MCC Technical Report Number: ACA-254-87

Canonical Shape Description for 3-d Stick Bodies

MCC/ACA Non-Confidential

Adolfo Guzman

July 13, 1987

Stick bodies are those three-dimensional bodies characterized by a juxtaposition of more or less elongated limbs (generalized cylinders or cones) meeting together at more or less round junctions (corners). Each component limb can extend in one of three roughly orthogonal directions; the junctions essentially have no preferred axial direction; limbs can have shapes, dimensions and other attributes associated with them; junctions too, although their shape is considered "boxy." Stick bodies are classified into *limb strings*, if at a junction only two limbs can meet; *limb trees*, if more than two limbs can meet in a junction, but they do not form cyclic paths, or *general stick bodies* comprising cyclic paths.

For stick bodies without cyclic paths, a canonical (i.e., unique) shape descriptor is defined, and a way to construct it is proposed. Since the descriptor is unique, to determine if two stick bodies have the same shape, it is only necessary to see if their descriptors are equal. In this manner graph comparisons and tree searches are eliminated. The shape descriptor does not take size or orientation into account, but the paper shows how to do this, too, should it be deemed necessary.

Cover sheet

Canonical Shape Description for 3-d Stick Bodies

Adolfo Guzman

Microelectronics and Computer Technology Corporation
3500 West Balcones Center Drive
Austin, TX. 78759 Tel. (512) 338 3753

Index terms: shape, 3-d description

Regular paper for IEEE Comp. Soc. Workshop on Computer Vision, Miami, Nov. 87

July 13, 1987

Canonical Shape Description for 3-d Stick Bodies

ABSTRACT

Stick bodies are those three-dimensional bodies characterized by a juxtaposition of more or less elongated limbs (generalized cylinders or cones) meeting together at more or less round junctions (corners). Each component limb can extend in one of three roughly orthogonal directions; the junctions essentially have no preferred axial direction; limbs can have shapes, dimensions and other attributes associated with them; junctions too, although their shape is considered "boxy." Stick bodies are classified into *limb strings,* if at a junction only two limbs can meet; *limb trees,* if more than two limbs can meet in a junction, but they do not form cyclic paths, or *general stick bodies* comprising cyclic paths.

For stick bodies without cyclic paths, a canonical (i.e., unique) shape descriptor is defined, and a way to construct it is proposed. Since the descriptor is unique, to determine if two stick bodies have the same shape, it is only necessary to see if their descriptors are equal. In this manner graph comparisons and tree searches are eliminated. The shape descriptor does not take size or orientation into account, but the paper shows how to do this, too, should it be deemed necessary.

Shape; 3-d description.

August 14, 1987

1. Introduction and Definitions

The *structural* approach to Pattern Recognition [Narasimhan, Shaw] and Scene Understanding relies on the decomposition of a scene into identifiable parts and its subsequent syntactical analysis. In this work, it is assumed that complex three-dimensional objects can be decomposed in two kinds of parts: "limbs" and "junctions."

"Limbs" are elongated parts (generalized cylinders [Binford, Ponce *et al*]); "junctions" are places (which could also possess shape and size) where limbs meet. The limbs are juxtaposed so as to form either three-dimensional "strings" or wires, "trees" or "graphs." If each limb is allowed to extend in one of three directions, the 3-d objects can be termed "stick bodies." In order to describe these objects, it is necessary to describe the shape of each limb and each junction, as well as the manner they interconnect: their incidence matrix. This paper will emphasize in the description of the juxtaposition or structure of the body, leaving to others the problem of describing limbs [Binford, Ponce *et al*] or junctions [Sagols]. The descriptor sought will be ***unique***, in that two 3-d stick bodies will yield the same shape descriptor, irrespective of size, orientation, etc. Should these parameters be needed, they can easily be introduced, too, in the descriptor.

The descriptor is unable to handle (describe) the following kinds of bodies:

- Stick bodies comprising cycles;
- Stick bodies where their limbs span in more than three roughly orthogonal directions;
- More general three-dimensional complex bodies that can not be characterized as a juxtaposition or "3-d graph" of limbs and junctions.

The descriptor resembles *shape numbers* [Bribiesca & Guzman] because absolute direction (as in Freeman chains) is disregarded; only relative directions ("turn left," "go straight," "turn right") are kept, but this time in three dimensions. The descriptor has some resemblance to that used by chemists for unique characterization of molecules.

1.1. Definitions

For convenience, a group of terms with their meaning is now given. The body of the paper will clarify them further.

Limb. A roughly cylindrical or conical elongated object or part of a body. It has a definite long axis, which has a direction in 3-d space. It has a roughly "circular" cross-section, and properties similar to generalized cylinders or cones [Ponce *et al*].

Junction. The place where two or more limbs meet. It has a "boxy" shape or no significant shape, but it definitely does not have a limb shape: it lacks a long axis.

Axes of expansion. Three roughly orthogonal axes along one of which the axis of each limb is allowed to extend. Limbs are ***required*** to extend along these axes; bodies which violate this property can not be described by the methods of this paper.

Stick bodies. Those 3-d bodies formed by limbs meeting at junctions.

Limb string or 3-d wire. A body formed by limbs and junctions in such a way that at each junction at most two limbs meet. It can be described by a succession of limbs and junctions, plus changes of orientation of the limbs' axes.

Stick tree. A body formed by limbs and junctions where more than two limbs can meet in a junction, but such that they do not form a closed loop (graph window). This paper shows how to derive the shape descriptor of stick strings and stick trees.

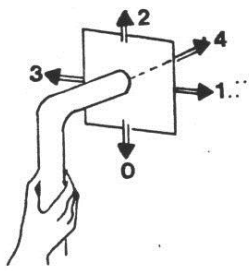

Figure 1.1. 1. Direction of Turns

Holding upwards the limb before the last turn (junction) just traveled, and placing the limb after such turn facing forward, the notation 0 through 4 defines the relative direction of the next turn.

Although this is apparently an undue restriction, notice that in practice rather complex real-life mechanical components and man-made artifacts can be described, since each limb and junction can have its own shape (under its own restrictions).

Base-five digit. One of the digits 0, 1, 2, 3, 4, used to describe a 3-d turn as shown in Fig. 1.1.1.

Chain. A sequence of base-five digits describing the sequence of 3-d turns (bending pattern) composing a limb string. It, together with shape information about each limb and each junction, forms the shape descriptor of a limb string.

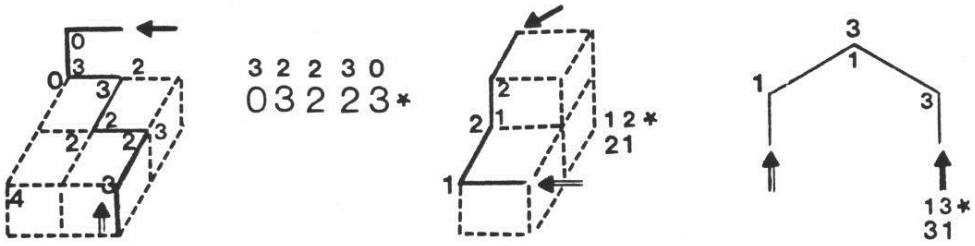

Figure 1.1. 2. Examples of Limb Strings

The canonical chains describing the shapes are marked with (*): they are the smaller (in the numerical sense) of the two possible chains for each 3-d shape.

Square list. A list starting with [, ending with], and having as elements either base-five digits or round lists. From an element of an square list to the next, one travels "along" a limb.

Round list. A list starting with (, ending with), and having as elements either base-five digits or square lists. From an element of a round list to the next, one travels "around" a junction; for instance, the list (0 1 4) represents a junction with limbs in the directions 0, 1 and 4 —but not in directions 2 or 3 —.

Wiggly list. A list starting with {, ending with }, and having as elements either base-five digits or square lists. An optional wiggly list may be present at the start of a shape descriptor, signifying the initial limbs sprouting from the "nameless junction."

Nameless junction. The first junction in a stick body; defined in Chapter 2.

Shape descriptor. For a limb string, a square list (consisting only of base-five digits; i. e., lacking round lists) enriched with shape information about each limb and junction. For a limb tree, a square list enriched as above, possibly having round lists as elements too, and with an optional wiggly list at its beginning.

Skeleton. The skeleton of a stick tree is a similar tree where all the limbs are represented by straight lines of the same length, and all the junctions are represented by points. Thus, the skeleton captures only the structure of the bends and turns of the tree.

Skeletal string. The skeletal string of a stick tree is a unique path (a limb string) derived in Chapter 4. It is central in finding a unique descriptor for the stick tree.

1. Notation for turns

Turns occur only at junctions. Instead of having a notation that gives an "absolute" direction, such as Freeman chains, this paper uses "relative" direction, such as shape numbers [Bribiesca & Guzman], but specified in three dimensions.

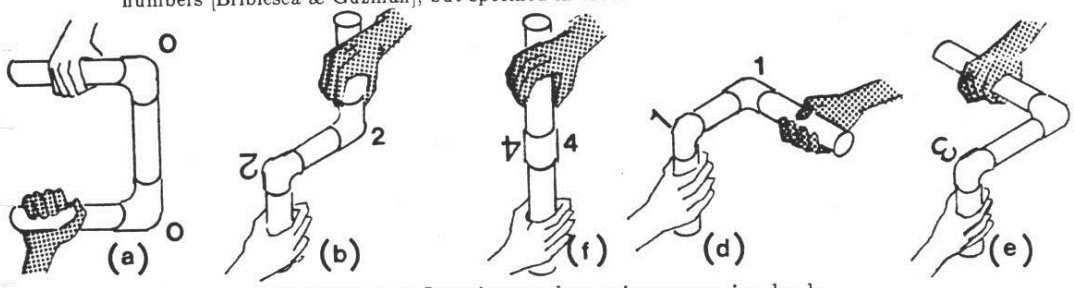

Figure 1.1. 3. Invariance when going or coming back

The first turn (junction) receives no number. For subsequent turns, Fig. 1.1.1 "Direction of Turns" specifies the five possible directions that a turn can have, *relative to the previous turn.*† Holding up the limb traveled just before the last turn, in such a way that the last limb is pointing forward, "0" means that the turn is *down* (going back or making a U turn with respect to the held limb) "1" means that the new turn is to *the right,* "2" is *upward* (stair-case fashion), "3" turns to *the left* and "4" *goes straight* through the junction following the direction of the last limb.

Starting at the white arrow, the white chains of digits are obtained for the "limb strings" of Fig. 1.1.2 "Examples of Limb Strings." Starting at the black arrow, the black chains are obtained for those same 3-d shapes.

2. The name of a turn does not change if traveled backwards

It is no coincidence that white chains are the inverse of black. For a limb string, the chain obtained by traveling the body in one direction is just the reverse of the chain obtained traveling the same body in the opposite direction. To prove this *theorem* it is

† If *this turn* is a 4, it is so labeled. If not a 4 but the previous turn was a 4, this turn receives its base-five digit with reference to the closest (previous) turn which is not a 4. In this manner orientation is not lost. If there is no such closest previous turn $\neq 4$, then *this turn* is arbitrarily assigned a 1, as if it were turning to its right. This introduces the slight anomaly of Fig. 1.1.4.

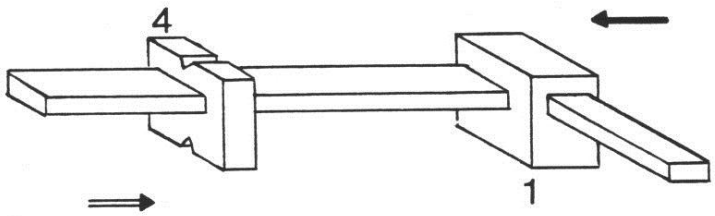

Figure 1.1. 4. Slight anomaly
The chain is 1 in one direction or 4 in the other, due to the arbitrariness of the footnote of the preceding page. The chain for this figure is min (1, 4), or 1, traveled →. No problem arises.

sufficient to examine the five names (digits) assigned to turns (Fig. 1.1.1). The white hand (Fig. 1.1.3 "Invariance when going or coming back") of (a) will assign a white 0 to the turn; the black hand will assign a black **0** to the turn, too, if both obey the rules of Fig. 1.1.1. The 0's will be assigned to "different" junctions, but what it is really happens is that it is the *spatial configuration* (the U-shaped turn) the one named 0. In (b), both hands sense a "staircase" and assign a 2. Similar symmetries hold for (c)-(e).

2. The Shape Descriptor for Limb Strings

The reversibility of the chain provided by the previous theorem comes handy for finding a unique descriptor of a body which is a limb string. The procedure is:

(0) Start from any end point of the body. The first turn (junction) receives no label; it is the "nameless junction."

(1) For subsequent bends, find the chain describing them, as Fig. 1.1.1 dictates. Let x be the resulting chain.

(2) Find x' = (reverse x). This is tantamount to traveling the limb string backwards.

(3) Choose as unique representation for the limb string the minimum among x and x', when seen as numbers formed by base-five digits. The construction just used proves that the chain chosen in (3) uniquely describes the 3-d bends. If x = x', that means that the 3-d body is symmetric with respect to the sequence of turns: they are the same no matter in which of the two possible manners the body is traveled. The chain by itself is unable to select a preferred direction of travel. Another rule, which *has* to take into account the shapes of the limbs, is needed:

(4) If x = x', prefer to travel (i. e., to describe) the body starting from the limb with the simpler (or smaller, in a limb-shape sense) shape descriptor. If the end-limbs have the same shape descriptor, use for comparison the shapes of the junctions, preferring to start with the smaller. How to compare two limb or junction shapes is outside this paper's scope.

For the bodies of Fig. 1.1.2, their shape chains are signaled by an *. Rule (4) was not needed to uniquely form their chains. See Fig. 2.1.2 for examples using rule (4).

Since the bodies have more than bending patterns (described by chains), additional shapes have to be inserted in the descriptor.

2.1. Additional properties of limbs and junctions

In addition to the chain describing the bending pattern of the limb string, it is necessary to describe the shape of each stick and of each junction, in order to accurately form a shape descriptor for a limb string. This is easily done inserting before each base-five digit first the shape of the corresponding limb and then the shape of the corresponding junction. In addition, since each limb string begins with an "nameless junction," it is necessary to introduce two more shape descriptions: those of the first limb and the unnamed junction. Also, after the last base-five digit, the shape of the last limb has to be inserted. Finally, at the beginning and end of the descriptor the descriptions of the beginning and ending junctions have to be inserted, if they exist.

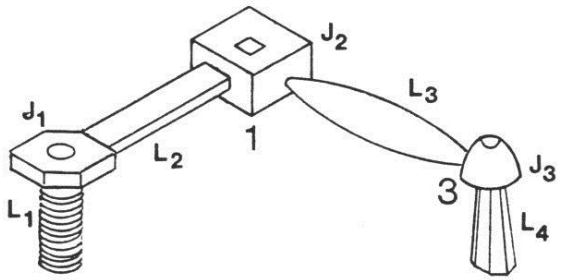

Figure 2.1. 1. Shape Descriptor for a Limb String
J0 is the beginning junction; J4 the ending junction, both considered non-existing in this case. J1 is the nameless junction. The chain for this limb string is 1 3. The nameless junction orders the insertion of L1 J1 at the beginning of the descriptor. Then L2 J2 is introduced before 1, and L3 J3 before 3. Then the final limb, L4, is introduced.

Figure 2.1.1 "Shape Descriptor for a Limb String" will be used as example. The chain representing its bending pattern is 1 3 and therefore its shape descriptor is

[L1 J1 L2 J2 **1** L3 J3 **3** L4]

where the chain **1 3** has been rendered black for easy visualization, and Li and Ji represent the respective shapes of limbs and junctions. In practice, they could be lists containing their parameters in some agreed-upon order. If these parameters are not direction-independent (for instance, a cone gets "smaller" going one way along the main axis, but gets "larger" traveling the opposite way), then their description must be referred to the direction of travel specified by the chain (1 3 in our example). Also, in order to describe major and minor axes of an ellipse which generates a "generalized cone," the directions of Fig. 1.1.1 come to play, and the ellipse axes must refer to them. Instead of going into details, the general principle to follow is that whatever shape descriptions are needed for limbs and junctions, they must be made *relative* to the direction of travel dictated by the chain, and by the sense of "up," "down," "to the left," etc., given by Fig. 1.1.1.

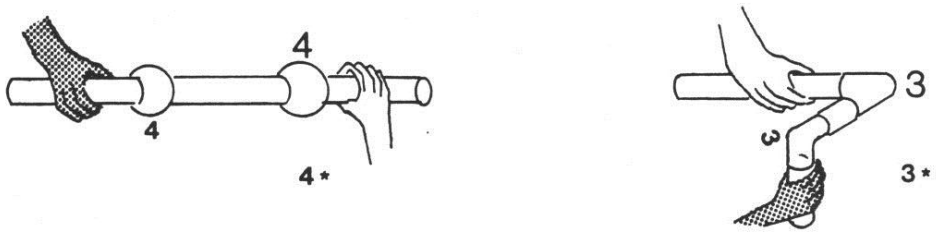

Figure 2.1. 2. Some Bodies with Three Limbs

1. Shapes for end junctions

It is possible for some bodies to have end junctions at one or both ends. Thus, the above shape descriptor has to be surrounded by the shapes of the junctions at their ends, which are J0 and J4, rendering

[J0 L1 J1 L2 J2 **1** L3 J3 **3** L4 J4].

If the beginning or ending junction is missing, () is used instead of J0 or J4.

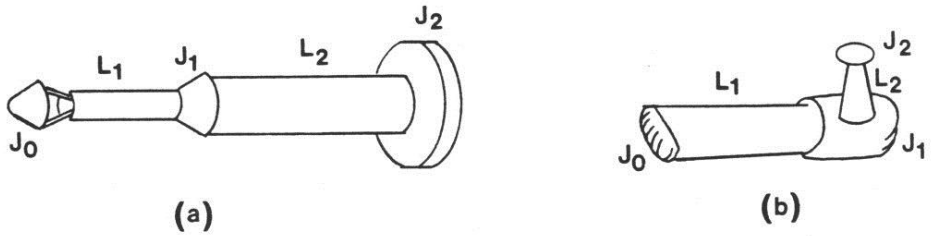

Figure 2.1. 3. The two Possible Bendings for Bodies with Two Limbs
Concentrating only in the bending pattern, two limbs can either be along the same axis or along two different axes. A preferred direction of travel has to be chosen, too; the text explains how.

2. The Descriptor for Bodies with Three Limbs

Bodies with three limbs (Fig. 1.1.3; Fig. 2.1.2 "Some Bodies with Three Limbs") have a unique chain: a single digit. To disambiguate among the two possible directions of travel, rule (4) at the beginning of Chapter 2 has to come to play. See also Fig. 1.1.4.

3. The Descriptor for Bodies with Two Limbs

The rules at the beginning of Chapter 2 assign *no* chain for bodies with two limbs. Nevertheless, as far as the shape of bends is concerned, two limbs can only be straight or be bent, as Fig. 2.1.3 "The Two Possible Bendings for Bodies with Two Limbs" shows. Clearly, rule (4) has to select the direction of travel. Suppose that for body (a) of Fig. 2.1.3, this direction is from junction J0 to junction J2. A possible descriptor may be then

[J0 L1 J1 mm L2 J2]

where mm is the "missing" information (one bit: it either goes straight through or it turns) about the knee: mm is for instance 4 for "going straight through" or 1 for *bending* (it does not matter in which direction, since the descriptor is about the shape). These numbers 4 and 1 have a different meaning than the directions chosen in Figure 1.1.1. This meaning is never confused, because bodies with two limbs have shorter descriptors than those with three, which have a descriptor of the form

[J0 L1 J1 L2 J2 **n** L3 J3]

where **n** is one of the digits of Fig. 1.1.1.

4. The Descriptor for Bodies with One Limb

One-limb bodies have, of course, no bends. Rule (4) has to apply, first to compare whether the limb going from one of its end to the other has a simpler (smaller) description than the same limb traveled backwards. In the unlucky case that the limb is symmetrical (shape → is same as shape ←), compare shapes of junctions as rule (4) indicates. The descriptor of a body with one limb is [J0 L1 J1] and thus it is easy to recognize by its length.

5. Taking Size into account

Both relative size and absolute size can be taken into account if, when specifying the lengths of the bodies, linear dimensions of cross-sections, etc., either relative or absolute numbers are used. Relative numbers may refer to the longest limb, for instance. Therefore, the descriptor proposed does not vary, although the meaning of the descriptions for Ji and Li may change to reflect the importance of size for a particular application.

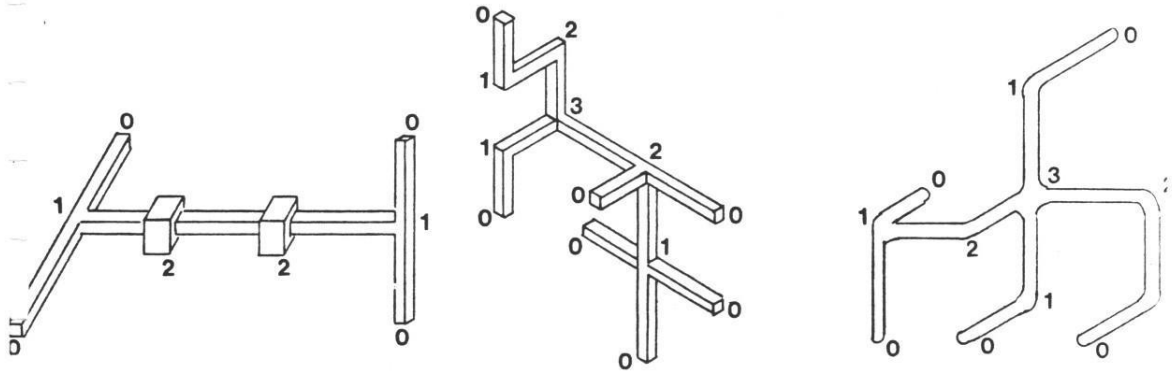

Figure 3.1. 1. Limb Trees

One of the challenges is finding a unique traversal from one (of several) ends to another end. The numbers in each junction are *privacy numbers* which indicate how "inside the tree" each junction is.

3. The shape descriptor for Limb Trees

When junctions are allowed to receive more than two limbs, limb strings become limb trees, which are the subject of this Chapter.

Just as in the limb string case, a significant challenge is to find, regardless of limb or junction shape, a unique traversal of the tree. The challenge is compounded precisely because the "skeleton" (the figure considered only from its 3-d bending pattern) is a tree, and no longer a string.

After a unique traversal, another challenge is to represent each junction, which is no longer only a shape plus a base-five digit telling how to turn; now, a junction is in general a sub-tree, a limb sub-tree.

3.1. Unique Traversal of the Tree

The *skeletal string* of a stick tree is a unique path (a unique limb string) that will serve subsequently to generate the descriptor for such a tree. This section prescribes how to find such a string among all possible leaf-to-leaf paths in a tree. *Privacy numbers* are introduced, which indicate how internal or buried into the tree each junction is. With these numbers' help, the search for a unique path is easier.

1. Privacy numbers

The following algorithm finds privacy numbers for a tree:

(1) Assign to every node a privacy number of 0.

(2) Leaf nodes retain the number thus assigned and are simultaneously (in parallel) removed from the tree (for purposes of this algorithm); all other nodes —those not removed— get their value incremented in 1.

(3) Go to rule (2), taking into account that the nodes just removed have caused *some other* nodes to become leaf nodes. When all the nodes have disappeared, the algorithm stops; the privacy number of each node is that number gotten by it before its dismission.

Examples are given in Fig. 3.1.1 "Limb Trees." Privacy numbers thus defined do not measure the distance from a particular node to the closest leaf.

Starting from the highest privacy number, go down to private numbers equal or immediately lower, until a 0 at each end is reached. If this path is unique, then this path can be considered as a *limb string*, and the methods of the preceding chapter can be applied to determine which of its two travel directions should be chosen. Once this is done, the *skeletal string* has been found.

In general, chains formed by privacy numbers are of the form

0 1 2 3 4 3 2 1 0 or

0 1 2 3 4 4 3 2 1 0

but it will be impossible to have a path such as ... 3 2 3 ... where a lower privacy number is in the middle of two higher ones. These paths are the longest possible paths in the tree. It is also guaranteed that next to a non-zero number either an equal or the next lower number is found. Also, although two consecutive equal privacy numbers can be along a path, three can not be.

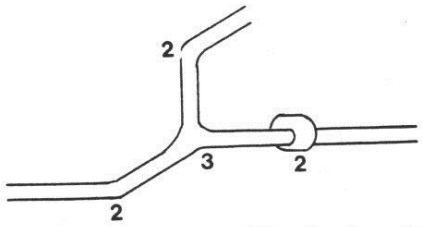

Figure 3.1. 2. In What Direction to go?

Possibly the chain is not unique because (See Fig. 3.1.2 "In What Direction to go?") there are more than two immediately lower privacy numbers next to a higher one. In this case, *each path* is considered, in turn, as a *limb string;* its chain number is determined by rules (0)-(4) of Chapter 2 *moving away from the higher to the lower privacy numbers*; the path giving the smallest chain is the selected one. This procedure could entitle in theory lots of search, but only a little occurs practically. Figure 3.1.3 "A Limb Tree that needs Search" gives an example.

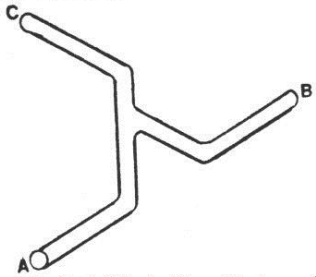

Figure 3.1. 3. A Limb Tree that needs Search
Path AB has chain **3 3**; A C has **4 1**; BC has **1 2**. Thus, BC (with its chain) is chosen to represent the "skeleton" of the stick tree.

The outlined procedure obtains a unique path for the limb tree. Also, a preferred direction for travel along this path is chosen by the rules (0)-(4) of Chapter 2. Thus, the *skeletal string* of the limb tree has been found. But it is not possible to travel along this skeletal string and use the descriptor of Chapter 2: one that takes into account multiple limbs at junctions is needed.

3.2. Constructing the Descriptor for the Limb Tree

The basic idea is to proceed along the skeletal string and write down the shapes of knees, limbs, and base-five digits, just as it was done with limb strings. Nevertheless, bundles of limbs coming at one junction prevent such a simple approach.

To simplify matters, the description that follows explicitly ignores the insertion of shapes for limbs and junctions, concentrating solely in the description of the *skeleton* (Definition in Chapter 1) of the limb tree. For this, square and round lists (Cf. Definitions in Chapter 1) are used. The procedure will be exemplified with the body of Fig. 3.2.1 "Limb Tree and its Descriptor." Since the skeletal string for this body is D B,

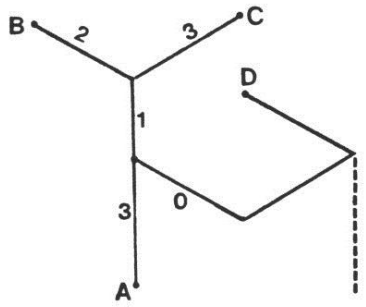

Figure 3.2. 1. Limb Tree and its Descriptor
The skeletal string is DB; its chain is **0 1 2**. The dotted limb, if present, gives rise to a wiggly list.

we will be tempted to write [0 1 2]. Unfortunately, "1" needs to be replaced by a composite junction (since "1" just refers to the right turn that goes to B, but what about the left turn that conduces to A?). Thus, "1" gets replaced by a round list, representing the junction where three limbs meet. The limbs sprouting from that junction (as seen by a traveler arriving from D) are 3 and 1; 1 goes before 3. Consequently, we write [0 (1 3) 2]. Then, one notices that the limb along 1 continues both to B and to C. Instead of 1, something like [1 ...] is needed. The branch that goes to C has (along the path 0 1) label 3. The other path has label 2. 2 goes before 3. Thus, [1 (2 3)] is written, and the complete path is [0 ([1 (2 3)] 3) 2].

The last 2 before] is just the 2 in (2 3); when deleted, the final descriptor for Fig. 3.2.1 is obtained:

[0 ([1 (2 3)] 3)].

A quick check: the number of digits is two less than the number of limbs. The number of round lists is equal to the number of joins having more than two limbs.

Conversely, given the descriptor [0 ([1 (2 3)] 3)], a way to generate the body is: first generate a bend 0, much like a U turn that makes a drawing from D to the central junction. Then, a junction with two branches follows: ([...] 3). Using the 3 first, the limb to A is drawn. Also, the sequence [1 (2 3)] needs to be drawn. The 1 goes to the right, where it meets the multiple junction (2 3), which leads to B and C. Inside square lists, moving from one member to another means "walking along the limb" (but staying in the same path or subtree) while in round lists, moving from one member to another means changing from one subtree to another.

1. Use of wiggly lists

Wiggly lists are used to describe sub-trees starting at the *nameless junction*. In figure 3.2.1, if the dotted limb were real, it would be described as {1}, traveling *backwards* on the skeletal string. Thus, the descriptor in this case would be

[{1} 0 ([1 (2 3)] 3)].

Additional examples of shape descriptors are given in Fig. 3.2.2.

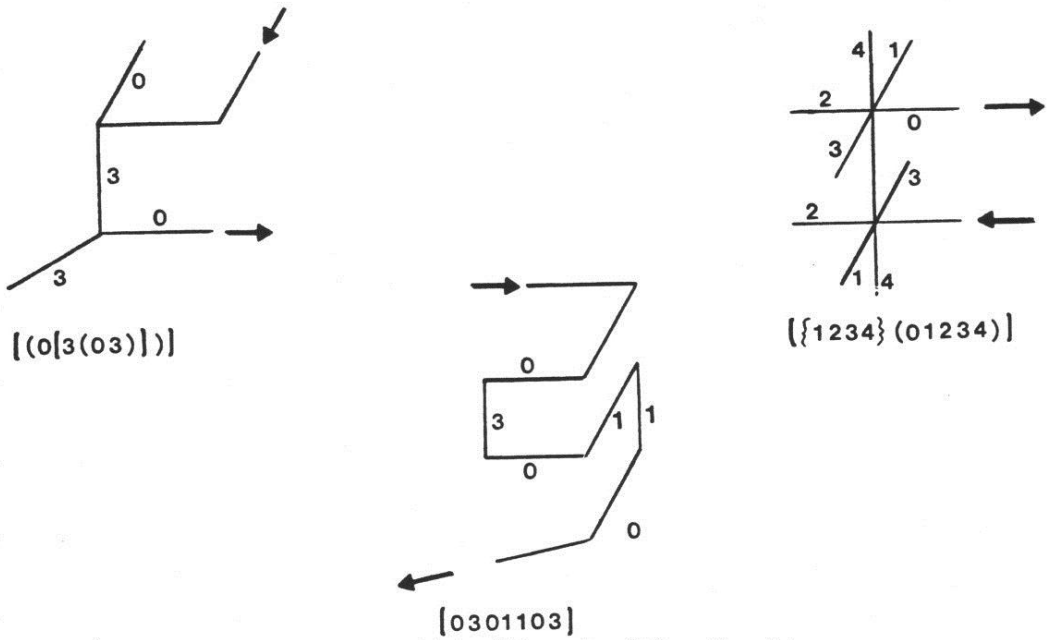

Figure 3.2. 2. Additional Examples of Shape Descriptors

4. Conclusions and Suggested Further Work

1. Conclusions

The paper gives a method for describing the shape of 3-d bodies, using a canonical shape descriptor. A unique descriptor takes more trouble to create than a non-unique one, but it is much easier to compare if two bodies have the same shape: they just have to have the same descriptor. Thus, tree or graph matching [Cooper & Holbach] is avoided.

The bodies are formed by elongated parts protruding in three roughly orthogonal directions; these parts have shapes describing them. These parts (called limbs) meet at *junctions*, which may or may not have shape.

The descriptor permits easy introduction of the shapes of limbs and junctions, if necessary. Also, sizes can be introduced, although this moves the work away from "shape description."

2. Suggestions for Further Work

The work on 3-d shape description, and in particular, its use in shape comparison, is relatively new; many unexplored paths are waiting. Some of them are:

- Extend the comparison to allow inexact matches among bodies.
- Extend and complete the work to stick bodies having closed graphs or loops.
- Allow descriptions much in the spirit of ***regular expressions*** of automata theory, in order to be able to express infinite or repetitive bodies.
- Relax the requirement that the axes be roughly orthogonal.
- Allow the existence of big or bulky junctions having "middle attachment points."
- From a junction, allow more than three directions along which limbs can expand [Cooper & Holbach], but preserving the uniqueness of the representation.

References

Binford, T. O. Visual Perception by computer. *Proc. IEEE Conf. on Systems and Control,* Miami, December 1971.

Bribiesca, E., and Guzman, A. How to Describe Pure Form and how to Measure Differences in Shapes using Shape Numbers. *Pattern Recognition* 12, 2, 101-112. 1980.

Cooper, P. R., and Hollbach, S. C. Parallel Recognition of Objects Comprised of Pure Structure. *Proc. Image Understanding Workshop*, 1987. Morgan Kauffmann Publishers. 381-391.

Narasimhan, R. Syntax-directed Interpretation of Classes of Pictures. *Comm. ACM* 9, 3, 166-173. March 1966.

Ponce, J., CHelberg, D., and Mann, W. Analytical Properties of Generalized Cylinders and their Projections. *Proc. Image Understanding Workshop*, 1987. Morgan Kauffmann Publishers. 340-350.

Sagols, F. D. Linear Representation of Solids. M. Sc. Thesis, Electrical Engineering Department, Center for Research and Advanced Studies, National Polytechnic Institute of Mexico. 1987 (In Spanish).

Shaw, A. C., and Miller, W. F. Linguistic Methods in Picture Processing a Survey. *Proc. AFIPS FJCC* Vol. 33, 279-290. 1968.

Table of Contents

Chapter 1. Introduction and Definitions 2
Section 1. Definitions 2
Figure 1.1. 1. Direction of Turns 2
Figure 1.1. 2. Examples of Limb Strings 3
1. Notation for turns 4
Figure 1.1. 3. Invariance when going or coming back 4
Figure 1.1. 4. Slight anomaly 4
2. The name of a turn does not change if traveled backwards 4
Chapter 2. The Shape Descriptor for Limb Strings 5
Section 1. Additional properties of limbs and junctions 6
Figure 2.1. 1. Shape Descriptor for a Limb String 6
Figure 2.1. 2. Some Bodies with Three Limbs 6
1. Shapes for end junctions 7
Figure 2.1. 3. The two Possible Bendings for Bodies with Two Limbs 7
2. The Descriptor for Bodies with Three Limbs 7
3. The Descriptor for Bodies with Two Limbs 7
4. The Descriptor for Bodies with One Limb 8
5. Taking Size into account 8
Figure 3.1. 1. Limb Trees 8
Chapter 3. The shape descriptor for Limb Trees 9
Section 1. Unique Traversal of the Tree 9
1. Privacy numbers 9
Figure 3.1. 2. In What Direction to go? 9
Figure 3.1. 3. A Limb Tree that needs Search 10
Section 2. Constructing the Descriptor for the Limb Tree 10
Figure 3.2. 1. Limb Tree and its Descriptor 10
Figure 3.2. 2. Additional Examples of Shape Descriptors 10
1. Use of wiggly lists 11
Chapter 4. Conclusions and Suggested Further Work 12
1. Conclusions 12
2. Suggestions for Further Work 13