

Description and Recognition of Curved Objects¹

Ramakant Nevatia

*Image Processing Institute, University of Southern California,
Los Angeles, CA 90007, U.S.A.*

Thomas O. Binford

*Artificial Intelligence Laboratory, Stanford University,
Stanford, CA 94305, U.S.A.*

Recommended by Raj Reddy

ABSTRACT

Analysis of scenes of three-dimensional objects has, in the past, been largely limited to the world of polyhedra. Techniques for generating structured, symbolic descriptions of complex curved objects by segmenting them into simpler sub-parts are presented here. The complexity of objects used is that of toy animals and hand tools. Recognition is performed by matching these descriptions with stored descriptions of models. A laser ranging technique is used to acquire three-dimensional position of points on the visible surfaces. Successful segmentation and recognition results have been obtained for scenes with multiple, occluding objects in various orientations and with a variety of articulations of sub-parts.

1. Introduction

This paper describes development of techniques for a machine to analyze scenes of complex curved objects, with the goals of generating useful symbolic descriptions and recognizing the objects. Analysis of such scenes has been largely ignored in previous work.

The techniques of classical pattern recognition are primarily concerned with the classification of two-dimensional images. These techniques do not use segmentation; features, such as moments or coefficients of expansion in some orthogonal series are derived for a whole image. Another popular method is that of template matching, by point to point correlation of a model pattern with the image pattern. These techniques are inadequate for three-dimensional scene analysis for many reasons, such as occlusion, changes in viewing angle, and articulation of parts.

¹ The research for this paper was supported by the Advanced Research Projects Agency of the Department of Defense under contracts DAHC-15-73-C-0435 and F0606-72-C-0008.

The patterns at image level are not invariant and more useful symbolic descriptions must be derived from the images.

Previous work in three-dimensional scene analysis has concentrated on scenes containing polyhedral objects only [8, 9, 12, 14]. The techniques used there have not generalized to scenes of wider classes of objects. The basic approach used in our work is that of describing objects by segmenting into simpler sub-parts. Symbolic descriptions of these sub-parts and their relationships are compared with stored models for recognition of objects. The complexity of objects considered is typified by toy animals such as a horse and a doll, and by hand tools such as a hammer.

Computer programs for description and recognition of objects have been written and successfully applied to a number of scenes. Recognition programs are able to recognize an object as the same with its limbs articulated. Also, our system has a visual memory with a limited *indexing* structure, i.e. a subset of similar objects can be retrieved from the memory using brief descriptions of the current objects. Models in the memory are obtained by storing previous machine generated descriptions of the objects.

The chosen paradigm is that of generating descriptions and then matching them for recognition, as opposed to the alternatives of using "high level", prototype specific knowledge for guidance from the very beginning. We believe that substantial "low level" analysis of an image is necessary for situations where the class of objects is large and not tightly determined from context.

This work is based on initial work of Binford and Agin [1, 2, 5], on the description of curved objects. The same representation is used here, but more powerful and complete description techniques have been developed and recognition of objects accomplished.

The remainder of this paper is organized as follows. Section 2 describes the chosen representation for describing the objects, Section 3 contains the techniques for segmentation of objects in a given scene into desired parts so that useful symbolic descriptions (in context of the chosen representation) are generated, as discussed in Section 4. Section 5 describes techniques for matching the generated symbolic descriptions against models stored in memory for recognition. A memory indexing scheme is described at the end of this section. Section 6 summarizes the performance results obtained from the computer analysis of a number of actual scenes.

This paper is abstracted from a Ph.D. thesis and many details, particularly the specifics of some algorithms, are omitted here. These details may be found in [11].

2. Representation

Only the shape properties of an object are used in this work. The concept of shape has an intuitive meaning, but is difficult to define precisely. We choose to represent the shape of a complex object by segmentation into parts. Each part may be further

segmented into simpler sub-parts. For example, a human shape is represented as consisting of two legs attached to one end of a body and the two arms and the head attached to the other end. An arm may be further segmented into a hand, a forearm, etc.

Segmentation into sub-parts allows incremental changes of an object to be described incrementally, and permits useful ways of describing similarities as well as differences between two shapes. Articulation of limbs is represented in a natural way. Segmentation into volume primitives is preferred to that into surface primitives, as the surface discontinuities often do not correspond to the intuitive notions of boundaries of parts of an object.

The connectivity relations of the segmented sub-parts constitute what we call the structure of the object and is the major component of its shape description. This structure is invariant with viewing angle, except for the occlusion of some parts. However, computation of this structure from certain viewing angles may be difficult. Decomposition of an object into parts is not unique. Some objects are reasonably described as having alternate structures in one or more views, but the number of such alternatives is small.

A generalized cone representation for primitives is described below; complete symbolic descriptions for objects using these primitives are discussed in Section 4.

2.1. Generalized cones

Generalized cones, introduced by Binford [5], are used as the main primitives. A generalized cone is defined by a space curve, called the *axis*, and planar *cross-sections* normal to this axis. The cross-sections may be of any shape and may change along the axis; the function describing the change is known as the *cross-section function*. The axis and the cross-sections must satisfy the following constraints:

- (i) The cross sections must be normal to the axis.
- (ii) The axis must pass through "corresponding" points (e.g. the centers of gravity) of the cross sections.

Note that these constraints do not necessarily determine a unique cone description for a given volume, nor do they specify criteria for segmentation of an object into sub-parts. Each segmented part is to be a simple and continuous generalized cone having a smooth axis and cross-section function.

Generalized cones as primitives are attractive for describing shapes where cross-sections change smoothly along an axis. This is often true of elongated shapes (but not restricted to them). Elongated shapes are commonly found in both man-made and natural objects. This representation has similarities with the Blum medial axis transform [6]; a detailed comparison and discussion of differences is found in [2].

Other primitives, such as spheres and planar surfaces, are allowed in the representation scheme. Holes are conveniently represented as negative volumes

described by the same primitives as for the solid parts. However, we have not implemented machine description techniques for primitives other than generalized cones or for description of holes.

3. Segmentation

The problem of separating different objects in scenes of polyhedral objects has been investigated extensively [8, 9, 14]. Only limited progress has been achieved in extending techniques successful in that domain to scenes of curved objects [7, 13]. Many of the difficulties are due to the use of a single, monocular view of the scene. Here, we have chosen to use direct range information about the visible surface of an object, obtained by a laser triangulation ranging scheme developed by Agin and Binford [1]. The ranging scheme is not described here; the reader may assume that the input to our system consists of three-dimensional positions of all points visible from a single camera view.

The range data allows immediate separation of non-touching objects. For



FIG. 1. A TV picture of a doll and a snake.

Artificial Intelligence 8 (1977), 77-98

example, Fig. 1 shows a TV picture of a doll and a snake shaped object; Fig. 2 shows a boundary derived from linking the discontinuities in the range data obtained for this scene. (Details of constructing boundaries from range data may be found in [11]. Note that in addition to the 2-d positions of the boundary points in the image coordinates, their three-dimensional positions are also known.) The two objects are separate, except that the lower left leg of the doll touches the snake. Also, this part of the leg is not directly seen as attached to the rest of the leg.

Given the three-dimensional positions of points on the surfaces visible from a single view, and the boundaries derived from such data, we aim to segment a single object into simpler parts. The representation described in Section 2 is not a transform representation and does not specify an algorithm for unique segmentation and description of an object. Instead, we segment an object into parts that can be described by "smooth" generalized cones, i.e. cones where axis directions and cross-section functions change continuously. Segmentations derived at this level are not expected to be perfect, in the sense in which humans would segment it. Context needs to be used to further segment a part or join segmented parts. Alternate descriptions result when multiple description hypotheses are reasonable; recognition programs choose the descriptions which match best with a stored model.

The body segmentation process consists of three main parts. First, parts of an object that can be described by local cones are determined by the use of a *projection*

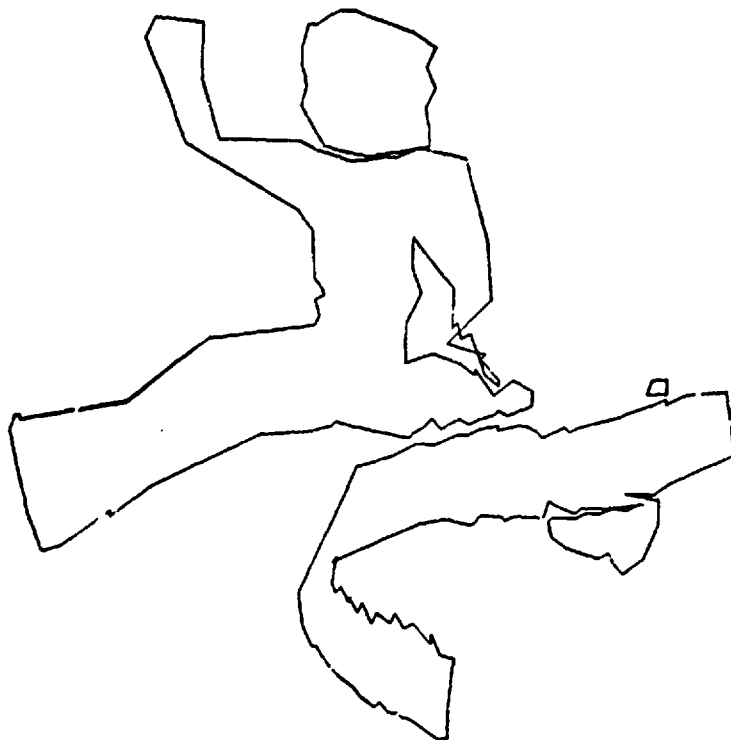


FIG. 2. Boundary derived from range data for Fig. 1.

technique (described below). A refined estimate for the axes of these local cones is then obtained and the cones extended to cover larger parts of the object, by allowing the axis directions to change smoothly. This often results in many parts of a body being described by more than one cone. A small number of the alternatives is chosen and symbolic descriptions generated for these choices.

In the previous section, specified constraints on generalized cones required the axis to pass through corresponding points of cross-sections, and centers of gravity were suggested as a choice of these points. However, as only part of the surface of an object is visible from any one view, these centers of gravity cannot be computed from available data. Instead, we require the axis to pass through the mid-point of the line joining the points of cross-sections that lie on the boundary of the visible surface of the cone. These mid-points provide a crude approximation to the centers of gravity, but are adequate for our descriptions, as the precise location of the axes is not very important.

3.1. Method of projections

Axis directions for cones describing parts of an object are not known a priori. We find these local cones by investigating cones with directions pointing in a

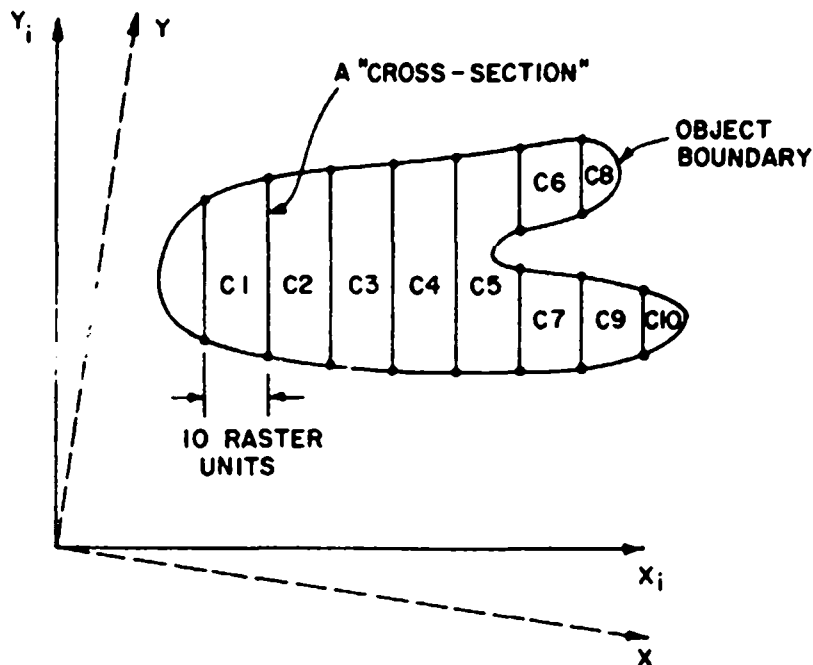


FIG. 3. Cross-sections for a selected projection direction.

number of equally spaced directions (typically 8). The following describes the steps of a projection technique for finding cones with axes pointing in a range of directions centered about a direction X_i .

1. Transform Co-ordinates: Transform the co-ordinates of the points on the boundaries of the object to a system with the new x -axis pointing along X_i .

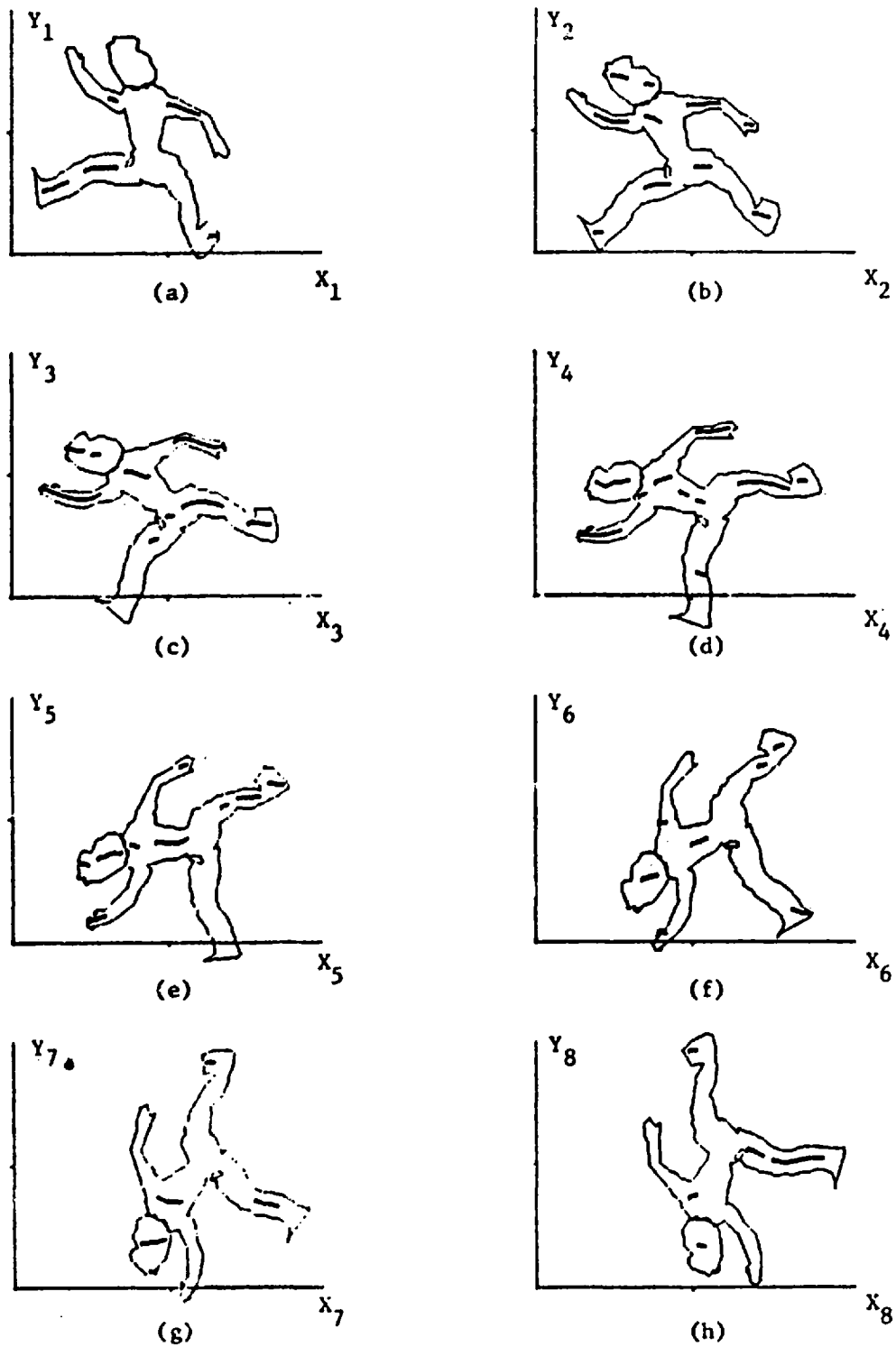


FIG. 4. Local cones generated by projection in eight directions.

2. Construct "cross-sections" normal to X_i at regularly spaced intervals, as shown in Fig. 3. Only the two points of the cross-section lying on the boundary are computed. We call the mid-point of the line joining these two points the mid-point of the cross-section.

3. (a) Find neighboring cross-sections. In Fig. 3, C_1 and C_2 are neighbors as are C_7 and C_9 , but not C_5 and C_7 nor C_6 and C_7 .

(b) If the angle between the line joining the mid-points of a pair of neighboring cross-sections and the new X -axis is less than the angle between two successive projection directions then these two cross-sections are taken to form a local cone (the axis is defined by the two mid-points). If either of them belongs to a previous cone, the other cross-section is included in the same cone. This step is repeated until all cross-sections have been examined.

The output of this procedure is a set of local cones, each defined by a list of points along the axis, a list of cross-sections and associated boundaries on the two sides of the cone. Figure 4 shows the axes of local cones generated by projection in 8 directions for a doll.

3.2. Axis refinement and cone extension

The axis of a cone computed by the above procedure is within a certain angle of the normal to the cross-sections (this angle is equal to the interval between two projection directions, typically 22.5°). A better estimate for the axis and the cross-sections is obtained by fitting a straight line to the current list of axis points and constructing new cross-sections normal to this straight line. The mid-points of the new cross-sections define a new axis. This process is repeated and usually converges in a very few steps. The divergence of this process merely indicates that this part of the object could not be satisfactorily described by a cone with axis pointing in the chosen direction.

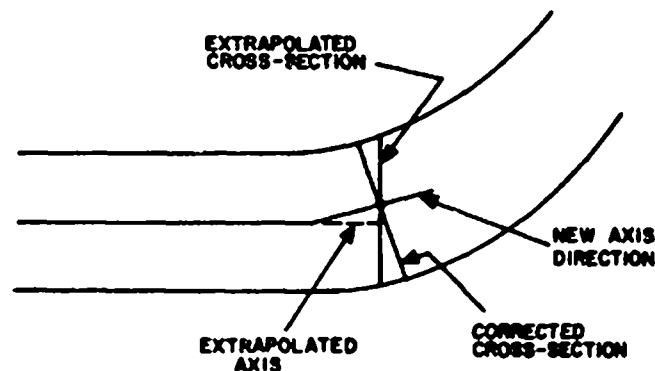


FIG. 5. Extension of a cone.

Each local cone is extended at both ends to cover larger parts of an object. Extension at one end proceeds by extrapolating the axis by a small distance and computing a new normal cross-section (see Fig. 5). If no new cross-section can be

generated, either because an end of the object has been reached or because no intersections can be found with the neighboring boundary of the cone, the extension terminates. The distance of the mid-point of the new cross-section from the extrapolated axis is computed. If this distance is larger than a threshold, a new axis direction for extrapolation is computed by including the new mid-point. Another cross-section is generated normal to the modified axis direction. It has not been necessary to iterate on this step.

Further tests are made on the acceptability of the new cross-section by checking the continuity of its size with the preceding few. A discontinuous jump terminates the extension. Each local cone generated by the projection method is extended in both directions. Figure 6 shows the cones resulting from the extension of local cones of Fig. 4.

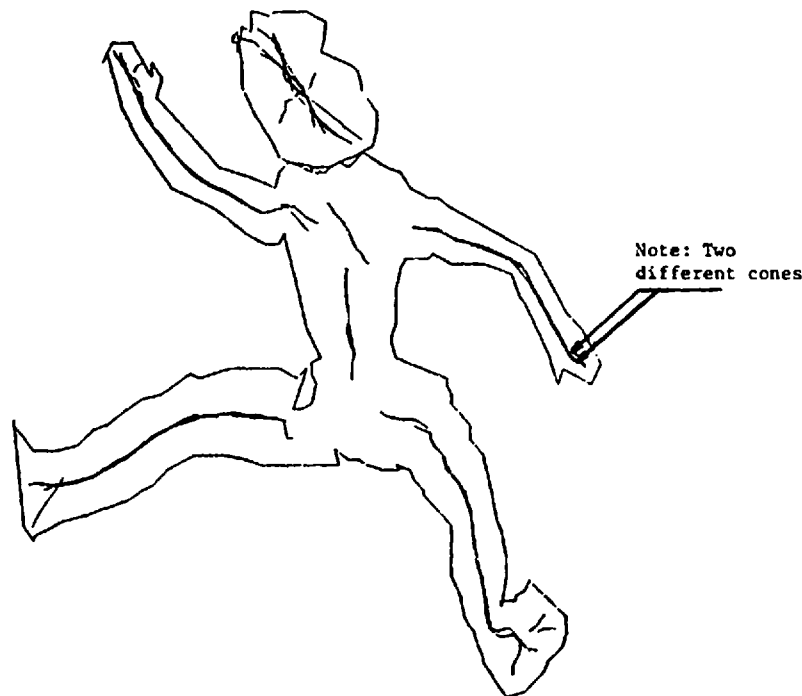


FIG. 6. Axes of extended cones for the doll of Fig. 4.

3.3. Choice of segmentation

The segmentation routines described above frequently generate several cones that describe the same part of an object (e.g., see the arms and the legs of the doll in Fig. 6). Each cone represents a possible segmented part. We aim to choose a small number of segmentations with the sub-parts in one segmentation being mutually compatible. The choice is based on preferring elongated and cylindrical parts.

The simplest form of overlap of two cone descriptions is when they describe substantially the same part of an object. This occurs when two separate local cones converge on similar extended cones. For example, see the two cones describing

the left arm of the doll in Fig. 6. In such cases, we simply choose the cone with the longer axis.

Other criteria for selection are explained by the example of a two-dimensional rectangle. Consider the various cones in Fig. 7, their axes being shown by dashed lines. The cones C_1 and C_2 , describing the corners are included in the cylinders C_3 and C_4 . The cylinders are the preferred choice. Among the two cylinders, C_3 is more elongated (has larger ratio of axis length to average cross-section width) and is chosen over C_4 .

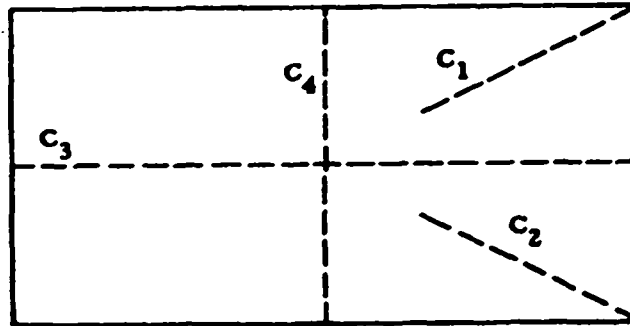


FIG. 7. Axes of different cones for a rectangle.

For some parts, two or more descriptions are equally suitable, e.g., the head of the doll is nearly spherical and many axis directions are equally good. In such

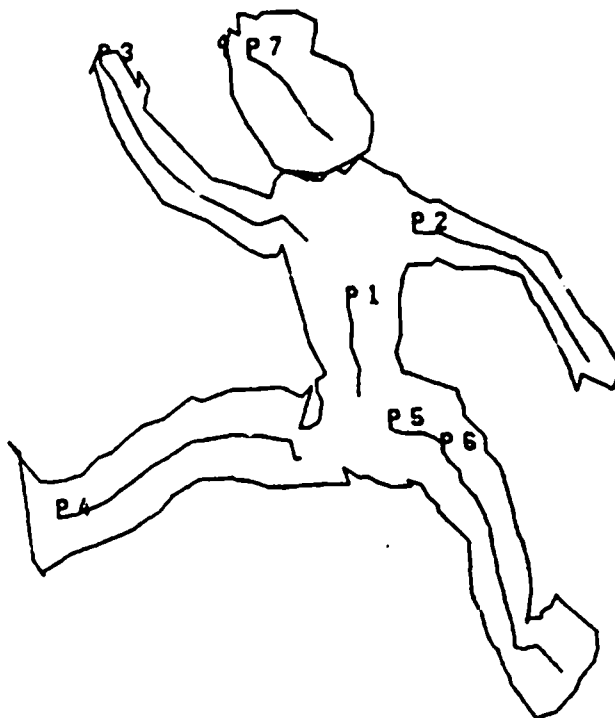


FIG. 8. Selected cones from Fig. 6 for a doll.

Artificial Intelligence 8 (1977), 77-98

cases, the various alternatives may be retained to yield multiple descriptions (our programs simply pick the cone with the longest axis).

Figure 8 results by applying these selection criteria to the cones in Fig. 6. At this stage, the cones P5 and P6 are determined to extend into each other continuously (the earlier cone extension processes failed to do so, because of a local discontinuity). An alternative description with these parts merged into one is also generated now.

Extension of some pieces is terminated by the interference of other pieces attached to this piece. For example, in Fig. 6, the piece describing the body does not extend into the shoulder area, the extension being inhibited by the presence of the arms. Descriptions of such pieces can be improved by redescribing them after removing the interfering pieces; we have not implemented such redescription techniques.

In previous work [1], Agin and Binford presented efforts at achieving similar segmentations. These methods were limited to generating descriptions of isolated parts, which were not related to each other to make a complete body. Major deficiencies prevented the use of their techniques on moderately complex scenes. They fit cylinders of circular cross-sections to the visible surface of the object. Such a method has no well-defined notion of the boundaries of a part and often merged two proximate but distinct parts, such as two fingers of a glove, by bridging boundaries. Such errors cannot be easily corrected at a higher level using context.

The description methods presented here are more structured because of the use of boundary. They exhibit substantially improved performance, are not limited to any particular cross-section shapes and are much faster as only the points on a boundary are considered.

4. Symbolic Descriptions

Symbolic descriptions for an object aim to capture the important shape properties and contain enough information for recognition of the object and for indexing into a memory of models for similar objects. The segmented parts (also called *pieces*) connect at joints. The symbolic descriptions consist of these connectivity relations, descriptions of the individual parts and joints, and global properties of the object.

Connectivity relations of an object may be viewed as a graph with joints as nodes and pieces as arcs between them or vice versa. Figure 9 shows the graph structure corresponding to the segmentations of the doll in Fig. 8 (this interpretation is for the alternative where P5 and P6 are merged). The "B" and "H" pieces of the graph are considered of key importance and are known as *distinguished pieces*, as explained later.

4.1. Piece descriptions

Summary descriptions for a piece describe the size and the gross shape of a piece, with the aim of facilitating quick, crude matching of two pieces. The size descriptors used are the length of the axis, the average cross-section width and the ratio of the

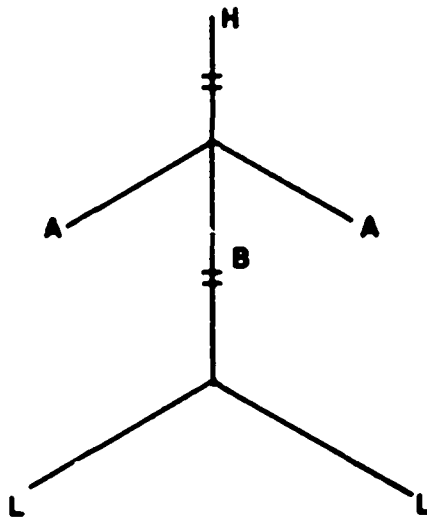


FIG. 9. Connection graph of a doll.

two. A linear approximation to the cross-section function determines the cone angle of a linear cone fitted to the piece. Pieces with cone angle exceeding a threshold are labelled *conical* (opposed to *cylindrical*). Elongated pieces (length to width ratio $>$ a threshold, say 3.0) are considered to be *well-defined* and of particular interest, as they are unlikely to appear spuriously in object descriptions.

The shape of the axis could be described in more detail by fitting a set of curves such as straight lines or splines to it, and could guide further segmentation of the piece. Cross-sections are a planar area and the generalized cone description

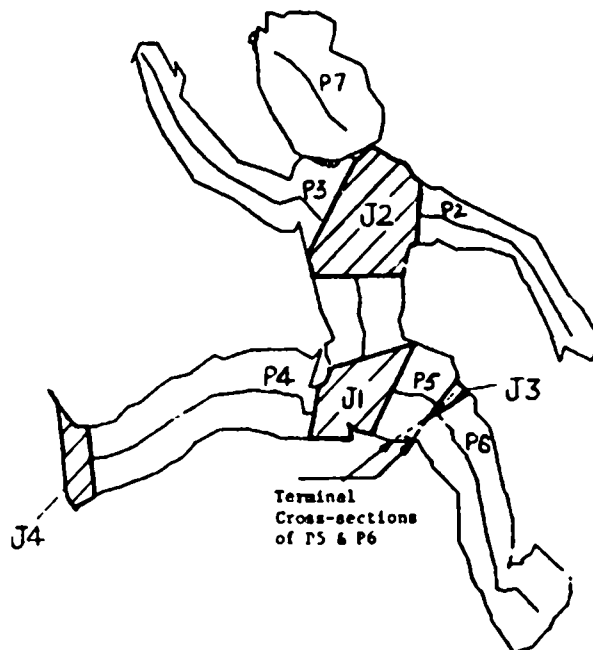


FIG. 10. Joints between pieces of the doll of Fig. 8.

Artificial Intelligence 8 (1977), 77-98

methods reduced to two dimensions could be used. Segmentation would permit description of complex cross-section shapes with discontinuities, as in a fluted cross-section for example. We have not used these detailed descriptions for the axes or the cross-sections (our description of cross-sections is limited to computation of their widths).

4.2. Joint descriptions

Two or more pieces of an object connect at a joint. The connectivity of parts is easily inferred from the boundary of the object (connection of isolated, shadowed parts is discussed separately). Areas of the body of an object, not covered by any piece are also considered part of the joints as shown in Fig. 10.

Symbolic descriptions of a joint consist of an ordered list of pieces connected to it and a dominant piece, which is defined to be the widest piece in the list. The order of pieces at a joint is not invariant with the viewing angle since the parts are connected to a two-dimensional surface, which does not have a useful, invariant ordering. However, for many cases, particularly where the parts occur along a plane curve, the order is preserved; our recognition programs assume such order preservation.

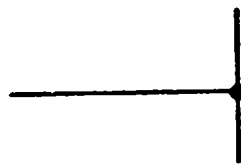
Joints may be described as being of certain types, based on the angular and size relationships of parts connected to it. Some joint types are shown in Fig. 11. However, these descriptions have not been useful for recognition because of the deficiencies of our description programs in determining the directions of the axes of the pieces near a joint, and the variability of joint types when the pieces are allowed to articulate.

4.3. Object description

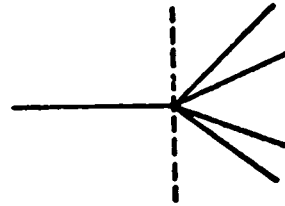
In addition to the connectivity relations of the pieces, object descriptions aim to capture important features of its structure. Simple descriptors are the number of pieces, number of elongated pieces and the number of joints. We define a set of distinguished pieces consisting of those pieces whose average widths are more than twice as large as that of the pieces not in this set. Another set of distinguished pieces is defined by pieces whose length to width ratio is at least twice as large as that of pieces not in this set. Usually, there are only a small number of such distinguished pieces, and they are used to speed up the process of matching for recognition.

With each distinguished piece is associated the number of pieces attached to it at either end and their sizes relative to its size. Also noted is whether the pieces at one end are clearly different (wider or longer) from the pieces at the other end. Unsymmetrical ends help in reducing the number of alternatives considered during recognition.

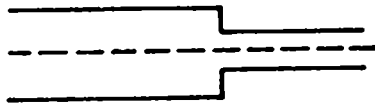
Bilateral symmetry of the object about different axes is examined. This evaluation requires comparison of similarity of pieces on either side of a symmetry axis.



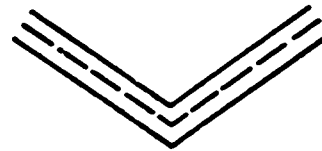
(a) T Joint



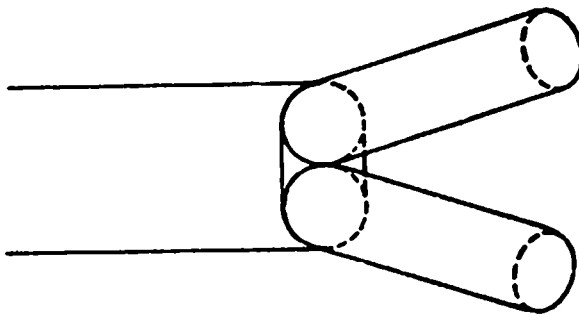
(b) Fork Joint



(c) Neck Joint



(d) Elbow Joint



(e) Cross-section Conserving Joint

FIG. 11. Different joint types.

As limbs are allowed to articulate, the angles between limbs are not used for symmetry evaluation. Because of the simple descriptors used for pieces, the symmetry calculation is correspondingly crude. However, we feel that with improved descriptions, symmetry can be very useful.

4.4. Linking of isolated parts

Some parts of an object are isolated from the rest, because of occlusion and shadows, e.g., the lower left leg of the doll in Fig. 2. (This problem is accentuated by our range measurement scheme, where the illuminating source is not collinear with the camera axis.) The connections of such parts are not known directly.

Among the clues that may be useful for hypothesizing their connections are the *Artificial Intelligence* 8 (1977), 77-98

proximity of pieces to others, continuity or collinearity of their axes, symmetry of the objects and evaluation of support and stability characteristics. However, the precise use and methods for their computations are unclear. We generate hypotheses simply by computing the joint closest to an isolated piece. These connectivity hypotheses are further examined during recognition. This approach is adequate only for scenes without heavy occlusion where enough connected structure is visible.

5. Recognition

An object is taken to be recognized when its observed description is determined to be a possible description of some previously seen or known object. Selecting a suitable sub-class of models to compare with the observed descriptions, the process of indexing, is an important and difficult part of the recognition process.

A store of model descriptions is required for recognition. We construct models by storing machine generated descriptions of objects from one view. Any gross errors, such as missing or extra parts, are corrected interactively. Better models could be generated by using descriptions obtained from several views. This is an interesting and non-trivial learning problem. Techniques similar to those presented by Winston in [15] are applicable, however, incomplete and imperfect descriptions must be reckoned with. Of course, models could be input by hand measurements, which is a tedious process. In our models, articulations of parts of an object are assumed to be completely unrestricted.

The procedures for matching must take into account the following. Arbitrary scale changes of objects and limb articulations are allowed. Machine descriptions are not invariant to viewing angles and articulations. Self-occlusion and occlusion by other objects result in some parts missing from descriptions, and so the ability to make partial matches is crucial. Occasionally, extraneous small pieces are generated during description. Non-circular cross-sections of pieces change with the viewing angles.

An object description may be viewed as a graph structure and the problem of matching against a model as the problem of matching two graphs. However, because of the expected variability of descriptions, the graph matching cannot be limited to discovering graph isomorphisms. Syntactic methods of graph matching are described in [3] and [4]. We present matching routines that take advantage of the semantics in descriptions.

Recognition consists of three major parts. First, important features of the description are used to index into memory models to retrieve a set of models similar to the object. The object description is then compared to each of these models and preferred matches chosen. Verification consists of checking whether the differences between a preferred model and the object description can be explained in a satisfactory way. We have not implemented verification methods.

The result of matching an object description with a model description is a

description of their differences. This form of matching finds similarities as well as differences and makes possible verification as well as learning of new models. A simple, weighted numerical evaluation of the differences does not allow deferral of decisions to a later stage, where more context is available.

Matching of two descriptions starts by matching similar, distinguished pieces only. The match is then grown to include other pieces, while maintaining consistent connectivity relations. The object description may have missing pieces but extra pieces result in an unacceptable match. The choice between models that match equally well with an object in the connectivity relations is based on the quality of individual piece matches. If one choice is not clearly better, then more than one recognition hypotheses are output.

Verification would aid in choosing between alternate matches. We have not implemented any verification methods; some possible techniques are described here. The metric differences between two matched pieces may be explained because of the different viewing angles. Model specific relationships between the visible pieces and the feasibility of some pieces being obscured could be checked. The support and stability relationships with the proposed piece assignments may be computed. New descriptions for some parts may use increased resolution, if permitted by hardware. As only specific differences need be resolved at this level, selective redescriptions of parts of an object are now possible.

Some details of matching are presented below, using an example. The indexing mechanism is discussed subsequently.

5.1. An example

As an example, consider the doll of Fig. 8, with its limbs articulated differently now, with piece segmentations as shown in Fig. 12 and connection graph shown in Fig. 13. Note that one arm and one leg are not connected to the rest of the object, but this arm is computed to be closest to the shoulder joint and the leg to the leg joint. The body and the head are labelled as distinguished pieces, as they are the two large pieces. A doll and a horse are suggested as the similar models (among the objects known to the program) by indexing.

Matching with a doll model is described in detail here. The doll model was generated by storing the description corresponding to that of Fig. 8. The head and the body are also the distinguished pieces of the model description.

The body as a distinguished piece is two-ended (connected at both ends) and the head one-ended (connected at one end only) in both the object and the model descriptions. The matching process starts by matching similar distinguished pieces. A two-ended object piece is not allowed to match a one-ended model piece. Thus the choices for the matching of the distinguished pieces are:

- (1) object body with model body,
- (2) object head with model head,
- (3) object head with model body.

Consider the first alternative. Here, two choices are possible for the matching of the joints:

(a) the object arm joint with the model arm joint; and the object leg joint with the model leg joint,

or

(b) the object arm joint with the model leg joint; and the object leg joint with the model arm joint.

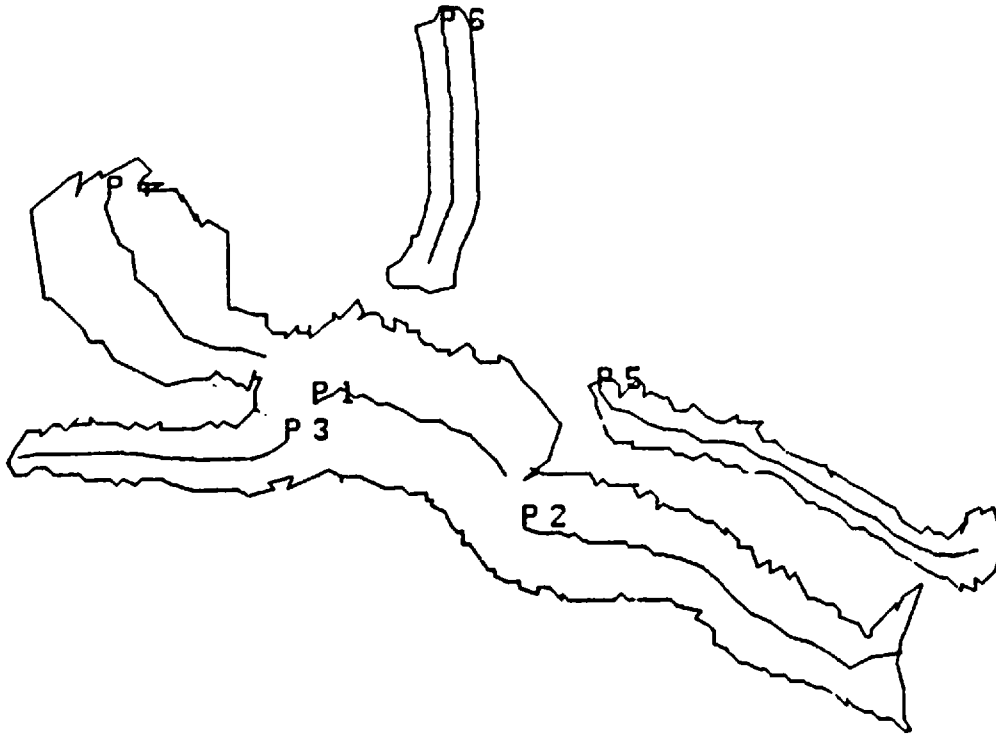


FIG. 12. Another view of a doll.

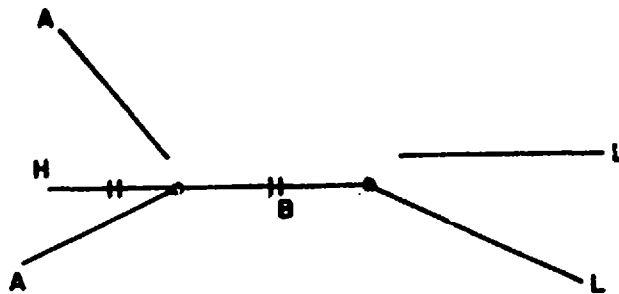


FIG. 13. Connection graph for the doll in Fig. 12.

The matching programs explore both alternatives. Consider option (a), and the details of matching of the two arm joints. The object doll arm joint has only two pieces besides the body attached to this joint (head and one arm), whereas the model description has three. These two lists are matched with each other, in

Artificial Intelligence 8 (1977), 77-98

direct and reversed order. For each order, the object head matched with the model head and the object arm with a model arm give the minimum total piece match error (piece match error is a weighted sum of the differences in the size descriptors, namely the lengths, the widths and the cone angles of the two pieces being matched). The left arm of the object matches marginally better with the left than with the right arm of the model. Note that the information about the angles between the limbs is not used, since the model does not contain articulation limits.

Having settled on the above matches, the programs note that the model has one extra arm and the object has an isolated arm, that could be connected to the joint being matched. This match is tried and found to be satisfactory and is retained.

Matching of the two leg joints proceeds similarly. In this case the isolated object leg is shadowed and its perceived width is smaller than the corresponding leg in the model description. However, it is known that the object leg is shadowed along its width and is allowed to match with the large model piece.

Now, examine the matching of the joints as in alternative (b) above. The preferred choices of matches are computed to be: the object leg with a model arm at one end; and the object arm with a model leg, and the object head with another model leg at the other end.

A choice is made between alternatives (a) and (b) now. The average piece error is clearly better for choice (a) (the ratio is $> 2:1$). The main discrimination was due to the poor mismatch of the head and the leg for the alternative (b).

Other possible distinguished piece matches are tried (alternatives 2 and 3 above). Matching the object head and the model head, ends up in a match that is identical to the above match. The other alternative, of matching the object head with the model body, is carried out but turns out to be clearly inferior as no possible matches can be found for the object leg if the connectivity relations are to be preserved. For the current example, the initial piece matches have used all the pieces and the joints of the two descriptions, and no extensions of the matches to the other joints are needed. (If the descriptions included the details of the hands at the end of the arms, we would extend the matches to the hands now.) Correct correspondence of the parts results from the matching of these two descriptions.

The object description is also matched against the model of a horse, and the following piece matches result as the best match between the two: the doll body with the horse body, the left doll leg with a rear horse leg, the right doll leg with the horse tail, the right doll arm with a front horse leg, and the doll head with the horse neck. No matches are found for the isolated doll arm, nor for one rear leg, one front leg and the head of the horse.

The output of the matching routines for this example is shown in Fig. 14. The models that are acceptable matches are shown in a preferred order. The assignments of pieces for match with each model are also shown, with object pieces referred to by their labels in Fig. 12, and model pieces by names assigned to them (at the time of storing models in the memory). The connectivity relations of the matches with a doll and a horse are identical, as they should be. More parts are *Artificial Intelligence* 8 (1977), 77-98

missing in the match with the horse, but we allow for the possibility of the parts to be hidden. Match with a doll is preferred based on the errors of piece matches. However, the differences are not large enough to make an unequivocal overall choice. If articulation limits of the limbs were known to the models, the angles between limbs would provide clear discrimination for this example.

```

matches in preferred order
DOLL
PRINTING PIECE CORRESPONDENCES
P3 ARM
P4 HEAD
P6 ARM
P1 BODY
P2 LEG

NO MATCH FOUND FOR THE FOLLOWING PCS OF THE OBJECT
none

NO MATCH FOUND FOR THE FOLLOWING PCS OF THE MODEL
LEG
HORSE
PRINTING PIECE CORRESPONDENCES
P1 BODY
P5 COMB_REAR_LEG
P2 TAIL
P3 FRONT_LEG
P4 NECK

NO MATCH FOUND FOR THE FOLLOWING PCS OF THE OBJECT
none

NO MATCH FOUND FOR THE FOLLOWING PCS OF THE MODEL
HEAD
REAR_LEG
FRONT_LEG

```

FIG. 14. Matching results for the doll of Fig. 12.

5.2. Indexing

If the number of stored models in the memory is large, it is impractical to perform recognition by comparing an observed description to each of these models. A number of important features are abstracted from a complete object description to form a description code and are used to index into memory for models with similar description codes.

Models with exactly the same description code can be retrieved efficiently by use of hash coding techniques. The problem of finding a closest match is more difficult, and is conjectured by Minsky and Papert [10], to require searching a

substantial part of the memory. However, use of semantics of the problem domain can reduce the required search.

Our approach is to generate one or more description codes for an object and retrieve models having the same description codes. Several more probes are made; based on the knowledge of variability of the descriptors in the description code, the code is modified and models with modified code are also retrieved. The object description is matched with each of the models so retrieved. Alternatively, each model could have been matched as it was retrieved and the process terminated as soon as an acceptable match was found.

The description code, in our implementation, is based only on the descriptions of the distinguished pieces of an object, each such piece generating one code. This permits indexing from partial views, if at least one distinguished piece is visible. The descriptors chosen to be included in the code are all binary valued. It is desirable that they be insensitive to variations caused by changing viewing angles, limb articulations and occlusion.

Only three descriptors have been used. They are the connectivity of the distinguished piece (connected at one end or both), its type (i.e. distinguished because it is long or because it is wide), and whether it is conical or not. Other descriptors that could have been added, if suitable description mechanisms were implemented, are the shape of the cross-sections (e.g., flat or curved, concave or convex), shape of the axis (straight or curved) and regularity of the piece (the cross-section function of the piece having a simple geometry).

Retrieval efficiency is increased by ordering the models with the same description code, by the minimum number of pieces attached at either end of the distinguished piece. Let N_1 and N_2 be the number of pieces attached at two ends and $N_2 < N_1$. Then the list is sorted by descending value of N_1 . Only those models having more attached pieces than the object are retrieved (assumes that an observed description, from any viewing angle, cannot have more pieces than the model). On the average, this reduces the number of models considered by a factor of two. Another factor of two can be obtained by further ordering the above list of models into sub-lists with the same value of N_1 , each sub-list ordered by value of N_2 . If the observed distinguished piece is connected on one side only, a modified description code also retrieves models with connections on both sides.

The efficiency of the indexing depends on the number of descriptors used. Assume n binary valued descriptors. Let m be the number of descriptors with value 1, for all models. The number of possible distinct codes then is the binomial coefficient $C_{n,m}$. Consider the case where the number of models is much larger than this and hence each code has a number of models associated with it. Modification of l descriptors for multiple retrievals from memory requires 2^l probes. The reduction in the number of models considered is then $C_{n,m}/2^l$, with a further factor of 4 obtained by ordering lists as described above. For example, if $n = 6$ and $m = 3$ then $c_{6,3} = 10$, and the best improvement factor is 40. For our implementation, $n = 3$ and $m = 1$ (or 2), and the expected improvement factor is 12. This is

based on the assumption, and belief, that the objects are evenly distributed over the chosen descriptors.

This indexing scheme has been a preliminary effort and suffers from the inadequacies of description mechanisms. Use of real valued descriptors, such as relative sizes of limbs will add to the effectiveness. Use of only a few models prevented further extensive experiments with the indexing.

6. Results and Conclusions

Five objects, a doll, a horse, a glove, a snake, and a ring were used in our experiments. Scenes of single as well as multiple objects in various orientations and with various articulations of limbs were processed. Pictorial results are not reproduced here (see [11]); the performance is summarized below.

Use of three-dimensional range data is very effective in separating occluded objects, except in some cases of objects which touch. The problem of separating such objects is similar to that of separation of objects in a monocular image and was ignored. Useful segmentations of an object into sub-parts are generated. Local discontinuities in boundary occasionally cause undesired splitting of parts. A major need is for redescription routines that utilize the existing segmentations.

Recognition programs discriminate well between objects of different structure, e.g., a hammer, a glove and a doll are not confused. In discriminating between objects with similar structure, such as a horse and a doll, the relative sizes of the parts are the discriminating factor. Use of models not containing information about complete cross-sections or articulation limits of sub-parts prevents reliable discrimination between similar objects in many cases. The widths and lengths of parts have large uncertainties which could be reduced by techniques of redescription which utilize the descriptions of neighboring parts. Further improvements are possible by verification. Verification would be aided by our use of comparison programs that produce a list of differences between the object and the model descriptions and assignment of labels to object parts.

Scenes with moderate amounts of occlusion are analyzed successfully. Objects are recognized from their partial views with the performance dependent on the amount of structure seen. Cases of heavy occlusion, where little or none of the structure is directly visible are not considered and will require additional techniques for hypothesizing structures from fragmented sub-parts.

The performance of the programs is intimately related to the representations chosen. The fact that objects with different structures are easily discriminated is because the representation brings out these differences. In other representations, e.g., coefficients of Fourier series expansion, the same differences may be extremely hard to detect. The chosen representation has proved useful for a variety of objects and can be strengthened by addition of more primitives.

The computer time required to process a typical scene is 5-10 minutes (on a PDP-10, KA-10 processor). Much of the time is spent in the lower levels of

program, such as the boundary linking and segmentation by projection. Many steps in this processing are independent of each other and amenable to simple parallel processing.

This work is aimed at solving the problems of scene analysis when the specific objects in the scene are not known a priori. In such tasks, it is necessary to generate descriptions from the scene before the local knowledge of models can be used. Ability to generate such descriptions, for a limited class of scenes, has been demonstrated. We believe that a general vision system will require ability to generate unguided descriptions of at least the complexity used here.

REFERENCES

1. Agin, G. A. and Binford, T. O. Computer description of curved objects, *Proc. Third Internl. Joint Conf. Artificial Intelligence*, Stanford, California (August 1973), pp. 629-635.
2. Agin, G. A., Representation and description of curved objects, Stanford Artificial Intelligence Laboratory Memo AIM-173, Ph.D. Thesis (1972).
3. Amber, A. P., Barrow, H. G., Brown, C. M., Burstall, R. M. and Popplestone, R. J. A versatile computer controlled assembly system, *Proc. Third Internl. Joint Conf. Artificial Intelligence*, Stanford, California (August 1973), pp. 298-307.
4. Barrow, H. G., Amber, A. P. and Burstall, R. M. Some techniques for recognizing structure in pictures, in: Watanabe, S. (Ed.), *Frontiers of Pattern Recognition*, Academic Press, New York, 1972, pp. 1-29.
5. Binford, T. O., Visual perception by a computer, *IEEE Conf. on Systems and Controls*, Miami (December 1971).
6. Blum, H. A transformation for extracting new descriptions of shape, *Proc. Symp. on Models for Perception of Speech and Visual Form*, Boston (November 1964), pp. 362-380.
7. Chien, R. T. and Chang, Y. H. Recognition of curved objects and object assemblies, *Proc. Second Internl. Joint Conf. Pattern Recognition*, Copenhagen (1974), pp. 496-510.
8. Falk, G. Interpretation of imperfect line data as three dimensional scenes, *Artificial Intelligence*, 3 (1972), 101-144.
9. Guzman, A. Decomposition of a visual scene into bodies, *Proc. AFIPS, Fall Joint Computer Conf.* 33 (1968), 291-304.
10. Minsky, M. and Papert, S., *Perceptrons, an Introduction to Computational Geometry*, MIT Press, Cambridge, Massachusetts, 1969.
11. Nevatia, R., *Computer Analysis of Scenes of 3-D Curved Objects*, Birkhauser-Verlag, Basel, Switzerland, 1976, to be published.
12. Roberts, L. G., Machine perception of three-dimensional solids, in: Tippett et al. (Eds.), *Optical and Electro-Optical Information Processing*, MIT Press, Cambridge, Massachusetts, 1965, pp. 159-197.
13. Turner, K. J., Computer perception of curved objects, *Proc. Artificial Intelligence and Simulation of Behavior Conf.*, London, 1974, pp. 238-255.
14. Waltz, D. A., Understanding line drawings of scenes with shadows, in: Winston, P. H. (Ed.), *The Psychology of Computer Vision*, McGraw-Hill, New York, 1975, Ch. 2.
15. Winston, P. H. Learning structural descriptions from examples, in: Winston, P. H. (Ed.), *The Psychology of Computer Vision*, McGraw-Hill, New York, 1975, Ch. 5.

Received March 1976