STRUCTURED DESCRIPTIONS OF COMPLEX OBJECTS*

by

Kamakant Nevatia
and
Thomas 0. Binford

Stanford Artificial Intelligence Laboratory
Computer Science Department
Stantord, California

ABSTRACT

Experimental techniques are demonstrated which generate segmented symbolic descriptions for complex objects with joints, such as a hammer or a glove. Complete "descriptions with relationship of parts at joints and descriptions of joints are presented. These techniques are elements of a larger scheme for description mechanisms for hypotheses, and for visual memory and recognition.

Index Words: Shape, complex objects, curved objects, recognition, perception, scene analysis, part/whole segmentation, joints.

1. Introduction

We describe experiments with symbolic description of curved objects. Tn previous work1,2 descriptions were generated for the pieces of objects, according to a volume representation of shape3. The pieces are an adequate description only of objecls with a single part, like a torus or a snake. We have generalized and improved techniques for descriptions of pieces. We have made complete descriptions which join the scattered pieces into a whole. These descriptions are the basis for recognition, and a few examples are discussed.

The motivations for studying representation of shape are: for visual systems of robots which use depth or image data; for the interactive programming of assembly robots for industrial tasks; for the computerization of industry-machining, parts description, systemstization of assembly operations; for display; and for its relevance to other areas of A.J. and biology. Our concern is implementation of symbolic descriptions oi shape and space which are adequate for integrated robot systems. We do not think a single representation is adequate throughout a system, but we have attempted to analyze tasks in terms of computational primitives originating from topological and geometrical primitives, and design representation abilities accordingly. Representative- tasks are manipulation, display, and recognition. Manipulation requires mass and momenta of articulated parts, and ability to calculate overall moments in various positions and orientations. Display requires description of Surface properties and two dimensional proximity, to find overlapping surfaces tn the image. Recognition requires visual memory for generation of hypotheses, prediction and verification facilities, geometric relations among parts, and a good choice of parts.

2. Representation and Models

The representations-' depend upon segmentation of complex objects into parts. Parts can themselves be composed of subparts. The value of the part/whole segmentation depends on a useful representation for primitive parts. The basic topological operations of cutting and pasting are used in joining parts. The representation oi parts and joints amounts to the basis tor an intelligent guess about structure. Special knowledge about joints, e.g. the anatomy of humans, is much more powerful.

Primitive parts arc described by volume representations. The primitive parts are arm-like pieces which are described as "generalized local cones" by localizing and generalizing translational invarianco--. These local coties are the volumes swept out by translating an arbitrary cross section, maintaining it normal to the path along which it is translated, while the scale of the cross section is changed smoothly. More generally, we have locally snake-like and locally screw-like shapes. The basis of this representation is transformation ot local descriptions. The typical element is a snake-like piece described by an axis fwhich is a space curve) and arbitrary normal cross-section valued functions .

We have chosen a high level representation to interface with heuristics rather than a low-level representation, which is directly calculable Jrom data, such as the Fourier Transform. We feel that heuristics should be at as high a level as possible and wish to provide *a* language for their expression.

A representation must have computational equivalents. In the following, we demonstrate computational equivalents for the treatment of generalized cross-section and axis determination.

3'. Data Acquisition and description of Hardware

We derive three-dimensional information about an object, by a laser triangulation ranging system. The details of the operation, construction and calibration of this apparatus are tully described by Agin1,2, who built this system. We will include only those details necessary for our discussion here.

A sequence of parallel planes of light generated by a laser beam are cast on the scene viewed by a TV camera. An interference filter allows the camera to see only the laser light. Each TV frame shows a space curve corresponding to one plane of light. Depth discontinuities in space appear as discontinuities in image. A second sequence of scans orthogonal to the first is useful because the first sequence gives only crude information about boundaries parallel to the planes. The initial data consists of two intersecting sets of curves tn the TV Image (Fig. 1). Any point among these curves can be mapped to 3-D coorindar.es.

Boundary Organization

A boundary for the object can be defined by joining the end points of the segments from the two sets

---

* This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. SD-IBJ.

of scans in a certain coherent fashion. There are several reasons for constructing this boundary. It helps link the two sets of scans, and provides an ordering on the segments that is useful in axis finding. It provides a sense of neighborhood that prevents the routines which extend pieces from crossing boundaries as discussed in a later section. Knowledge of the boundary helps in keeping track of the areas already described by associating them with corresponding parts of the boundary. There is also considerable psychological evidence to indicate that humans make extensive use of boundaries in their perception of visual scenes.

To construct the boundary, we need to order the end points of the segments linearly. Ordering the points by a nearest neighbor approach does not yield the right boundaries.

The boundary should not cross a solid part of the body, i.e. through any scan segments. Consider successive scan segments S1 and S2 (See Fig. 2). Where there is a cross scan segment On which interesects both S1 and S2, the two segments are connected. The boundary cannot cross the cross segment Cn and must turn hack. The boundary extends from one end of S1 through either the first cross scan C1, if it terminates between S1 and S2, or to the end of S2. The process is continued in the same fashion. We obtain one or more closed boundaries outlining the object. The boundaries obtained for some objects are shown in Figs. 3 and 4.

Boundary linking requires calculating the intersections of segments from the two crossed scans. We need calculate only a few of the possible intersections near the ends of segments. Given a segment S1, we can calculate which cross scan angles contain segments which intersect S1 (by calculating the angles of endpoints of S1 from the laser viewpoint, and obtaining cross scans in that range of angles). The intersection of two segments is determined by making piecewise linear approximations to the two segments. A few minor errors and extra effort are caused by slight errors in the intersection process and by calibration uncertainties which give small angle errors in choosing cross scans.

An alternative approach would be to use a large array, where each byte of the array corresponds to a position In the Image plane. For each point that belongs to some segment in one scan orientation mark the corresponding byte in the array by this segment number.

Now, for each point that belongs to some segment, 5?, in the cross scan check whether the corresponding byte in the memory is marked. If so, then the segment S?, Intersects with the marked segment in the memory at this point. This method will give us all intersections without searching. The obvious disadvantage is the requirement of a large memory. The time required will be proportional to the boundary length.

I4 . Description of Primitives. Axis Determination

We wish to describe a complex object in terms of components which are simple to describe, and the relationships of these parts. We thus need to segment a given object into simpler parts. We describe primitives by an axis, which is an arbitrary space curve, and normal cross-sections along this axis. Our basic criteria for accepting a piece as simple are that the direction of the axis be continuous and that the cross-section function along the axis be continuous. We look for portions with large changes in the cross-section, and consider them as likely places for segmenting the object. The segmentation process must be flexible and able to generate alternate propositions. A higher level routine could sometimes guide this process.

Preliminary Segmentation

We derive a preliminary segmentation, based on the descriptions of scan segments. We look for a group of segments, that are adjacent and have continuously varying lengths. Adjacency is determined by their proximity in the boundary list. We use the length of a segment, to check for the continuity of cross-sections. However other measures, such as moments of the curves describing the segments, could be used as descriptors of the cross-sections. A number of segments so linked together is called a "group". Note, that a given part of the object may be included in more than one group, corresponding to different directions of the scans. The groups are generated for the two sets of segments obtained earlier. However, some parts of the object, are perhaps not well described in these directions. We generate synthetic segments in other directions, by computing intersections of lines of a certain orientation with the boundary. These new segments are grouped in a similar fashion. Preliminary segmentation is intimately connected with description of pieces and should be guided from higher levels .

The initial groups are treated only as starting places. In the process of description, we may extend a group or break up a group. Also, groups may be suggested after we have removed some parts that have been described well.

Piece Description

The piece we wish to describe is given by a group of segments, and the corresponding boundary. We wish to find an axis, cross-section description for this piece. The constraints for the description are the following:

1. The cross-sections must be normal to the local axis .

2. The axis must pass through corresponding points of the cross-sections. In the 2-D version, the corresponding points are taken to be mid-way between the end points on the boundary.

An initial guess is obtained by taking the mid-points of the segments in the preliminary group. We then construct cross-sections normal to the axis at these points and compute their intersections with the boundary. A new axis is defined by the mid-points of these intersections. If the distance of the new axis points from the old axis points is not sufficiently small, we iterate this process until it converges. Cross-sections may be in either 2-D or 3-D coordinates.

This process usually converges for pieces that are part of well defined cylinders. Parts of the object where convergence fails must be described by an alternate group or after interfering pieces have been explained and removed.

An alternative to iteration is to find a best cone that fits the given boundary segments. The cone Is a generalized cone, with arbitrary axis and cross-section function. When looking at a small part of the piece, we constrain the axis to be a straight line or a parabola. The cross-section function is limited to be linear. A best fit in the least squares sense, is found with these constraints, giving us the axis and cross-sections directly. The solution is particularly simple for the case of a straight line axis, and may be found without iteration.

Extensions of a Piece

Once an axis, cross-section description of a part is found, we see whether this description extends in a continuous manner over a larger part of the body. We extrapolate the axis at either end by a small distance. A normal cross-section to the axis is constructed at this point and its intersections with the boundary are computed, A test is made to see whether this cross-section satisfies the constraints noted above, by computing its mid-point and comparing the distance of the mid-point from the extrapolated axis. If the distance is sufficiently small, we accept the new cross-section and take its mid-point as the new point along the axis- and attempt to extend further. If not, then we make a modified guess at the extrapolated axis, by including the newly found point. We recompute the normal cross-section and repeat the above test. This allows us to trace the axis for a smoothly curving object. We have not found it necessary to iterate in this phase of extension. The criteria for terminating the extension process are:

1. When no intersection can be found with the neighboring boundary. This usually occurs because of a sharp change in the slope of the boundary, and agrees with our intuitive sense of a proper segmentation point, or at the end of an object.

2. *It* the length (radius) of the new cross-section is very different from the lengths of previous cross-sections .

When the extension process terminates, a check is made to see if a termination for the piece has been reached. We check whether the unexplained piece of boundary between the two segments on the sides of the piece is largely contained in a small extension of the piece near the terminating end. In 3-D processing, normal terminations would be detected in a very natural way. Special tests can be made to check for spherical or oblique terminations. For examples of piece descriptions see Figs. 5. 6 and 7.

Summary descriptors for the piece such as the length of its axis, the ratio of this length to the average width of the cross-sections, and curve fit descriptions of the axis direction and cross-section function are determined. These include descriptors such as straight or parabolic for the axis and constant, linear etc, for the cross-section function.

This method of deriving piece descriptions is similar to that described by Agin. The important differences are in our use of the boundary for finding new cross-sections, requiring significantly less computation than his scheme, which involves computing intersections with many segments. Also, the process of extension is more structured, as it follows the boundary, and spurious extensions in distant parts of the bodies are not made. Agin's procedures did not use the boundary, and when attempting to extend would sometimes make extensions crossing the boundary, e.g. some extensions would extend across two fingers in a glove. We are able to test for terminations. Also we make no assumptions about the shape of the cross-sections (Agin assumes circular cross-sections).

Extensions are found for all groups found by initial segmentation. Thus many parte of the body will be included in more than one description. This is not a disadvantage, but a crucial advantage; it allows us to compare alternatives and choose on the basis of some global context.

Generalized Cross-Sections

We have not made use of the shape of the cross-sections. We see only a small part of them. We have little information on the cross-sections, particularly for small objects, due to ranging errors. This makes their detailed description difficult. However, we can make crude Judgements, e.g. that the palm of a glove is not circular. The continuity of the pieces can be checked by applying translational invariance criterion to these cross-sections, either directly on a point by point basis or by comparing their descriptors. These descriptors may be some moments of the cross-section[4] or expansions in an orthogonal series. We can use the continuity of cross-sections in preliminary segmentation as well as verification of piece extensions.

Other interesting descriptions are the directions of minimum and maximum curvature. Minimum curvature direction is the direction of axis for a circular snake. However, they have no direct relation to the axes directions for many shapes. Also, these directions are sensitive to noise in the data and hence ill-defined.

5. Linking of Pieces

Local description of pieces generates redundant piece descriptions. Consider, for example, a rectangle. We may generate one description of an axis along the length, with terminations at both ends. We may also generate other descriptions near the corners, describing the piece as part of a cone.

Each overlapping pair of pieces is compared. Parts of piecel that are not covered by piece;-' are computed. The areas of these parts and the distances of their constituent points from the common edge are computed. If the area of the uncovered part is relatively small compared to the area of the piece involved, and all points are close to the common boundary, then piecel is assumed to be totally contained in piece 2. All pieces that are completely contained in some piece are removed from further consideration. As example: two pieces were found describing the handle of the hammer in Fig. 5. These overlap each other completely and only one is retained.

Connectivity of Joints

We cut off those parts of the body that have been described by the pieces. We merely follow the boundary and eliminate the segments belonging to any piece. We thus get a boundary for the remaining part of the body of the object. The remaining parts may be disjoint. In this case, we have more than one joint. We will call the remaining parts as joint pieces. With each joint piece, is associated the order o£ pieces that were joined to it. Examples of these are shown in Figs. 5,6, and 7.

Other Joint Descriptors

We wish to characterize these Joints further. When a large joint piece is left, it may be sent to the axis-finding routines and perhaps a useful description obtained for it.

Tests are made to examine whether two pieces are collinear, i.e. could they be continuous extensions of each other. This requires that their respective axes be nearly parallel at the near ends, cross-sections be continuous, and they have a small continuous segment of the boundary connecting them. Alternatively their respective areas must be close in 2-D. That is, we check for either boundary proximity or distance proximity. It may be noted that distance proximity does not always Imply boundary proximity.

If the joint is largely covered by an extension of the pieces, then we call it a point-type. Such is the case in the example of the hammer. It may be further described by the angle relationships of its members. For the hammer two pieces are nearly collinear and the third piece is nearly normal to these two. Thus it is described as a T-joint.

The example of the glove, shows a different structure. The area described by joining the ends of the four fingers is very small. Also the amount ot unexplained boundary between them *coriBlst of* several very short segments. This we describe as a Fork. We check for planarlty of the area these fingers join to. We call this a planar fork joint.

Other descriptions include: the number of limbs at a joint, their proportions, detection of horizontal or vertical members, identical components, designation of the dominant *pieces by* their size, and *axes of* bilateral symmetry if any. Not all of these descriptions are made initially. Some are reevaluated in the course of a match.

6. Symbolic Data Structure

The descriptions are converted to a symbolic data structure, using the facilities *of* LEAP[5]. LEAP allows associations and retrievals of the form

attribute ® object = value.

We will describe the symbolic structure derived for the hammer.

Description of joints

```
JOINTS ⊗ OBJECT ≐ {JOINT1}
PIECES ⊗ JOINT1 ≅ {PC1, PC2, PC3}
TYPE ⊗ JOINT1 ≡ {THREE_JOINT, T , POINT_JOINT}
RELATIONS ⊗ JOINT1≡{COLLINEAR⊗PC1 ≡ PC2,
  NORMAL ⊗ PC3≡{PC1,PC2}}
```

Description of pieces

```
AXIS ⊗ PCI ≅ {AXIS1}
XSECT ⊗ PC1 ≡ {XSECT1}
```

AXISI is a descriptor for a list of axis-points, tangents to the axis at these points and their curve-fit description. XSECT1 is a descriptor for a list of cross-sections and a curve-fit description of the cross-section function.

7- Recognition

Recognition consists of matching the symbolic descriptions for the current scene, with some descriptions stored in memory. Whether we match against models input by hand or against previous descriptions, the problems are the same. We seek a best match such that the current description is most likely to be a description of the matched model. We cannot expect to find an exact match of two descriptions since the objects may be viewed from different angles, and will have different degrees of self-occlusion. Our descriptive mechanisms allow a number of alternate descriptions to be made for the same scene. We may expect some pieces at some Joints to be missing and some extra joints created. We also allow for articulation of the pieces of an object.

The matching problem can be cast in graph-theoretic terms. Descriptions may be viewed as graphs with pieces and joints as nodes. The relations of these pieces and joints are the arcs of the graph. We also have properties associated with these nodes, describing the individual nodes. We wish to find a matching of two such graphs. The graphs are not expected to be identical and we wish to *evaluate* the transformation required to map one graph into the other. Barrow[6,7] et. al. discuss some approaches to a partial graph matching problem of this type. One of these methods tries to find a maximal self-consistent set of assignments from one graph structure to the other. However, regarding the problem as a graph matching problem ignores important semantics.

Our matching scheme first attempts to find a set of best matchings of joints satisfying piece connectivity relations. The number of Joints in a typical scene is rather small (3 or 4), allowing for a fairly complete search for joint assignments. Moreover, the joints can be ordered in a heirarchy by the size of their dominant pieces. We need only match the joints *at* the same level. As example; consider *a* humanoid figure with joints at

a) the hips (joining the legs to the body),

b) the neck and shoulders (joining the neck and the arms to the body) and

c) the *two* joints attaching the two hands *to* the two arms.

Here the neck joint is directly connected to the hip joint through the body and also *to the* two hand joints through the two arms. The other joints have no direct connections with one another. Consider trying to match this description with another one like it. We wish *to consider* matching any joints in *one* description with any other in the second. However, it is clear that if we match the neck joint against the neck joint in the other description, the connectivity relations of the *two* descriptions will be most similar. There is still ambiguity about which hand joint should match a given hand joint in the other description. In other examples we may have more such ambiguities. Nonetheless, the number of alternatives to be evaluated more fully is drastically reduced. For this example we could also have ordered the joints in a heirarchy by the size of their constituent pieces. Thus the neck and the hip joints would be of a different level then the hand joints. We only need to consider the matches between two joints of the same level, further reducing the size of the search space.

We evaluate the total quality of a match by comparing the quality of matches of Joints and their pieces in the above assignments. The piece matches are evaluated by the similarity of their descriptors (such as relative sizes). We sometimes need to evaluate alternate descriptions of a scene. The model with the best match is computed, and if a sufficiently good match is found, a recognition is claimed.

Our recognition effort is at a preliminary stage. Soon, we hope to be able to recognize objects of the complexity of a toy horse and doll as distinct from one another and also to recognize the same object from different orientations and different articulations of its limbs.

ACKNOWLEDGEMENTS

We would like to express our thanks to G.J. Agin for the use of his laser ranging programs and picture files.

REFERENCES

1. Agin, G.J., "Representation and Description of Curved Objects", Stanford Artificial Intelligence Laboratory Memo AIM-I73, October, 197?.

P. Agin, G.J., and Binford, T.O., "Computer Description of Curved Objects", To be presented at the Third IJCAI, August, 1975.

3. Binford, T.O., "Visual Perception by Computer", presented at the IEEE Conference on Systems and Control, Miami, December, 1971.

I4 . Alt, F.L., "Digital Pattern Recognition by Moments", Journal of ACM, February, 1972, pp. 240-258

5. Feldman, J.A., and Rovner, P.D., An Algol-based Associative Language, Comm. ACM, August, 1969. pp. 434-449.

6. Barrow, H.G., Ambler, A.P., and Burstall, R.M., "Some Techniques for Recognizing Structures in Pictures", Int. Conf. on Frontiers of Pattern Recognition, Honolulu, Hawaii, January, 1971.

7. Ambler, A.P., Barrow, H.G. , Brown, CM., Burstall, R.M., and Popplestone, R.J., "A Versatile Computer-Controlled Asaemly System", Department of Machine Intelligence, University of Edinburgh. (Draft Report, February, 1973).

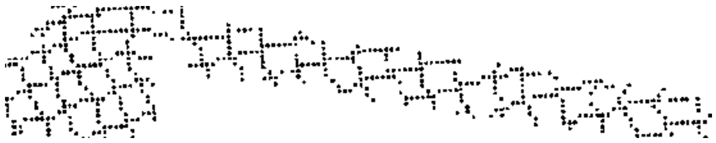

Fig. 1. Laser Scans For a Hammer.

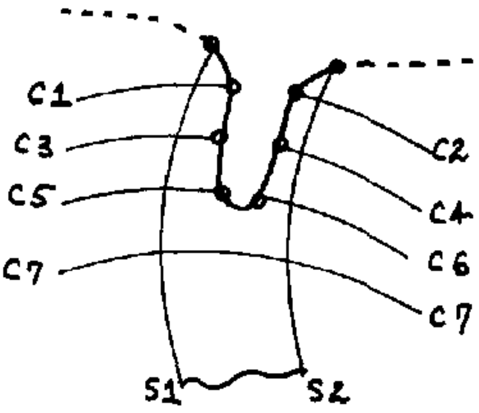

Fig. 2. Linking of Cross-scans in a Boundary.

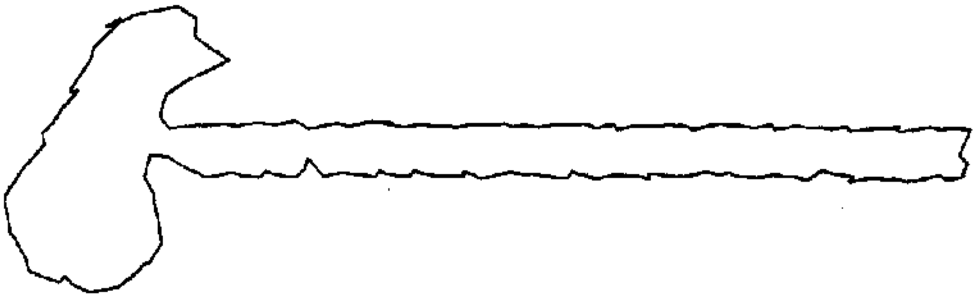

Fig. 3. Derived Boundary of a Hammer.

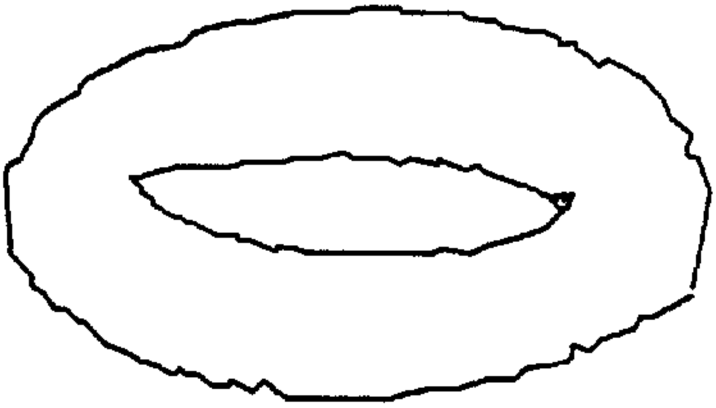

Fig. 4. Derived Boundary of an Object with a Hole.

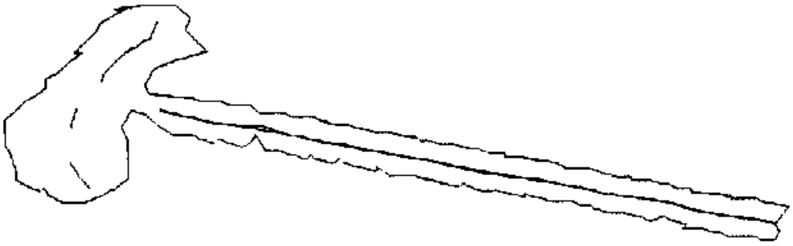

Fig. 5. Piece Descriptions of a Hammer.
(Note multiple descriptions)

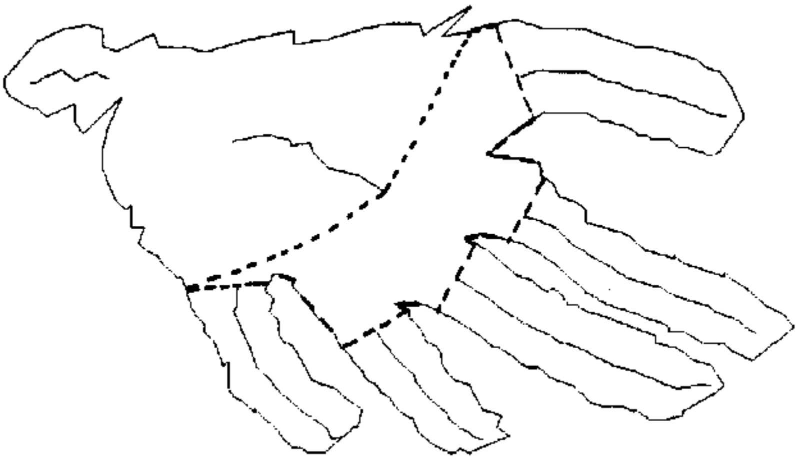

Fig. 6. Piece Descriptions of a Glove.
The dotted lines indicate a joint area.

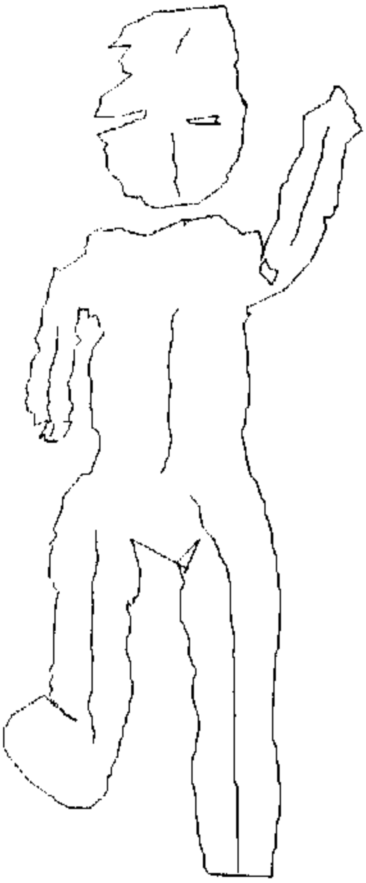

Fig. 7. Piece Descriptions of a Doll.