW),DIR N
S_8:LINE,LOC NW,DIR E
S_9:EXT.VIRT.LINE,LOC (NW,N),DIR E
S_{10}:LINE,LOC (N,NE),DIR E



TOUR: $(P_1,S_1,P_2,S_2,P_3,S_3,\ldots,P_9,S_9,P_{10},S_{10},P_1)$

ISLANDS:

NODE 2:
(LOC N,SCALE 2:1,DIR W)
LINKS: $P_2,S_2,P_3,P_9,S_9,P_{10} \rightarrow P_2,S_2,P_3,P_9,S_9,P_{10}$

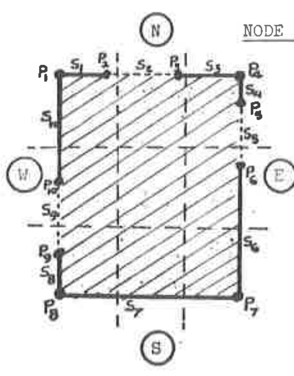
NODE 3:
(LOC S,SCALE 2:1,DIR W)
LINKS: $P_6,S_6,P_7 \rightarrow P_4,S_4,P_5$

LINKS:

$(P_5,S_5,P_6)_{\text{node 2}} \rightarrow (P_8,S_7,P_7)_{\text{node 3}}$

ISLAND

ISLAND



NODE 2: BLOB, 3:2

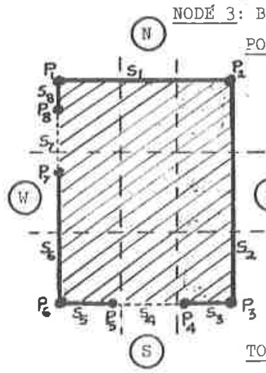
POINTS:

P_1:RIGHT CONVEX CORNER,LOC NW
P_2:STRAIGHT,LOC NW
P_3:STRAIGHT,LOC N
P_4:RIGHT CONVEX CORNER,LOC NE
ETC.

SEGMENTS:

S_1:LINE,LOC NW,DIR E
S_2:INT.VIRT.LINE,LOC N,DIR E
S_3:LINE,LOC NE,DIR E

TOUR:$(P_1,S_1,P_2,\ldots,P_{10},S_{10},P_1)$



NODE 3: BLOB, 3:2

POINTS:

P_1:RIGHT CONVEX CORNER,LOC NW
P_2:RIGHT CONVEX CORNER,LOC NE
P_3:RIGHT CONVEX CORNER,LOC SE
ETC.

SEGMENTS:

S_1:LINE,LOC N,DIR E
S_2:LINE,LOC E,DIR S
S_3:LINE,LOC SE,DIR W
ETC.

TOUR:$(P_1,S_1,P_2,\ldots,S_7,P_8,S_8,P_1)$