REPRESENTATION AND DESCRIPTION OF CURVED OBJECTS

Gerald Jacob Agin

Stanford University

Prepared for:

Advanced Research Projects Agency

October 1972

STANFORD ARTIFICIAL INTELLIGENCE PROJECT
MEMO AIM-173

STAN-CS-72-305

REPRESENTATION AND DESCRIPTION OF CURVED OBJECTS

BY

GERALD JACOB AGIN

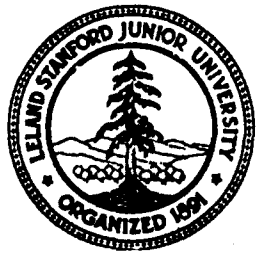
SUPPORTED BY

ADVANCED RESEARCH PROJECTS AGENCY

ARPA ORDER NO. 457

OCTOBER 1972

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY



STANFORD ARTIFICIAL INTELLIGENCE PROJECT
MEMO AIM-173

COMPUTER SCIENCE DEPARTMENT
REPORT CS-305

OCTOBER 1972

REPRESENTATION AND DESCRIPTION OF CURVED OBJECTS

by

Gerald Jacob Agin

ABSTRACT

Three dimensional images, similar to depth maps, are obtained with a triangulation system using a television camera, and a deflectable laser beam diverged into a plane by a cylindrical lens.

Complex objects are represented as structures joining parts called generalized cylinders. These primitives ae formalized in a volume representation by an arbitrary cross section varying along a space curve axis. Several types of joint structures are discussed.

Experimental results are shown for the description (building of internal computer models) of a handful of complex objects, beginning with laser range data from actual objects. Our programs have generated complete descriptions of rings, cones, and snake-like objects, all of which may be described by a single primitive. Complex objects, such as dolls, have been segmented into parts, most of which are well described by programs which implement generalized cylinder descriptions.

This research was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense under Contract No. SD-183.

ACKNOWLEDGEMENTS

I would like to express my appreciation for the guidance and assistance of my advisor, Dr. Thomas O. Binford. His insight has been helpful on many occasions. Many of the original ideas on which this research is based were his, including generalized translational invariance, and the basic configuration of a laser ranging system.

The International Business Machines Corporation has supported me and my family during my three years at Stanford. Without their aid the rewarding experience of attending school and persuing a doctoral program would have been extremely difficult, if not impossible.

Thanks are also due to Dr. Arthur L. Schawlow for the generous loan of a laser, to Victor Scheinman for the design and assembly of the laser deflection apparatus, to Richard Underwood for routines for the numerical solution of the generalized eigenvalue equation, and to Bruce G. Baumgart, R. K. Nevatia, and J. M. Tenenbaum for their helpful comments and suggestions.

TABLE OF CONTENTS

CONTENTS

1 INTRODUCTION

Our present interest in representation and description of curved objects arose from a desire to extend the capabilities of the Stanford Hand-Eye System [Feldman]* to recognize a wider class of objects than plane-bounded solids. Our initial attempts to recognize cones, cylinders, and spheres were not carried far enough to demonstrate the usefulness of existing techniques in recognizing this limited addition to the class of recognizable objects. But there appears to be no insurmountable barrier to doing so.

There have been few other attempts to recognize curved objects. B. K. P. Horn [Horn] reconstructed the depth contours of three dimensional surfaces from their photographs, based on assumptions about the light reflection characteristics of the surface. Krakauer [Krakauer] was able to distinguish among several kinds of fruit by examining the connectivity and structure of brightness contours in a two-dimensional image.

It soon became apparent that little useful purpose would be served by these techniques. A significant improvement in performance of vision systems would come only when they were capable of recognizing the sort of everyday objects that a robot of the future might have to deal with. To accomplish this, we make use of two new tools or techniques.

The first of these is a representation general and flexible enough to carry varied information about an object, its parts, and their relation to one another. The representation we propose will allow several different models for primitives or parts of objects. These models may include prototypes of various sorts. The particular prototypes we employ, generalized cylinders, are useful for describing a large class of natural and man-made objects. Generalized cylinders consist of a space curve or axis and a set of cross sections described on this axis. They describe in a natural and intuitive way pieces which possess elongation, or which have axial symmetry. The generalized cylinders may be linked together in various ways to form complex objects. The most significant departure from previous methods of description is that the representation is essentially a volume representation, as opposed to a surface representation. Chapter 2 of this report describes in detail representation by generalized cylinders, and gives some examples of how objects might be modelled with such a representation.

---

* Names in square brackets denote references which may be found at the end of this report.

The second new technique is the use of direct depth measurement for recognition. Every two-dimensional image of a three-dimensional scene is ambiguous, in that there are an infinite number of realizable physical objects (or groupings of objects) which could give rise to the image. These ambiguities may be resolved only by a priori assumptions about the nature of the scene. The use of direct depth information eliminates the need for assumptions such as squareness of corners, requirements that stored prototypes exist for every possible object in a scene, or knowledge of the reflective characteristics of surfaces. Chapter 3 describes a ranging system, which obtains depth information by triangulation, using a laser and a television camera.

The test of any vision system must be how well it deals with actual objects. Chapters 4 and 5 describe an experimental system to generate descriptions of physical objects, from depth data derived from the ranging system. The system has so far given good results in describing some simple curved objects, and has been moderately successful in describing parts of complex objects. Some results may be seen in Figures 5.15 through 5.28 at the end of Chapter 5. Chapter 6 contains some suggestions for further research.

So far we have not achieved our goal of recognition of complex objects; we cannot yet identify a given object, say, as a humanoid figure or as a hammer. But we hope our method of representation and our work in description of objects has laid the groundwork for progress in this area.

2 REPRESENTATION

A fundamental problem in the design of intelligent machines and computer programs is representation. Problem solving requires facts about the problem domain. Only a finite number of facts may be stored directly in a machine's memory; other information which the machine requires must be deduced or inferred from the available data. We shall address ourselves specifically to information about solid, three dimensional objects. The question which must be answered in the preliminary design stage of an intelligent machine is, then, what types of data will be stored, and how should it be organized in order to obtain other useful information?

Frequently, problem solving requires a transformation or a series of transformations between different ways of representing the same data. For the robot which must acquire information about the external world through sensors, this means going from the raw input from the sensors into a set of facts upon which decisions may be based. We use the word description to denote the process of deriving a high level representation from low level data. We may also call the data structure so derived a description.

Representation and description are intimately related. The form of the representation dictates the description strategy. A useful representation makes description easy. But a useful representation must also provide ways of dealing with and manipulating its descriptions. It must provide us with the kinds of information we really want for decision making. Intelligent machines will utilize this information for recognition and classification, for manipulation by external machinery, for man-machine communication, and for graphical display or drawings. Section 2.1 spells out some desirable characteristics of a representation for decision making.

Many higher level representations involve the use of models. Models are the atoms or primitives of which descriptions are composed. Usually the models themselves possess some structure of their own. Descriptions are built up by specifying the relationships between primitives. Specifying a representation involves a choice of what models to make use of, and how they are to be interrelated. In Section 2.2 we outline the structure of a new representation making use of generalized cylinders and other models. Generalized cylinders are treated in Section 2.3.

Other representations have been used and are being used in intelligent machines for describing solid objects. In Section 2.4 we compare some of these with representation by generalized cylinders, and explain how they fail to meet the requirements we have set forth.

2.1 REQUIREMENTS FOR A REPRESENTATION

A general purpose intelligent machine should be capable of dealing with a wide variety of information. Specifically, it should be able to recognize and manipulate a wide class of objects, assess the similarity between two or more objects, and determine the relationships among objects within its universe. To design such a machine, it is necessary to develop some concepts of the "shape" of a three-dimensional object.

A primary attribute of shape is structure. Structure specifies the major features of an object, and how major features and minor details are related. Some other important attributes of shape are topological connectedness, elongation, length-to-width ratios, presence and orientation of edges, and symmetry.

An adequate representation of shape should be capable of segmenting an object into parts. Parts may be described independently, then the relationship of the parts may be specified to form objects. Segmentation is essential in the description of complex objects. It allows a small number of basic parts to be linked in a large variety of ways. The structure of a complex object is then contained in the relationships between parts. If the segmentation is done properly, the parts will possess simpler descriptions than the object would without segmentation. When the parts may be further segmented into sub-parts, a hierarchical representation is obtained.

A hierarchical representation also allows a variable degree of detail to be studied. Detailed information should be available for comparison between objects which are nearly alike, but this detailed information should not be allowed to hide the relavent structural information when the detailed information is unneeded.

Yet a hierarchical organization does not completely solve the problem of making available a variable degree of detail. In general, some redundant information must be stored (or generated). A given part might contain several descriptions,

successive approximations giving greater and greater precision. The grosser descriptions will not contain as much information as the finer ones, so the additional storage to keep the grosser ones will not be great.

Comparison between objects must be allowed, and the degree of similarity assessed. Comparison should be on the basis of structural similarity first, and then on degree of similarity between parts of the structure.

Many natural and manufactured objects possess elongation. Most higher orders of life are distinguished by their extremities -- legs, arms, heads, stems, and branches. The shape of man-made objects are frequently dictated by the shape of their raw materials, their method of manufacture, and their function. And where elongation is present, the direction of the elongation usually bears some useful or functional relationship to the object as a whole.

2.2 A HIERARCHICAL VOLUME REPRESENTATION

We present here a hierarchical representation which we believe to contain most of the desirable features set forth in the preceding Section. The representation is essentially a volume representation, which makes use of armlike primitives, and builds descriptions by relating the primitives and their axes in various ways. The primitives are called generalized cylinders, which are capable of describing "snakes" of arbitrary cross section. The representation may be extended by the addition of other primitives.

What we outline in this chapter is a framework for representation, which hopefully will model most of the objects with which a robot might have to deal. Only a small subset of the representation has actually been implemented in our programs and experimental work.

Let us begin by describing a human figure. The significant structural information may be conveyed in a "stick figure", such as is shown in Figure 2.1. The six main parts which make up the figure (the body, two legs, two arms, and the head) are represented as lines or axes. The structure which the axes make up we call the skeleton of the object. The structure of the figure is apparent from the way these axes are related. Most people would have little difficulty recognizing the figure as representing a man.

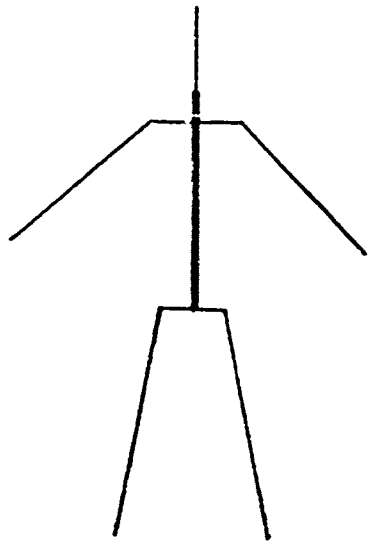


Figure 2.1
Stick Figure of a Man

The line representing the body is drawn broader than the lines of the other parts. We denote that line as the principal axis of the figure. The other axes join to the principal axis, and are represented as subordinate axes. A principal axis is not always definable in an arbitrary object or part, but when it can be defined, it serves to provide emphasis to a description.

We depend on segmentation to divide the picture into pieces which may be simply represented. If desired, each part may be further segmented into smaller pieces to show a greater degree of detail. Figure 2.2 shows how we might represent the arm of the stick figure. The upper arm and the forearm constitute the principal axis of the arm. The legs may be similarly segmented and represented.

What the lines of the figure actually represent are solid volumes. We represent the volume by a cross section function on the axis. A screwdriver may be represented by a simple straight line skeleton, with a cross section function such as that indicated in Figure 2.3. A formal description of how volumes are generated within our description follows in the next Section.

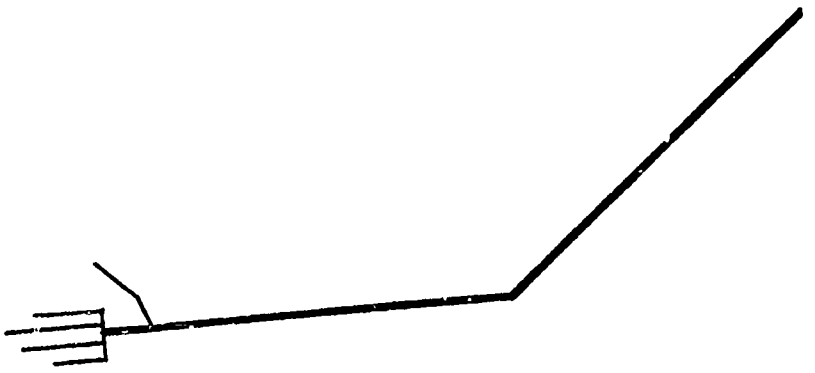


Figure 2.2
Stick Figure of an Arm

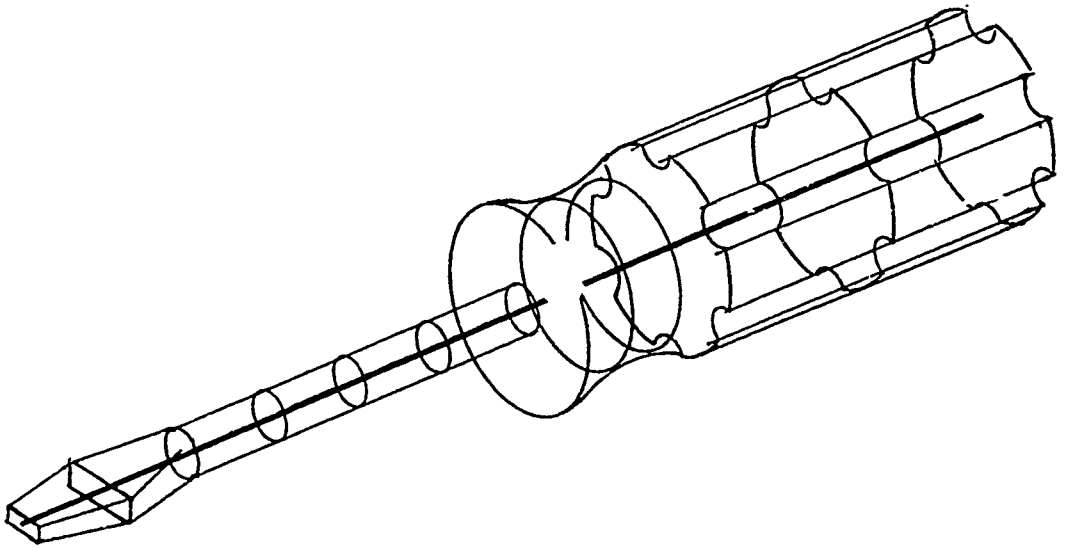


Figure 2.3
Representation of a screwdriver

Although we represent volumes, for a structural description it is convenient to find the center lines of elongated parts and refer to these instead of to the more complicated descriptions of the volumes themselves. These axes may be arbitrary space curves in three dimensions, as dictated by the volumes they represent.

The places where axis segments meet we call joints. Joints are described by the number of parts being joined, the orientations of the of the segments, which of the segments contains the principal axis of the assembly, and whether the joint is fixed or movable.

Joints in which one part may move with respect to the other we call articulated. Three common kinds of articulated joint are a pin joint, which allows lateral motion in one dimension; a ball-and-socket joint, which allows lateral motion in two dimensions, and a rotating joint, in which rotational motion of one part with respect to the other is allowed. The most general type of joint is both a ball-and-socket joint and a rotating joint.

The data structure of our representation will account for a general type of joint by storing the appropriate information about each part comprising the joint. But there are exactly three types of joint which recur so frequently that we give them special names.

The simplest type of joint joins two parts end-to-end. This type of joint is best exemplified by an elbow, as shown in Figure 2.4. The principal axis of the assembly contains the principal axes of both parts. Although almost every end-to-end joint could be eliminated by describing the two parts together as a single generalized cylinder, the end-to-end joints are useful where there is a discontinuity in cross section, a sharp bend or change in curvature of the axis, or a movable connection.

The second common type of joint occurs when the end of one part joins the middle of another. We call the first part the crossbar, the second the upright, and the connection a T-joint. The thumb of Figure 2.5 is a good example. The principal axis of the assembly generally follows the crossbar. Usually the cross section of the upright (perpendicular to the plane of the T) will not exceed the cross section of the crossbar.

The third type of joint, which we call a bifurcation, occurs when one part meets (or divides into) two subordinate parts. A good example is found where the thighs meet the hips. The

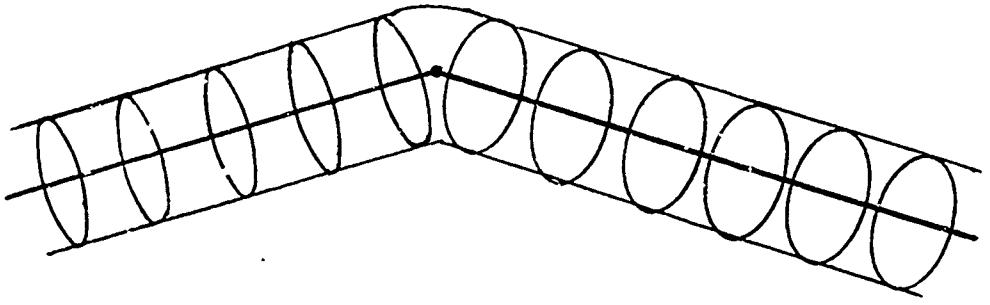


Figure 2.4
End-to-End Joint

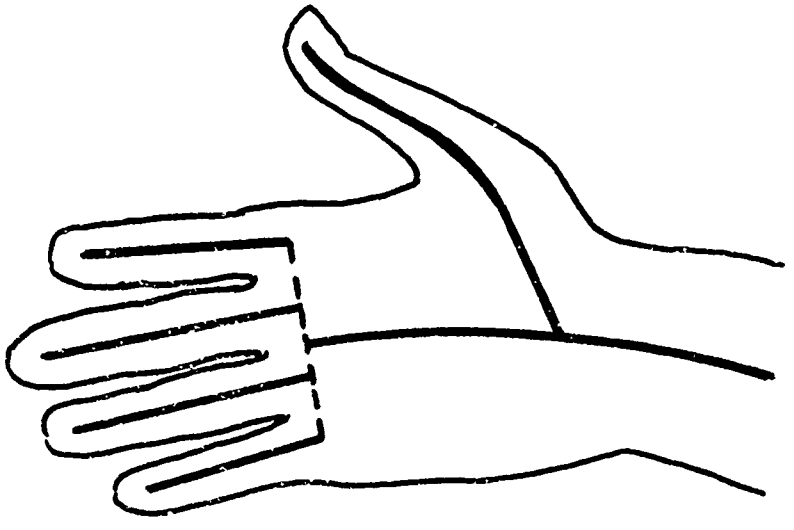


Figure 2.5
A T-Joint and a Multifurcation

subordinate branches are approximately parallel to each other and to the principal branch. A conservation of cross section holds at a bifurcation; the cross sections of the subordinate branches, together, should roughly equal the cross section of the principal branch. Sometimes the principal part will divide into more than two subordinate branches; in this case we call the joint a multifurcation. Figure 2.5 also contains an example of a multifurcation.

The description process may be likened to cutting and pasting. The image of an object is cut into parts which may be described as generalized cylinders. Holes, notches, and concavities are cut out from the primitives. The pasting operation will put the pieces together into a framework which describes the structure of the object.

Our representation is non-unique. Segmentation is carried out on the basis of what results in the simplest description, and depends on heuristic choices of simplicity and local models. Routines which compare descriptions of objects must therefore take into account the various ways which segmentation might be applied.

Figure 2.6 shows two different ways in which a disk may be represented. This illustrates the importance of symmetry in selecting an axis. If we use elongation as the sole criterion for determining the direction of the skeleton, the representation on the left of Figure 2.6 is obtained. Clearly the representation on the right is preferable, since axial symmetry allows a simpler description.

A shallow dish or bowl is more difficult to represent, since if the axis of symmetry is to be the skeleton of the object, then the cross section is not a simple curve. The use of a hollow, thin-walled cross section, will provide a simple representation in this case.

Generalized cylinders are not capable of representing concisely an arbitrary degree of detail. A hierarchical approach can improve the level of detail to which objects can be represented, but beyond a certain point the simplicity of the method is lost. It would certainly be possible to generate a human face with generalized cylinders, but to describe a face to the level of detail necessary to distinguish one face from another would exceed the limits of practicality.

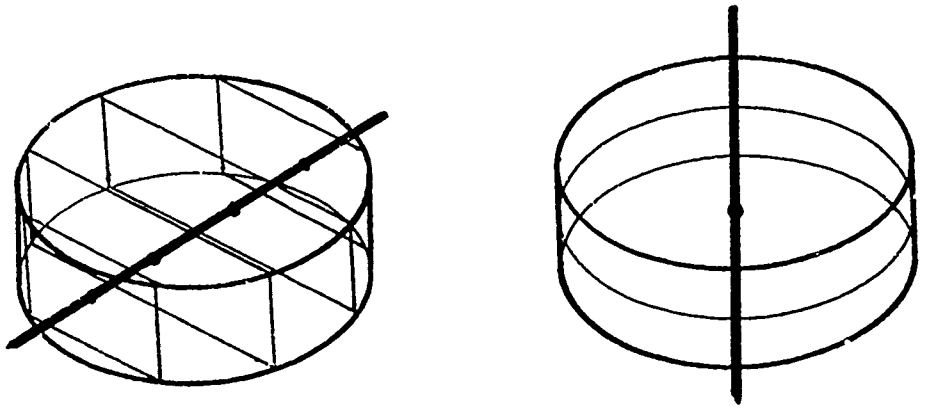


Figure 2.6
Two Representations of a Disk

There are shapes which are not very well represented by our methods. Objects formed from sheet metal are a frequently occurring example. For this reason we allow other primitives to be added. The main emphasis of this thesis is on generalized cylinders, but we may take a little time to suggest what other primitives might be helpful.

Thin-walled, hollow cross sections will be useful in describing a number of objects, principally containers of various sorts. This is easily accomplished in our representation. But to include a wider class of thin shapes, we should allow a new primitive consisting of flat sheets. A strictly two-dimensional representation of sheets will not be useful because it cannot account for bending or warping of the sheets. Flat sheets may be represented in a manner similar to the way we model generalized cylinders, except that the cross sections become one-dimensional.

Negative, or subtractive volumes can make descriptions much simpler in modelling holes and other concavities in objects. Truncating planes and surfaces will be useful in describing hemispheres and the like. These are a special case of a subtractive volume, where only one surface of the truncating volume need be described.

The strength of this method of representation lies in segmentation which results in simple descriptions of the major features and proportions of man-made and natural shapes, and in the usefulness of the descriptions in inferring shape and function. The segmentation depends on a class of primitives which allows a natural part-whole decomposition and which suggests simple approximations.

2,3 GENERALIZED TRANSLATIONAL INVARIANCE AND GENERALIZED CYLINDERS

To describe the snakelike primitives of our system, we make use of the concept of generalized translational invariance, first formalized by T. O. Binford [Binford 71], and extended here.

The model we propose consists of:

1. A space curve, called the axis.

2. Planes perpendicular to the axis, and a set of axes on the plane.

3. A variable cross section described on these axes.

These primitives we call generalized cylinders.

A simple cylinder may be generated as follows: Suppose there is an arbitrary simple closed curve lying in an arbitrary plane, as shown in Figure 2,7. If the plane is translated in a direction normal to the plane, the locus of the curve traces out a portion of a surface. The surface so traced is a cylinder (in the usual geometric sense of the word). With suitable terminating surfaces at the ends, this process can generate a solid body enclosed by the surface.

We can formalize the generation of simple cylinders as follows: Let $\vec{C}$ denote the curve, $\vec{P}$ the plane in which it initially lies, and $\vec{p}$ the direction normal to $\vec{P}$. Let $T_{(\vec{p},a)}$ be a translation operator which we may apply to space curves, such that $T_{(\vec{p},a)}*\vec{C}$ represents the curve $\vec{C}$ after it has been translated a distance a in the direction $\vec{p}$. The volume $T_{(\vec{p},s)}*\vec{C}$ $(0\leq s\leq\lambda)$, the locus of the area inside $\vec{C}$ as it is translated a distance λ

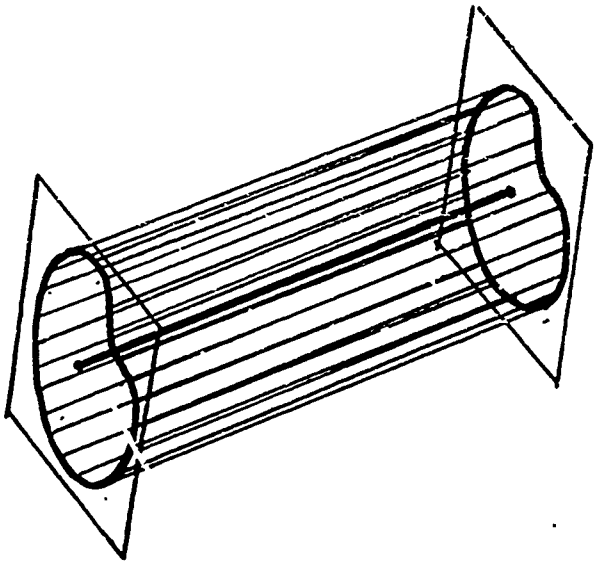


Figure 2.7
A Cylinder

in the direction normal to $\vec{P}$, is a cylinder. If $\vec{D}$ is some distinguished point on $\vec{P}$ associated with C (such as the center) then $T(\vec{p},s)*\vec{D}$ $(\emptyset \leq s \leq \lambda)$ represents a straight line in space which we call the "axis" of the volume.

We generalize this "simple" cylinder in two ways. The first is to allow the direction of translation to vary. The second is to allow the shape of the cross section to vary.

Generalized translational invariance, as formulated by Binford, generalizes both the translation and the cross section variation by including in the operator T a congruence transformation on cross sections. That is, the operator will transform the translated cross section to force a congruence with the normal cross section at the translated position. This is basically a descriptive technique which will be useful if we can find an axis choice such that the congruence transformations are sufficiently simple. It is this technique which we follow in the computer description of primitives set forth in Chapter 5. But for a higher level representation of complex objects we prefer

the generational technique which follows, We believe that the separation of the axis and the cross section descriptions gives a more general and more intuitive representation of shape.

Allowing the direction of translation to vary will allow us to generate "snakes" of constant cross section, as shown in Figure 2.8. In our computer-implemented representation, the axis is represented as a sequence of points. But in general, we wish to describe the axes as continuous space curves. We choose helices to represent space curves of constant curvature; if the curvature of a curve varies, it may be approximated by a sequence of segments of the proper curvature.

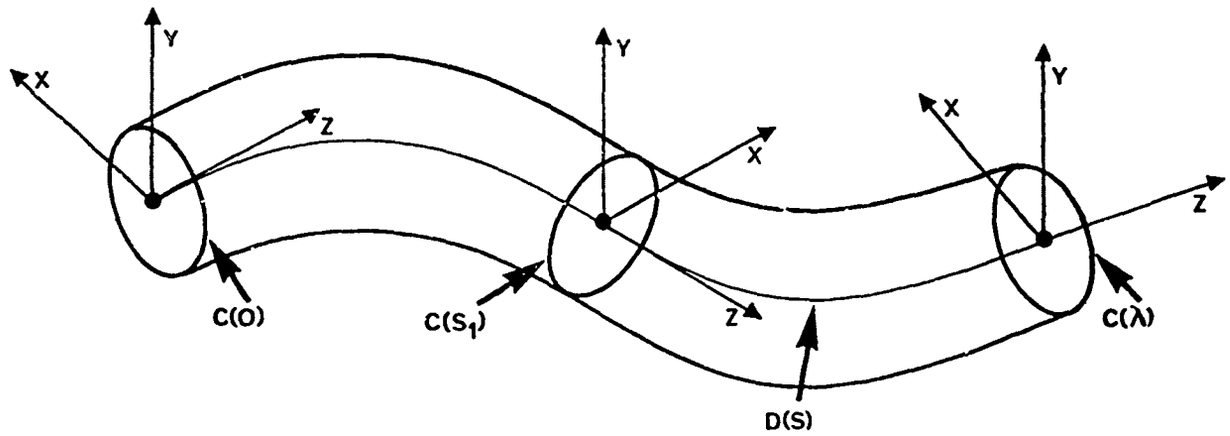


Figure 2.8
Generalized Cylinder with Constant Cross Section

To describe helical axes, it is convenient to introduce an auxiliary coordinate system, A, centered on the distinguished point, D. The curvature of the axis will be described relative to this coordinate system. The direction of translation will be the z-axis of A. The curve C lies in the x-y plane of A. The orientation of A with respect to the rest of the world will be some function of arc length, s, along the axis. That is, the orientations of the x-, y- and z-axes of A are the (mutually

orthogonal) vector-valued functions $\vec{x}(s)$, $\vec{y}(s)$, and $\vec{z}(s)$. An axis $\vec{D}(s)$ may be represented as the solution to the difference equation $\vec{D}_{new} = T * \vec{D}_{old}$, or

$$\vec{D}(s+ds) = T(\vec{z}(s),ds) * \vec{D}(s) \qquad \text{[Equation 2.1]}$$

in the limit as ds approaches Ø. Or equivalently,

$$d\vec{D}(s)/ds = \vec{z}(s). \qquad \text{[Equation 2.2]}$$

If we represent the curve C parametrically in the x-y plane of A (although we will not necessarily depend on a parametric representation) we might choose the form

$$\vec{x}=f_x(t),\ \vec{y}=f_y(t),\ (Ø \le t \le t_{max})$$

where t is arc length along the curve. Then the surface C(s,t) may be represented in three dimensions as

$$C(s,t) = \vec{D}(s) + f_x(t)\,\vec{x}(s) + f_y(t)\,\vec{y}(s), \quad (Ø \le t \le t_{max}). \qquad \text{[Equation 2.3]}$$

The "generalized cylinder" is bounded by the locus of C(s) as s varies from Ø to λ.

Three independent derivatives are necessary to describe the motion of D and its coordinate system. ([Coxeter] contains a discussion of the generation of helices.) We choose torsion, τ (tau), curvature K (kappa), and an angle θ (theta) denoting an axis of curvature in the x-y plane, such that the following relationships hold:

$$d\vec{x}/ds = -K\,\vec{z}\cos\theta + \tau\,\vec{y} \qquad \text{[Equation 2.4]}$$

$$d\vec{y}/ds = -K\,\vec{z}\sin\theta - \tau\,\vec{x} \qquad \text{[Equation 2.5]}$$

$$d\vec{z}/ds = K\,(\vec{x}\cos\theta + \vec{y}\sin\theta) \qquad \text{[Equation 2.6]}$$

Figure 2.9 shows a helical skeleton and a geometric interpretation of torsion and curvature for the case where $\theta = 0$. Pure torsion results in rotation of A about its z-axis. Pure curvature results in rotation of A about the axis of curvature.

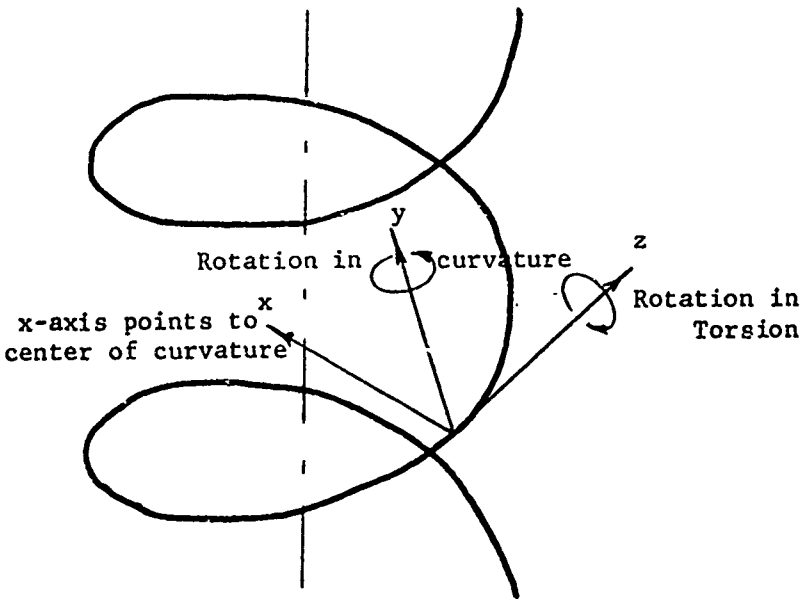


Figure 2.9
Torsion and Curvature of a Helix

Only an initial position and orientation for the auxiliary coordinate system A, plus our three derivative parameters, are necessary for Equations 2.4 through 2.6 to generate space curves which are segments of arbitrary helices. If τ is zero, a torus is generated. If K is zero, the equations generate a straight line.

The variable θ is redundant in this description. It is possible to specify the orientation of the auxiliary coordinate system A such that the center of curvature always lies in the x-z plane. The additional variable allows us to specify the cross section shapes more economically.

The second way we generalize our formal model is to allow the shape of the cross section to vary. Let C(s) be a cross section-

valued function of s. That is, given some specific value for s, the function C(s) will tell us how to construct a closed curve (or a family of non-intersecting closed curves) on the x-y plane of A. In specifying the form of C(s), we have chosen to use a small set of regular geometric shapes, with variable scalar parameters to specify size, proportions, or rotation. These parameters may be functions of arc length s along the axis, so that we may generate cones, drills, or piano legs.

Representation of cross section shapes has not been the attention it deserves in this research. Describing cross sections is a representation problem in two dimensions, and many of the comments we have already offered on representation might be applied here as well. In particular, segmentation is important in describing complicated cross sections. Our formal model is capable of extension by the addition of other means of specifying cross section curves. Polar functions (radius vs. angle) and other parametric representations of the cross section outline can provide a very general descriptive power. But these are not entirely satisfactory since their constructs do not provide an intuitive description of shape.

An anomaly arises when the axis of a cylinder curves with a radius of curvature less than the "inside" radius of the cross section. Such a situation is illustrated in Figure 2.10. The surface described by the cross section's outline is convolute. But the volume of the solid is the locus of the interior of the cross section. The fact that this locus crosses itself should introduce no difficulty. The surface of the object in the vicinity of the bend will be those points of the object which are adjacent to its exterior.

With a sufficiently general cross section representation, generalized translational invariance will generate any realizable object, in a trivial but uninteresting way. Let the axis be a straight line, and describe the cross sections cut by planes perpendicular to the axis as a function of length along it. The description of these cross sections will not, in general, be easy or elegant. The utility of the method of description will lie in its simplicity and its usefulness in extracting significant features. The fundamental question is whether we can do this simply for an interesting class of objects.

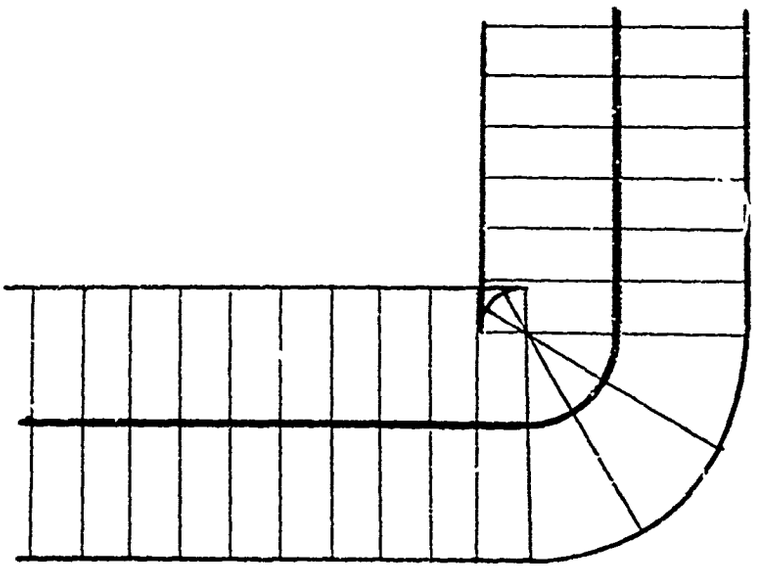


Figure 2.10
Surface Anomaly

2.4 COMPARISON WITH SOME OTHER METHODS OF REPRESENTATION

Many ways of representing three-dimensional objects exist. Different representations make different kinds of information available, and achieve different degrees of success in presenting this information in a useful manner. We shall mention some of the more significant of these, and compare them with representation by generalized cylinders with respect to how well they describe those attributes we call shape.

Among the most basic methods of representing solid objects are the space map and the depth map. A space map is a three-dimensional array corresponding to a subdivision of space into contiguous cells. With a coarse cell size, this model is useful in obstacle avoidance, or in identification of subareas of special interest in a larger scene. A depth map is a two-dimensional array containing the distances from a camera or ranging device to the nearest surface of an object or scene. The low-level input to many programs is organized in this fashion. (A variation on the depth map is the depth grid, described in

Section 3.2.) While space maps and depth maps are convenient ways of organizing low-level data, they have little applicability to shape description.

Some surface description methods may be classified with depth maps, when the description substitutes depth contours or analytic functions for the two-dimensional depth array. Bivariate spline fitting [Coons] fits into this category. Horn created a depth contour representation from assumptions about illumination and reflectance characteristics [Horn].

A common representation uses a body-face-edge-vertex model. Roberts [Roberts 63] and the Stanford Hand-Eye system [Paul] make use of prototypes where the relationships of faces to edges and vertices is defined in advance for a fixed number of primitives. Some more flexible models have been used by other researchers,[Shirai, Baumgart]. This type of model is mainly suited to polyhedra, but approximation of curved objects by a faceted representation has had some success in computer graphics. This extension of what is essentially a method for plane-bounded objects into the domain of curved objects cannot be seriously considered a reasonable candidate for shape description. The body-face-edge-vertex model describes surfaces, not volumes. And the number of planes to be dealt with makes the method unwieldy for curved objects.

Medial axis transforms [Blum, Mott-Smith] bear some superficial resemblance to representation by generalized cylinders. We shall devote a paragraph to explaining how medial axis transforms are derived, then show why they do not provide an adequate description of shape.

In two dimensions, the medial axis transform is sometimes called a Blum transform. The Blum transform for a given shape is generated as follows: Associated with every point in the interior of a (two dimensional) shape or outline, is a maximal disc neighborhood. Figure 2.11 shows some points in the interior of a rectangle, and their maximal neighborhoods. The maximal neighborhood of any point is the largest disc which may be centered on that point and still remain within the boundaries of the outline. The Blum transform of the shape consists of the centers of the maximal neighborhoods which are not wholly contained in any larger neighborhood. In Figure 2.11 the neighborhood of point F is wholly inside that of point E, and that of point C is wholly inside that of point F. Only the neighborhoods of points A, B, and E cannot be wholly contained in any larger maximal neighborhood, and those points will be part of

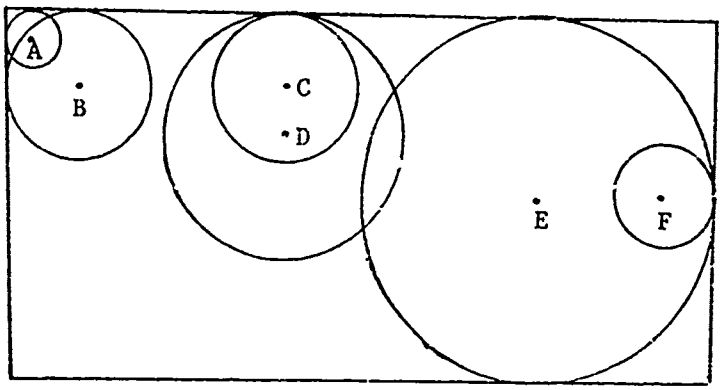


Figure 2.11
Maximal Neighborhoods

the Blum transform of the rectangle. The complete Blum transform of a rectangle is shown in Figure 2.12.

When the Blum transform is extended to three dimensions, the transforms become not space curves, but portions of curved surfaces in space. The medial axis transform of an ellipsoid is an ellipse-shaped piece of a flat surface, centered in the ellipsoid.

The principal objections to medial axis transforms are that if segmentation is applied, it must be done in a rigid manner, that generation of the transforms requires knowledge of the complete object, and that the transforms it generates are non-intuitive.

Segmentation may be applied to a medial axis transform, after it has been computed, at branches or forks in the transform. Suitable conditions must be established to differentiate those branches of the transform which represent extremities of the object from those which arise from corners. What we really want from a representation is the ability to segment first, and then describe each part, and that the segmentation be applied so as to yield the simplest description of each part. With medial axis

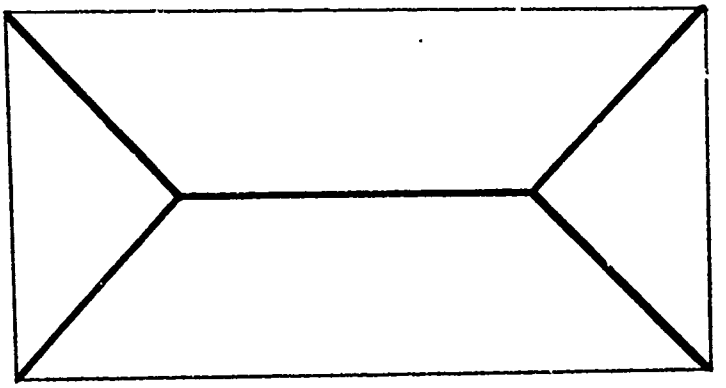


Figure 2.12
Blum Transform of a Rectangle

transforms, segmentation before description will create extra corners on each part, which will introduce spurious branches on the transform.

The entire extent of an object must be known in order find the maximal neighborhoods. This precludes use of the method on objects obtained from only one view. Two dimensional shapes are usually viewed in their entirety, but in three dimensions the entire surface of an object is difficult to obtain.

A non-intuitive two-dimensional medial axis transform may be seen in in Figure 2.13, the Blum transform of a rectangle with a notch. A minor variation in the outline produces a major perturbation in its transform. In three dimensions the situation is analogous. The medial axis transform of a coin or flat disc will be a circular portion of a plane with rim-like extensions at its circumference.

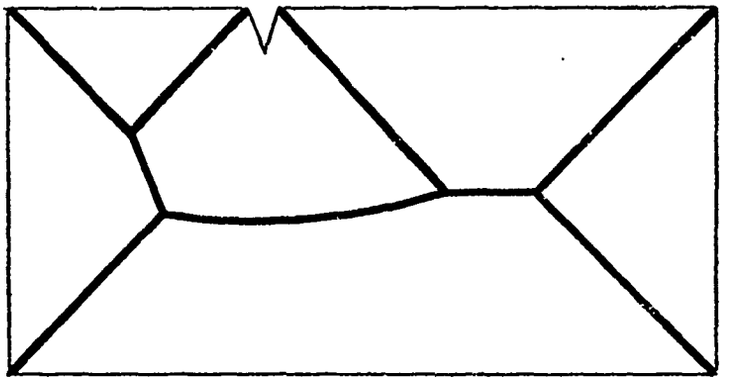


Figure 2.13
Blum Transform of a Rectangle with a Notch

3 DEPTH MEASUREMENT

The recognition and representation of objects as performed in the experimental portion of this research requires three-dimensional data on which to operate. The primary requirement is that the data be reasonably dense and reasonably consistent. While the special characteristics of our laser triangulation system are made use of in many of the techniques to be described, we believe our methods are general enough that other means of ranging can be substituted.

3.1 SOME METHODS OF DEPTH MEASUREMENT

A fairly comprehensive catalogue of methods of depth measurement is given in Chapter 7 of [Earnest]. Only the more suitable of these are discussed below.

Devices exist which are capable of directly measuring the distance from the device to some point on an object placed before it. All of these devices are variations on the basic method of time-of-flight measurement of light. For the distances in which we are interested, this usually takes the form of a laser beam modulated by a sinusoidal signal, and a detector and phase measuring circuit which determines the phase shift of the reflected light with respect to the emitted light. An example of such an instrument is the Geodolite, manufactured by Spectra-Physics, Mountain View, California. Its depth resolution of 1 millimeter is probably adequate for our purposes. Direct ranging devices require a two-axis deflection system (usually a pair of rotating mirrors) in order to scan a scene. The response time of the Spectra-Physics Geodolite is one millisecond. With a properly designed mirror scanning system, it would require only 90 seconds to scan an entire scene with a raster resolution comparable to our television cameras. At present, its cost is prohibitive, compared with other methods available. As techniques in this area improve, direct ranging may become competitive with other ranging methods.

Two-camera stereo is attractive mainly from the point of view that it imitates human stereo depth perception. While research using two-camera stereo may shed light on human depth perception (or, more likely, stimulate further research in this area) we feel two-camera stereo is hardly the best way to measure depth by computer when we are interested in speed, efficiency, or accuracy. Triangulation by laser (see the next Section) requires one TV camera. If other characteristics of an object, such as

color or texture are required, a separate TV image of the scene may be obtained with little additional cost. In general, it is desirable to have alternate modes of perception for varying tasks.

To measure depth by stereopsis it is first necessary to identify points in each image which correspond to the same point on the actual object. Either some preliminary recognition must be performed on the scene, or correlation must be performed on the fine texture of the scene. To use a higher level analysis (such as a sort of low-level recognition) to control the acquisition or processing of low-level input is attractive as a goal for future research, but to this date such techniques have not been demonstrated. Correlation of texture either restricts us to coarse textured objects or requires a much higher spatial resolution than is currently available in imaging devices.

R. K. Nevatia has used motion stereo with texture correlation to measure depth at selected points on the surface of rocks. His methods yield a depth accuracy similar to that of our laser triangulation system, but the average processing time he estimates to be about 10 seconds per point.

Triangulation using a beam or plane of light and an imaging device, as outlined in the following Section, appears to be the best practical means of three-dimensional scene acquisition available at present.

3.2 TRIANGULATION BY LASER

Triangulation by laser, (for the case where the laser beam is not diverged,) is geometrically similar to stereo. Consider replacing one stereo camera by a deflectable laser beam. The horizontal and vertical deflection angles of the beam correspond to the raster coordinates in the camera. The problem of identification of a single point in the two "views" is practically eliminated, since in the remaining TV image the bright laser spot is easily detected.

Data rate for triangulation by an undiverged beam would be rather low, since for each point measured the laser must be deflected, the TV camera must be read, and the bright spot identified. The maximum data rate using an undiverged beam would be 30 to 60 data points per second, based on the time required to read one TV image.

A significant improvement in data rate is obtained by diverging the laser's pencil beam into a plane of light, as, for instance, by passing the beam through a cylindrical lens. Such a lens will magnify or spread the beam in one dimension only, transforming the circular laser beam into an elongated ellipse, or plane of light. The angle of divergence of the plane of light will be equal to the diameter of the undiverged beam, divided by the focal length of the lens. An additional advantage of diverging the beam is that only one rotating mirror will be necessary to enable the beam to cover every part of a scene.

To see how depth may be measured with a plane of light, it may be instructive to think of the plane as being composed of many individual rays of light emanating from a point. As long as the rays do not cross or coincide in the camera's image, each ray is identifiable. This restriction is equivalent to the condition that the plane of light (and its infinite extension) not include the focal point of the camera.

The only visible illumination on the scene is in the plane of light; hence all illuminated points are on the plane. There exists a unique collineation (one-to-one correspondence) between points on the plane and points in the TV image. Once the collineation is known, the three dimensional coordinates of any illuminated point may be determined from its location in the TV image.

The depth accuracy of a triangulation system depends on the resolution of the imaging device and on the angle of separation between the two points of view. (Refer to Figure 3.1.) The inherent resolution of an imaging device gives rise to a cone of uncertainty for any given point in an image. The width of the cone at the object being viewed we call D. The uncertainty in lateral position because of the resolution of the imaging device is D / 2. If the angle of separation between the camera and the laser is Φ, then the maximum uncertainty in position is $\frac{D/2}{\tan \Phi}$.

The root-mean-square uncertainty will be 0.707 times this, or

$$\langle \text{RMS range error} \rangle = 0.353\ D \cot \Phi. \qquad \text{[Equation 3.1]}$$

If the width of the laser plane is more than one raster unit, then the uncertainty D should also include the uncertainty in locating the centerline of the plane. For a typical configuration of our ranging system, the contribution to D due to the resolution of the TV camera is about 0.05 inches.

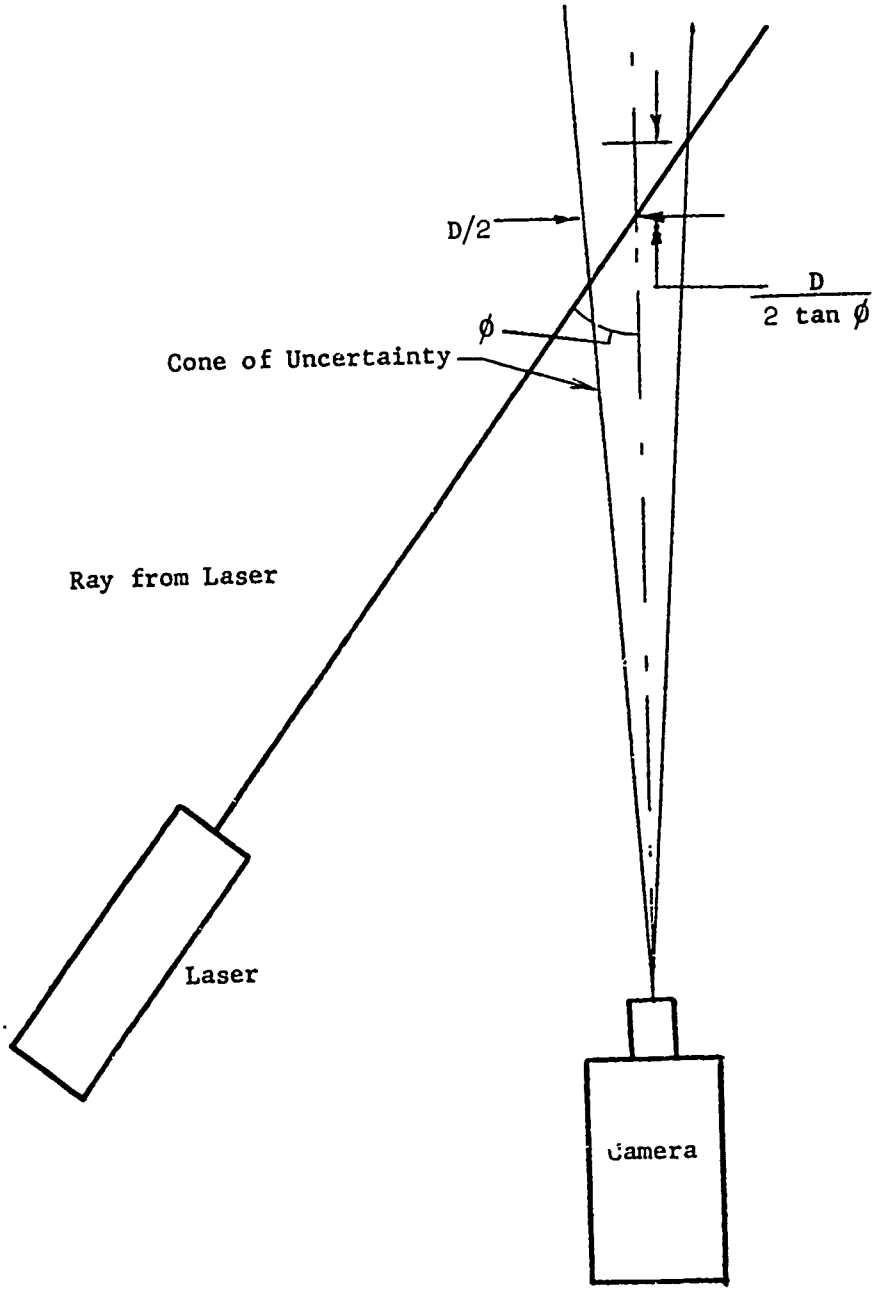


Figure 3,1
Accuracy of Ranging

Uncertainty in locating the laser centerline may add an additional zero to 0.05 inches to this figure. Φ is usually about 30 degrees, giving a limiting range accuracy of 0.035 to 0.07 inches. Calibration errors will add further inaccuracy, but these will not change relative accuracy. Calibration and calibration errors are discussed in section 3.4 and in Appendix A.

A system similar to the above was independently designed by Shirai and Suwa at Electrotechnical Laboratory, Tokyo [Shirai]. Conventional optics were used in a slit projector to project a plane of light.

The advantages of using a laser over using conventional optics are practical, not theoretical. The principal advantage is that a plane of light from a laser is uniformly thin throughout -- hence the depth of field of the source is less limited. (There still remains the problem of depth of field of the camera's optics.) In addition, placing a narrow band-pass optical filter in the camera optics blocks most ambient light. With the filter in place, our system will operate in a sunlit room with no noticeable degradation in performance.

The major disadvantage of using a laser results from its monochromaticity. Objects sensed by a Helium-Neon laser must be either white or a color with a red component. Objects of other colors reflect little or no laser light. Tunable lasers or multi-wavelength lasers may eliminate this problem in the future, but such lasers are now rather expensive, and would require detection of the beam without a filter.

For optimal depth discrimination, the plane of light should be perpendicular to the plane defined by the TV camera, the laser deflection assembly, and the center of the scene being scanned. For best coverage of the scene, the axis of motion of the plane of light, when it is scanned across the scene, should also be perpendicular to the same plane.

In order to obtain more complete and isotropic data, scanning takes place with two different orientations of the plane of light. The plane of light in the second orientation is at right angles to the plane in the first orientation, and both are at 45 degrees with respect to the optimum plane for best depth accuracy. Although the orientation degrades depth accuracy for each scan, the fact that we have two independent measurements increases the accuracy, and the final accuracy is identical to that computed in Equation 3.1, or $0.353 \cdot \cot \Phi$.

In scanning a scene, the TV camera is read, the plane of light is moved by means of a rotating mirror, the TV read again, and the cycle repeats until the entire scene has been covered. The cylindrical lens is then rotated 90 degrees, and the entire scene scanned in this orientation. A laser scan refers to one image from the TV camera, or the data derived from one image. When the data is converted to three-dimensional coordinates, the result is a depth grid. The distance between successive laser scans is wider than the resolution of the TV raster. For each laser scan we have a depth profile along a line. The depth profiles form a sort of depth map, if we imagine viewing the scene from the point of view of the laser deflection apparatus.

An interesting possibility for high speed scanning of a scene is suggested by the work of Will and Pennington [Will]. Their approach to ranging is to project a uniform coded grid onto a scene from a slide projector. This is equivalent to reading many laser scans in a single frame from the TV camera. Will and Pennington made no attempt to measure depth directly, which would have required identifying each line in the image. They were able to extract the normal directions to plane facets illuminated in this manner, but performed no recognition or determination of the boundaries of the facets. However, if one were to use a coded grid in which the code carries positional information, the time to scan a scene would be only the time it takes to read one TV image. Some types of coded grids they suggest are a shift register derived code plate, or the grid known in optics as a linear zone plate. Additional processing would be necessary to identify each line in the image. The techniques to extract the coding will be expensive, but for some applications, a tradeoff of data acquisition time for computation time may be desirable.

3.3 HARDWARE

The basic components of the laser ranging hardware are:

The laser.

Periscope and auxiliary mirror(s) to bring the beam to the deflection assembly.

The deflection assembly, consisting of a spherical focussing lens, a cylindrical diverging lens, and a rotating mirror.

The interference filter.

A television camera capable of being read by the computer.

The hardware is located on the Hand-Eye table at the Stanford Artificial Intelligence Laboratory. An overall view of the setup is shown in Figure 3.2.

The laser is a Spectra-Physics He-Ne laser, model 125, emitting red light at a wavelength of 6328 angstroms. Rated power of this model is 50 milliwatts, but measurements indicate an actual power output of about 35 milliwatts. Calculations based on maximum sensitivity of the vidicon tube and the optical parameters of the system indicate an output of 10 milliwatts to be the minimum for this application. Our 35 milliwatts appears to be adequate, provided

(1) control is maintained of the focussing of the beam, and
(2) an interference filter of sufficient quality is used.

For optimal scanning of a scene, the angle at which the laser beam impinges upon the deflection assembly is important. The beam should be perpendicular to the axis of rotation of the mirror, and as shown in the previous Section (Section 3.2), the axis of the mirror should be perpendicular to the plane determined by the camera lens center, the center of the rotating mirror, and the center of the scene to be scanned. The periscope and auxiliary mirror are for bringing the beam from the laser, (located under the table), to the deflection assembly, in the proper orientation. The periscope consists of two telescoping steel tubes, with a front-surface mirror glued into each end at an angle of 45 degrees to the axis. The telescope arrangement allows adjustment of the height of the beam above the table and of the azimuth of the beam. The auxiliary mirror is mounted on a ball-end-socket clamp which allows an arbitrary orientation of the beam.

Figure 3.3 shows the essential features of the laser deflection assembly. The focussing lens is necessary mainly because of the poor collimation of the beam. At the laser, the beam is a uniform spot about 3 millimeters wide. This diverges to a complex pattern of spots and rings about 5 millimeters wide at the deflection assembly. The focussing lens has a focal length of 500 millimeters, and brings the beam to a spot about 2 millimeters across at the center of the scene, when the cylindrical lens is absent.

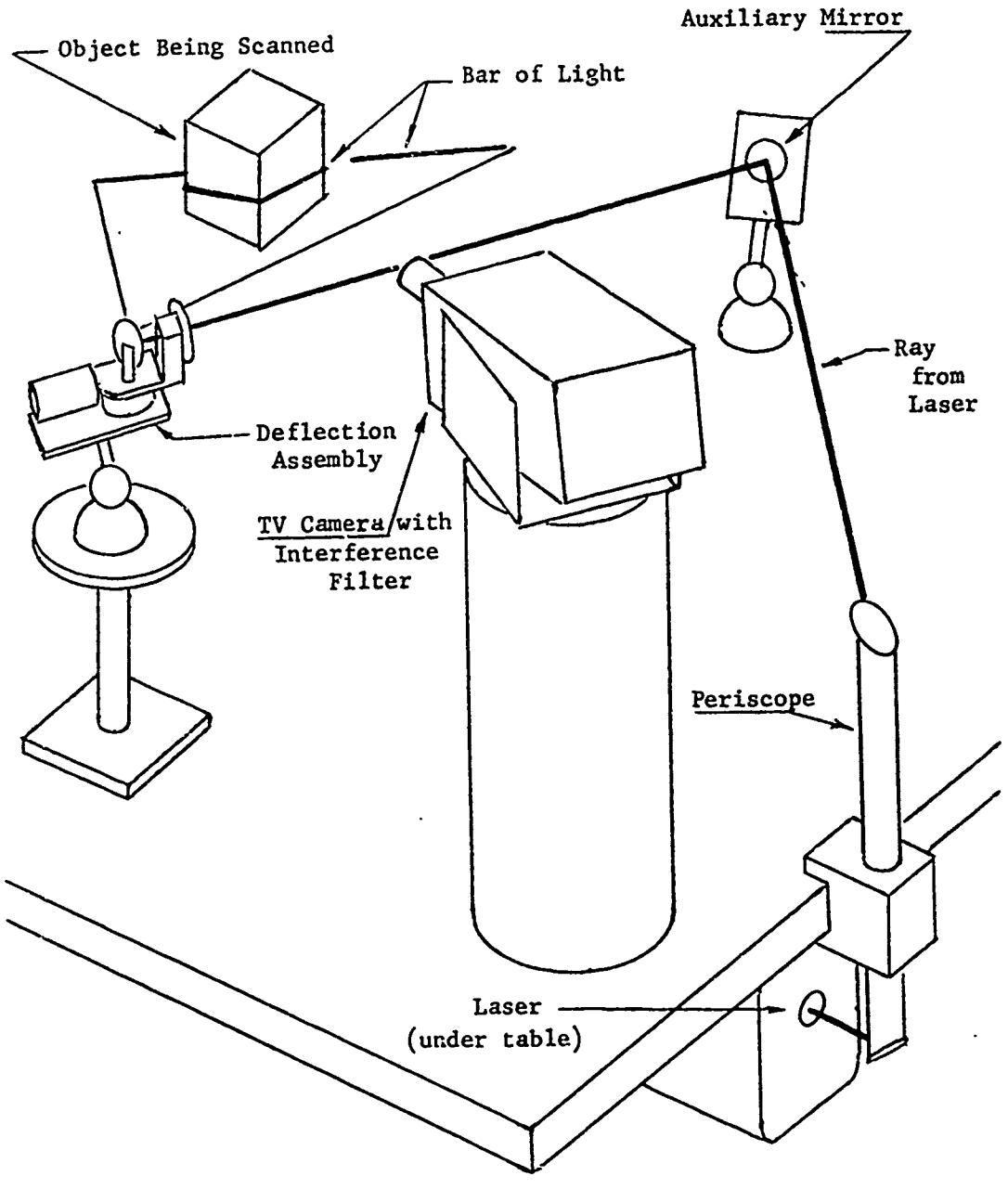


Figure 3.2
Laser Ranging Apparatus

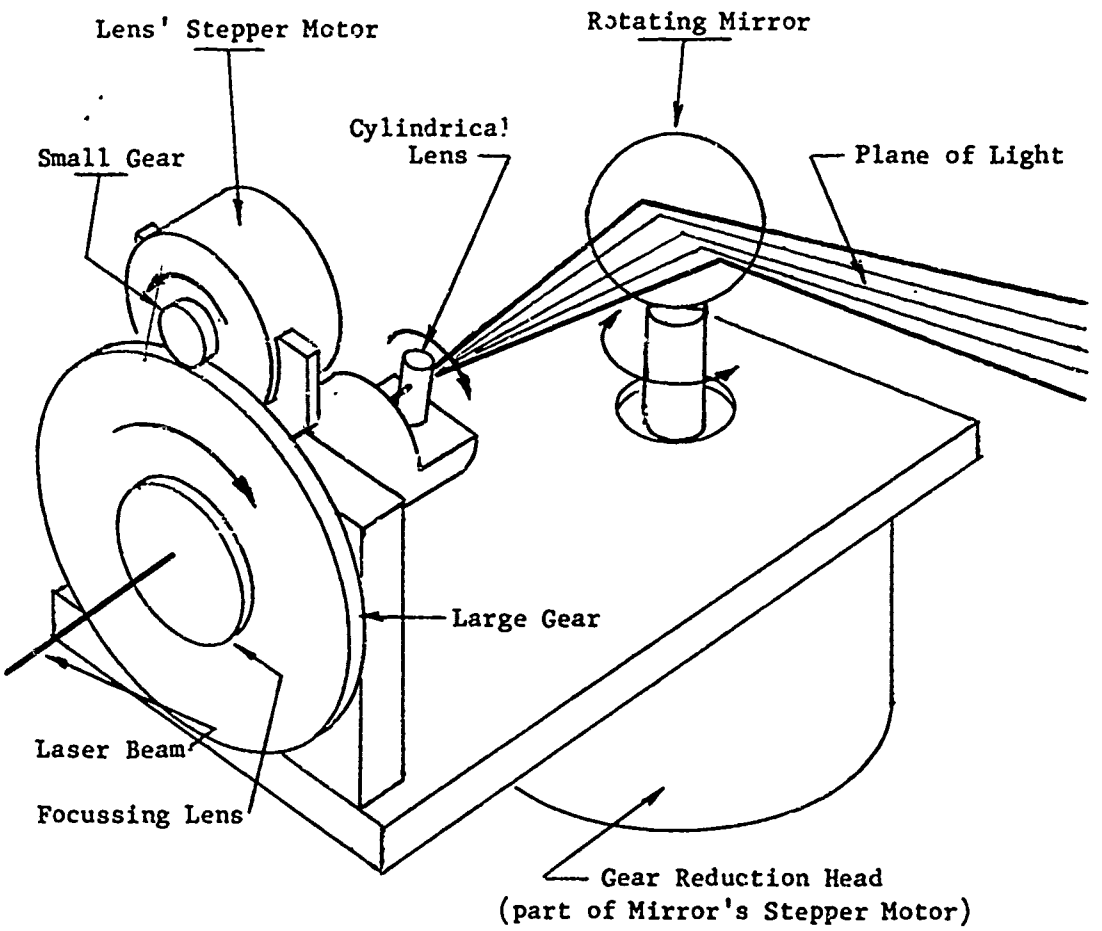


Figure 3,3
Laser Deflection Assembly

The cylindrical lens is a short piece of Pyrex glass rod. Its focal length is approximately 4 millimeters. The cylindrical lens diverges the beam only in the direction perpendicular to the axis of the lens, and changes the circular cross section of the laser beam into an elongated ellipse. A stepper motor may rotate the lens to change the direction of elongation.

A front-surface mirror is mounted on the shaft of a gear reduction head attached to a stepper motor. This arrangement is capable of scanning the beam or plane of light across the scene. The resolution of the motor-plus-gear-reduction is 1728 steps per revolution at the output shaft.

The entire deflection assembly is mounted on a ball-and-socket clamp to allow alignment with the incoming beam and proper orientation of the output illumination.

The function of the interference filter is to screen out ambient light, and let only reflected laser light reach the vidicon tube. Its use is necessary only when working in a darkened room is undesirable. We have experimented with two different filters, both manufactured by Optics Technology, Inc. The first has a bandpass of 8.6 angstroms, and a transmission of about 55 percent at 6328 angstroms. With this filter we have had no difficulty obtaining good TV images in daylight. However the 0.37 inch thickness of this filter made it unsuitable for incorporation into the color wheel of a new television camera presently being instrumented. A thinner filter purchased for the color wheel proved unsatisfactory. Its bandpass is about 20 angstroms, but other calibration data are lacking. Lower transmission at 6328 angstroms and higher transmission at other wavelengths leave too little contrast for daylight operation.

The present configuration is tricky and time-consuming to set up. Usually about one hour is required to set up and calibrate the equipment. A more permanent setup, perhaps with the deflection assembly mounted on top of the periscope, is being contemplated.

Figure 3.4 is the television image of a Barbie doll in place on the table, ready for scanning. The table has been covered with a dark cloth, to suppress the background of the picture (the tabletop). The laser plane of light may be seen illuminating the subject, starting at the right shoulder of the doll, going across the right breast and the stomach, to cross the left leg near the knee.

Figure 3.4
TV Image of a Barbie Doll

3.4 CALIBRATION

Calibration is the process whereby we measure our system parameters in order to be able to reconstruct the three-dimensional coordinates of points in a scene.

Calibration, at Stanford, has come to mean the determination of exact parametric models for cameras and arms (manipulators), so that all coordinates may be measured relative to a fixed coordinate system. The ability to refer to and to access absolute coordinates makes communication between parts of a large system simpler. But for a general purpose intelligent machine, absolute locations will not be as useful as relative ones. There

is little evidence to indicate that human perception makes use of an absolute frame of reference, Rather, we are aware of the locations of objects relative to each other, and of our own position relative to our surroundings. Aharon Gill [Gill] has shown how visual servoing of a computer-controlled manipulator can achieve alignment of parts, with reference to absolute positions only for initial positioning and searching.

For recognition of objects, the important cues come from relative information. Sizes and distances on an object are invariant with rotation and translation. Normal directions to surfaces are computed locally. The directions we compute are, for convenience, relative to some global coordinate system, but what matters is the relative orientations of different surfaces.

In consideration of this philosophy, the emphasis of the calibration of the laser ranging system is on relative accuracy of perception.

Calibration takes place in two phases, camera calibration and laser calibration.

Camera calibration involves establishing a correspondence between points on the tabletop and points in the TV image. The plane of light is projected on the table top, and six laser scans are read by the camera, in the pattern shown in Figure 3.5. The nine intersections of the laser lines are measured on the table top, relative to a local coordinate system (usually with its center near the center of the calibration pattern. Straight lines are fit to each laser scan in the TV image and the nine intersections of the straight lines are calculated. The PRAXIS minimization program developed by Richard Brent and Irwin Sobel determines the values of the camera parameters which give the best correspondence between the nine measured intersection points and the nine intersections of the laser scans in TV coordinates.

Five camera parameters are determined by the calibration program: the camera's orientation with respect to our local coordinate system (pan, tilt and swing angles), and the horizontal and vertical scale factors of the camera. Three additional parameters, denoting the position of the camera lens center, are measured approximately (to the nearest inch), and are held fixed during the minimization process. From these eight parameters, it is possible to determine a matrix, CAMTRANS, relating points in the scene to their coordinates as observed in the TV image. The relation is:

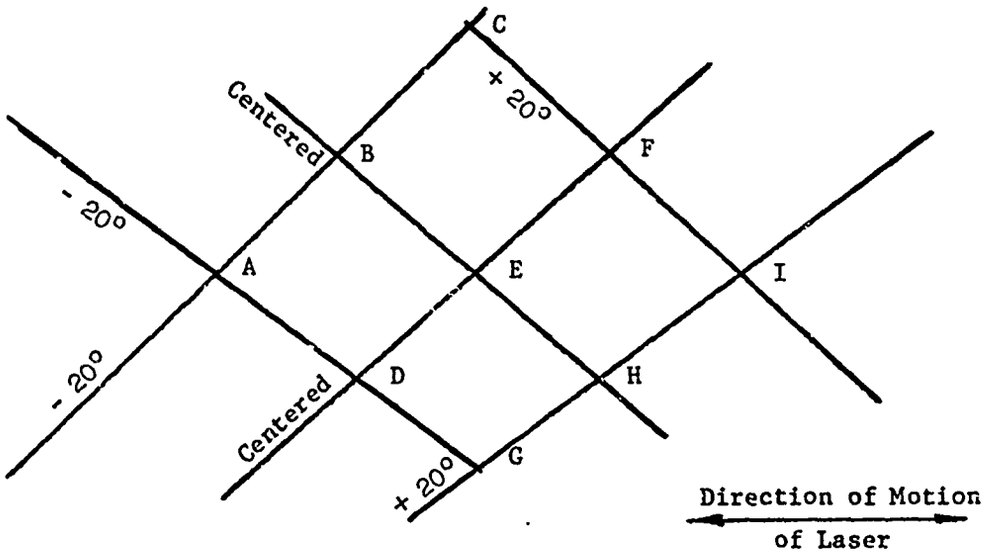


Figure 3.5
Calibration Pattern

$$\begin{matrix} U \\ V \\ H \end{matrix} = [\text{CAMTRANS}] * \begin{matrix} X \\ Y \\ Z \\ 1 \end{matrix} \qquad \text{[Equation 3.2]}$$

where X, Y, and Z are the coordinates of the object point, and the image point coordinates are given by U/H and V/H.

Errors due to inaccuracy in measurement of the camera's position with respect to our local coordinate system may produce absolute errors in location of points located far from the center of the scene, but the relative errors will be limited to a slight warping or scaling of the scene.

Laser calibration is based on minimization of matching errors. Where two laser traces, arising from two different rotations of the cylindrical lens, cross in a scene, different determinations of depth may be obtained for the point where they cross. We call this difference the matching error. Laser calibration adjusts the parameters of the laser deflection system to minimize the sum of the squares of the matching errors over a scene.

If the position of the laser deflection assembly is known, the remainder of the laser deflection parameters may be computed from the calibration pattern and known physical parameters, as we shall show below. Thus, the only parameters which must be adjusted by the minimizer are the position of the laser deflection assembly with respect to our local coordinate system. A rough initial measurement provides a starting point for the minimizer.

The "position of the laser deflection assembly", (or, for brevity, the "laser center.") we take to mean some point on the surface of the rotating mirror. The laser light actually emanates from a virtual point behind the mirror, but there is a small area on the surface of the mirror through which all planes of light pass. For the purposes of calibration, we assume the light to emanate from the center of the area. (We discuss the validity of this assumption in Appendix A.)

Since three points determine a plane, the homogeneous coefficients of the plane of light in the vertical position may be determined from point B and point H in Figure 3.5 and the laser center. The plane in the horizontal position is determined from points D and F and the laser center. Points A and I and the laser center determine a plane which we assume to be perpendicular to the axis of rotation of the rotating mirror. (The validity of so assuming we discuss in Appendix A.)

To obtain the coefficients of the plane of light when the mirror is at some position other than zero rotation requires a series of transformations. These are a translation, T, to transform the laser center to the origin of coordinates, a rotation R to align the mirror axis of rotation with the vertical, a rotation M about the vertical axis corresponding to twice the angle of rotation of the mirror, followed by rotation by R inverse and a translation by T inverse to restore the original orientation of the axis and position of the laser center. When the plane of light is expressed in homogeneous coordinates, the transformations may be represented by matrix multiplications. (See [Roberts 63] for an exposition of homogeneous coordinates and transformations.) To obtain the coefficients of the plane of light at a mirror rotation of θ use the following matrix equation:

$$\text{PLANE}(\theta) = \text{PLANE}(0) * \text{SHIFT} * \text{ROTATE}(\theta) * \text{SHIFT}^{-1}$$

[Equation 3.3]

where SHIFT is the product of the translation T and the rotation R, and ROTATE(θ) is a rotation about the vertical axis.

Given the homogeneous representation for the laser plane in a specific orientation, a system of equations may be solved to yield a collineation COLL such that the physical coordinates of any point on the image of the scan line is given by

$$\begin{matrix} X \\ Y \\ Z \\ H \end{matrix} = \text{COLL} * \begin{matrix} U \\ V \\ 1 \end{matrix} \qquad \text{[Equation 3.4]}$$

where U and V are the raster coordinates of the point, and X/H, Y/H, and Z/H are its physical coordinates.

The complete calibration data for a scene contains the CAMTRANS matrix, the eight camera parameters (the location of the lens center, the pan, tilt, and swing angles, and the horizontal and vertical scale of the camera), PLANE(Ø) (in homogeneous coordinates) for both horizontal and vertical orientations, the matrix SHIFT and its inverse, the location of the laser center, and the orientation of the mirror rotation axis.

Appendix A shows that errors in the calibration, principally those due to non-perpendicularity of the axis of rotation of the mirror to the plane determined as mentioned above, contribute at least as much to the uncertainty of range determination as the resolution of the camera. The average absolute error in range, due to all known sources is about Ø.Ø5 inch.

After minimization, the matching errors typically run about Ø.Ø1 inch. Since the resolution of the TV camera and the width of the laser lines lead us to expect an absolute error in the neighborhood of Ø.Ø35 to Ø.Ø7Ø inch.

4 PRELIMINARY PROCESSING AND CURVE FITTING

This chapter describes the low-level processing necessary to convert the TV images obtained with laser illumination, to useable depth information for the description routines.

The unprocessed data exists as images on the vidicon of the TV camera, the output of the preliminary processing routines will be a set of curves, which may be combined with the laser calibration data, to reconstruct the original scene in three dimensions.

4.1 LINE EXTRACTION

The TV camera is read by a standard routine from the Stanford Hand-Eye Library [Pingle]. Output from this program is an array of four bit brightness samples, packed nine to a 36 bit PDP-10 word.

We scan this array for non-zero brightness bytes, and store the brightness and coordinates of the nonzero samples in an array. One sample occupies one word of data. Although the storage requirements are higher for each sample, because most of the picture is dark, there is a net reduction in storage required of about 20 to one as a result of this operation. Detection typically requires about 5.9 microseconds per word which contains all zero samples, and about 271 microseconds per nonzero word. For a typical picture of 2300 words, this processing requires about 48 milliseconds. This averages to 20 microseconds per word or 2.2 microseconds per point.

Figure 4.1 displays the nonzero samples detected in one frame of television input for the Barbie doll of Figure 3.4. (The plane of light for this image passes from the left shoulder, across the left breast and the stomach, and along the right thigh.) The line appears wider across the body than along the leg. This is principally because the body of the doll faces the laser deflection assembly; light from the deflection assembly strikes the surface perpendicularly. The leg is oblique to the laser illumination, and the total illumination per unit surface area of the leg is smaller.

The lines thus detected are usually several raster units broad. The breadth obtained is principally due to a low brightness threshold made necessary by intensity variations along the line. The line is generally brighter near its middle than at

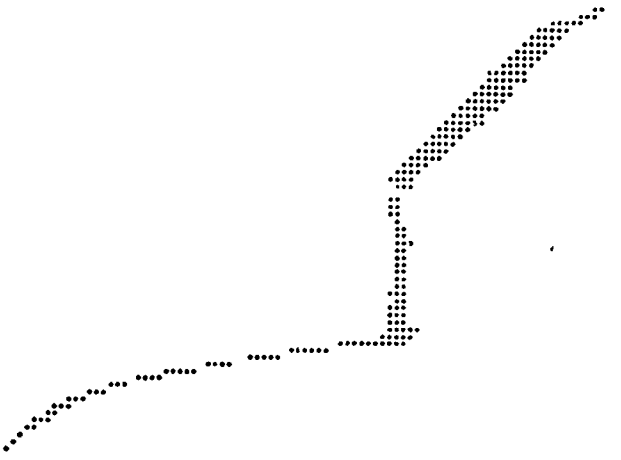


Figure 4.1
Non-zero Brightness Samples From One Laser Scan

either end, and hue and intensity on the object being scanned account for a great deal of variation. In order to detect the dimmest line in the scene a low clipping level or threshold is imposed on the hardware. This results in saturation in the brighter areas of the line. Although the edges of the line are dim with respect to the center, the edges are detected where the line is bright. Refocussing of either the laser beam or the camera have not been found to have a significant effect on the width of the line.

The line is thinned by locating its centerline. Horizontal and vertical "slices" are made through the picture. For every horizontally or vertically contiguous set of points, the coordinates of the center of that set are placed in the array for the thinned line. The center is computed by weighting the position of each sample by its brightness, and taking their average. Figure 4.2 shows the line of Figure 4.1 after thinning. In this figure we see several points that are not near the centerline of the trace, principally where the trace is vertical. These points arise from vertical sections passing along the edge of the unthinned trace. The operation which links the points will eliminate these spurious points.

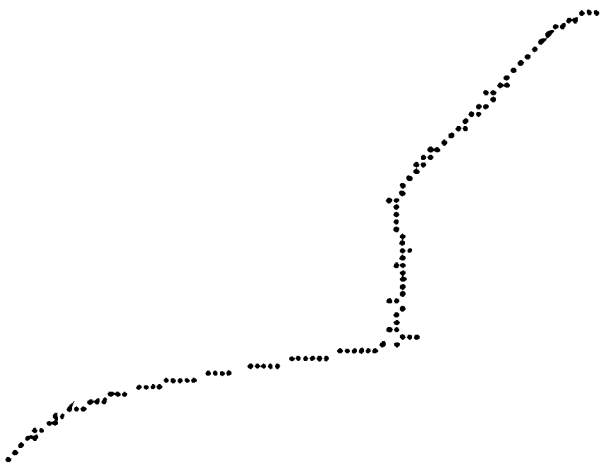


Figure 4.2
Laser Trace After Thinning

A better thinning operation would take centerlines only perpendicular to the approximate direction of the line. However, no single direction of search can apply to an entire laser scan, even if the orientation of the laser plane is known; lines may lie at any orientation. The only way to apply this would be to first apply a local analysis to the line at several places to determine the direction of the line.

A single program controls the rotating mirror and television camera, and performs line detection and thinning. The usual mode of operation is to use this program to manually adjust the camera parameters for optimum contrast, then to specify to the program the initial beam position, step increment, and number of steps to scan a scene. The program will automatically step the mirror, read the TV camera, detect the nonzero brightness samples, and store their coordinates on the disk. A second pass of the same program reads the data stored on the disk for the thinning operation.

On a disk file are stored, for each laser scan, the angular positions of the mirror and the cylindrical lens, the number of points detected in the image, the coordinates of each point, and if thinning has not yet been done, the brightness of each point.

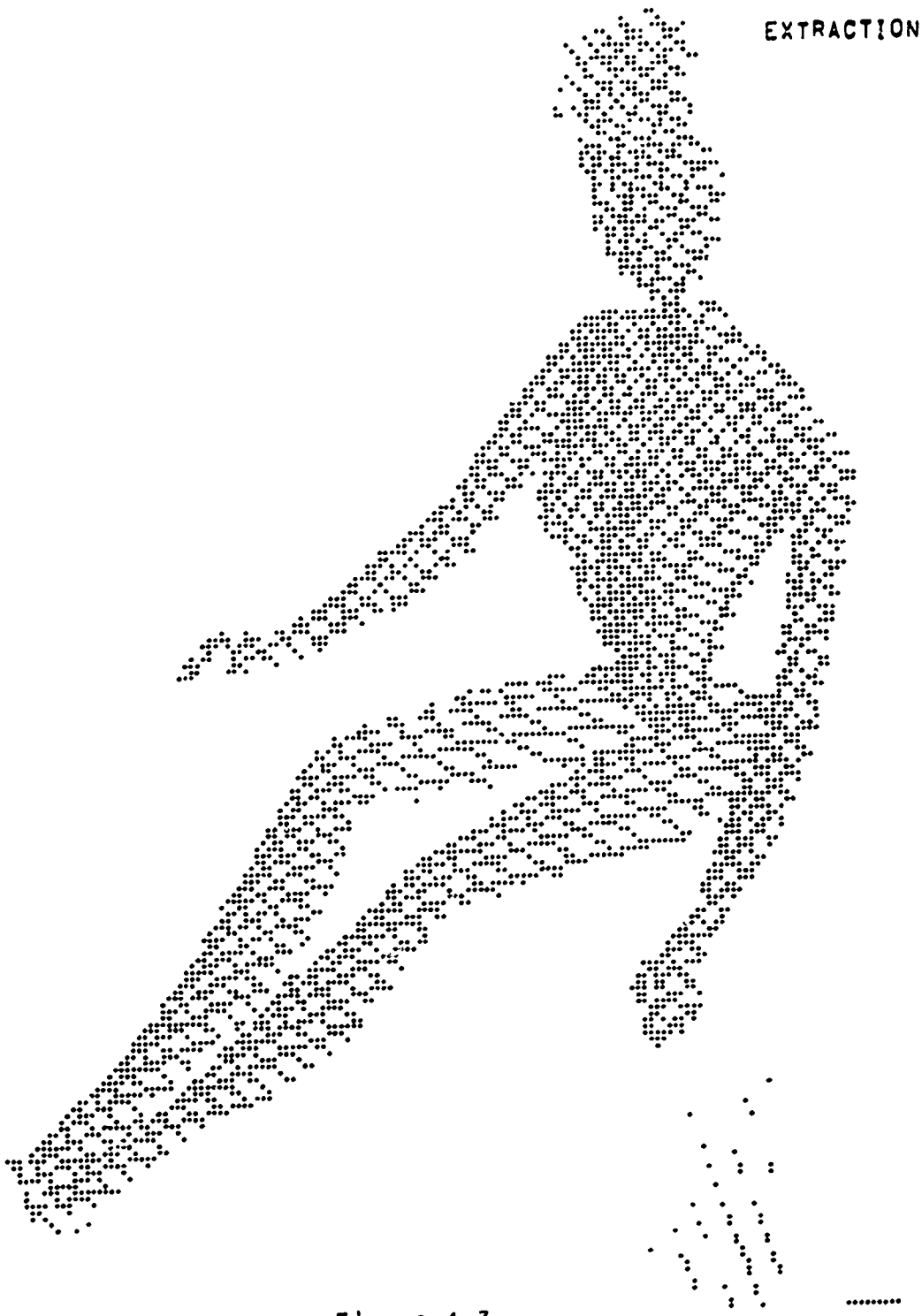


Figure 4,3
Multiple Laser Traces of a Barbie Doll After Thinning

Figure 4.3 is a composite of all the laser traces of the Barbie doll, after thinning. Near the lower right corner we see several traces where the black cloth on the tabletop has failed to completely suppress the background of the figure. The short horizontal line in the lower right corner is due to errors in the TV input hardware.

The image of Figure 4.3 contains 100 laser traces, 9777 non-zero brightness samples were detected, and 4956 remained after thinning. The program required 8 minutes of elapsed time to scan the scene, and 1 minute, 18 seconds of CPU time. The 8 minutes includes some time for user interaction with the program. A one-second settling time is imposed, after each step of the rotating mirror, for persistence on the vidicon to die, but this is partly overlapped with processing time. The balance of the time not accounted for by computation, waiting, or interaction is due to time sharing on the computer. The thinning operation required 39 seconds of CPU time additional. The program which accomplishes this runs in 27K of core on the PDP-10 computer.

4.2 LINKING THE POINTS

The centerline points resulting from the line thinning operation (Section 4.1) are stored in the data structure in raster order, that is, they are sorted by their coordinates. In order to separate line segments, perform curve fitting, eliminate spurious points, and provide a more useable structure to the data, contiguous sequences of points must be identified as lines, and the points sorted in order along these lines.

The points are linked by a "maximal minimum distance" method. This is akin to finding a minimal spanning tree linking contiguous or nearly contiguous points, and finding the longest path in this tree. Because of the linear nature of our data points, we may take some heuristic short cuts.

To do this, we first shall define a "minimum distance" between two points in a contiguous set corresponding to the shortest path through the set which link the two points. The "end points" of a contiguous set are the pair of points which have the greatest minimum distance between them. The linking routine orders the points of the set according to the shortest path between the end points, and deletes those points of the set which do not lie on this path.

We define a minimum distance function as follows: Between any two neighboring points, the "point-to-point distance" is defined in Table 4,1 (Two points are neighbors if and only if there is an entry in the table corresponding to their relative location.) For any two points, not necessarily neighbors, a "path distance" may be found by summing the point-to-point distances over some neighbor-to-neighbor path connecting the two points. In Figure 4.4, A, B, and C are paths connecting points p and q, corresponding to path distances of 7, 6, and 7. The "minimum distance" between two points is the minimum, over all paths, of the path distance between them. In Figure 4,4 the minimum distance between p and q is 6, corresponding to path B.

6	5	4	5	6
5	3	2	3	5
4	2	*	2	4
5	3	2	3	5
6	5	4	5	6

Table 4,1
Point-to-Point Distance (From Center Point)

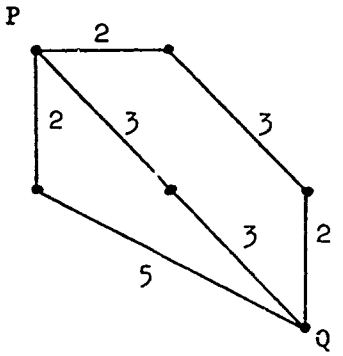


Figure 4,4
Some Path Distances

To find the end points we take the first point on the list as a starting point, and compute the minimum distance to every other

point to which a path exists. (By considering the nearest points first, the time to compute these distances is proportional only to the number of points.) Assuming there are no long circular paths, (and the nature of our data precludes this,) the point with the greatest minimum distance from the starting point must be one of the end points. (See Figure 4.5.) Taking this end point as a new starting point, we again compute the minimum distance to every other point, to identify the other end point. (Figure 4.6.)

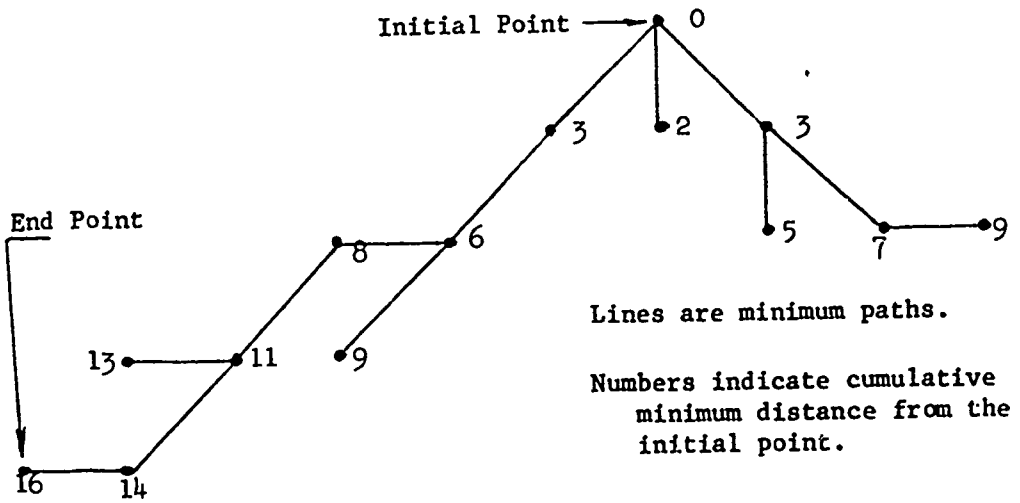


Figure 4.5
Locating One End Point

After one pair of end points has been found, the points on the minimum path are copied into a separate array in their order along the path. All points to which a path exists but which do not lie on the minimum path are to be considered as noise, and the entire contiguous set of data points is deleted from the list. The process is repeated to find other line segments until no points remain unaccounted for. Segments containing fewer than five points are discarded as spurious. The linked points in each line segment are then passed on to the segmentation and curve fitting routines.

The point linking routine found two line segments in the data of Figure 4.2. These two lines are shown in Figure 4.7. A gap

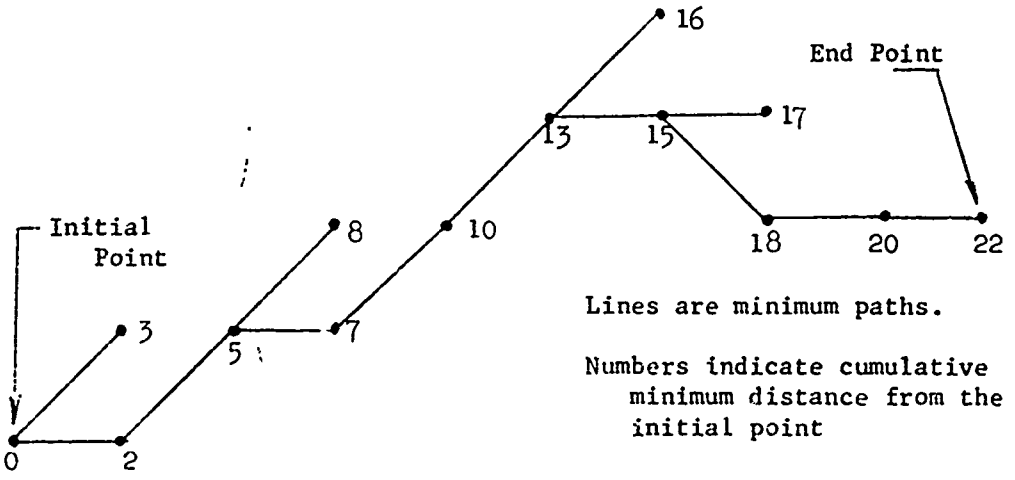


Figure 4.6
Locating the Other End Point

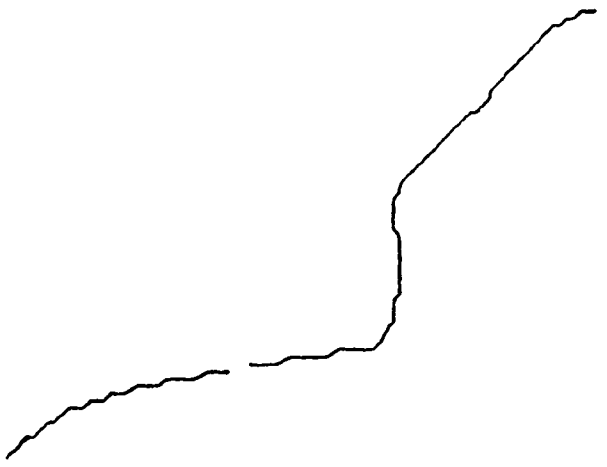


Figure 4.7
Two Lines Found by the Linking Algorithm

In the points along the thign of the doll prevented identification of the trace as a single line. Figure 4.8 shows the traces of Figure 4.3 after the linking operation.

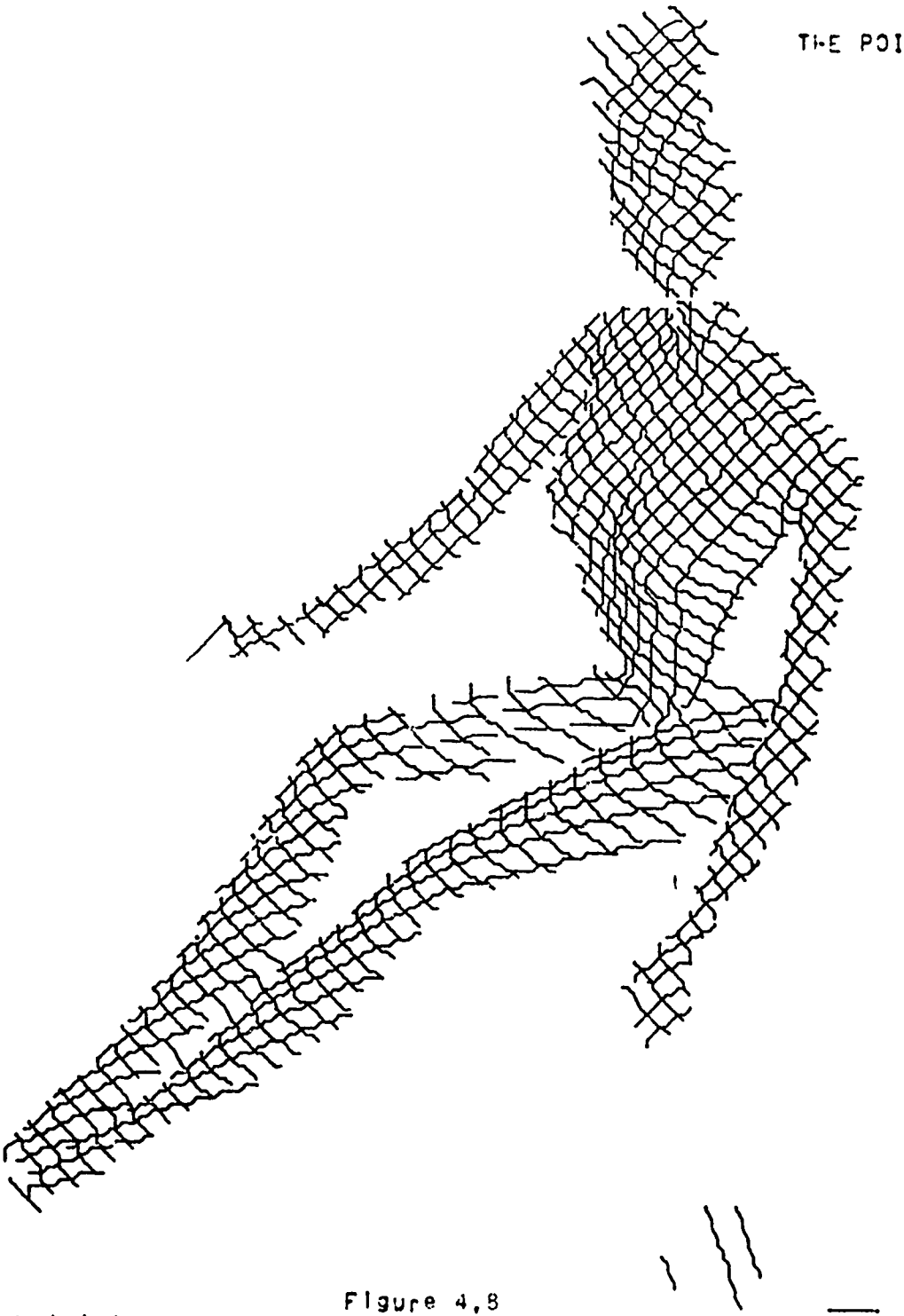


Figure 4.8
Multiple Laser Traces of a Barbie Doll After Linking

4.3 RECURSIVE SEGMENTATION

The next step in the description of objects is curve fitting of the laser data. For each line segment found by the sorting operation (Section 4.2) a curve or set of curves is found which may represent or approximate the points along the line segment, dividing the segment into subsegments, as necessary, to obtain an adequate approximation.

There are several advantages to curve fitting the data. One obvious advantage is a compression of the gross amount of data which must be handled, a compression usually of about 4 to 1. Curve fitting smooths the data and rejects spurious points. The segmentation necessary for curve fitting is useful in segmentation of objects. And the more richly structured data base of curves facilitates further processing. However it is time consuming, and it introduces a certain amount of systematic error.

Segmentation, at the present status of our experimental research, is used primarily to assure a good fit to the data of the fitted curve segments. We have found that the methods we use are accurate in locating corners in lines, or other abrupt changes in characteristics (see Figure 7.1). Corner data are not presently used by our programs, but will be useful when this research is carried further.

Curve fitting and segmentation is done in TV raster coordinates, before any use is made of the calibration information to obtain depth. Whether this takes place before or after conversion to three dimensions is only a minor consideration. The transformed data are planar for each laser scan, and fitting could also take place in a coordinate system aligned on the plane of light which illuminates the scene. The principal reasons we have chosen to segment and fit in two dimensions are that the data structure is simpler in two dimensions, and that transformation of the parameters of the curves to three dimensions requires less processing than transforming the points they approximate.

The next Section (Section 4.4) and Appendix B describe how a given line segment may be represented by either a straight line or by a general second order curve. The curve fitting routines find a curve which best fits the line segment, in a least squares sense.

The recursive segmenter starts with a line segment found by the sorting routine (Section 4.2). This line segment is passed to the curve fitter for fitting by a straight line or second-order curve. If the fit is acceptable (see Section 4.4 for criteria of acceptability), then the routine exits. If not, then the curve must be broken into two pieces, and each subsegment passed to the recursive segmenter for fitting and, if necessary, further segmentation. After the two subsegments have been fit, if either of the subsegments required further segmentation, then an attempt is made to merge the two sub-subsegments on either side of the original break. Figure 4.9 illustrates this. If merging fails or is unnecessary, it is determined whether a shift in either direction of the point of segmentation will result in a better overall fit, as shown in Figure 4.10

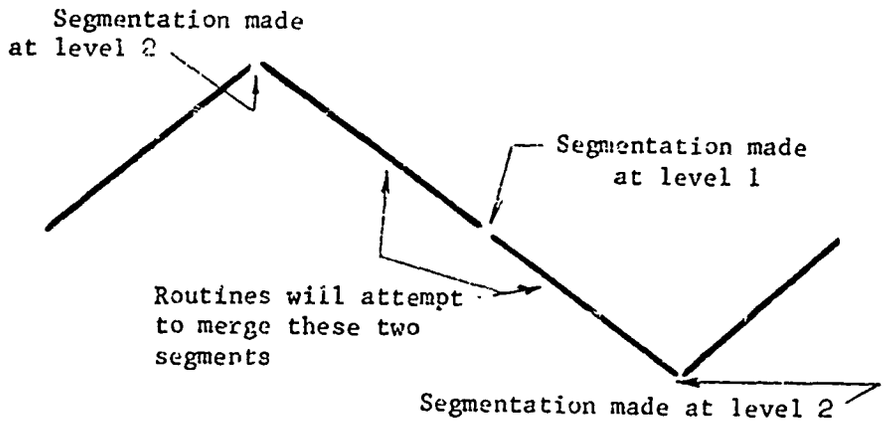


Figure 4.9
Merging of Curve Segments

Two tests determine the point at which a line segment is broken: inflection and maximum excursion. A "baseline" is drawn between the end points of the line segment. If the segment is S-shaped, (i.e., contains an inflection point or change in sign of curvature,) then the segment is broken at the point where it crosses the baseline. Otherwise it is broken at the point where it attains its maximum excursion from the baseline. Figure 4.11 and Figure 4.12 illustrate these two methods of segmentation.

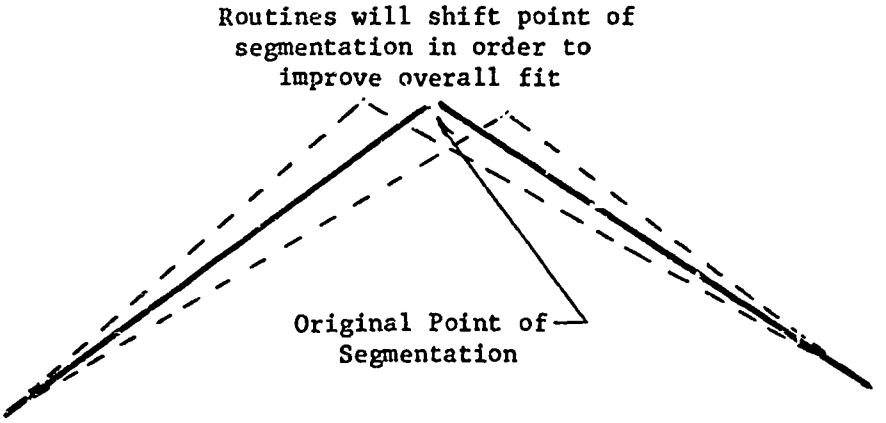


Figure 4.10
Shift of Segmentation Point

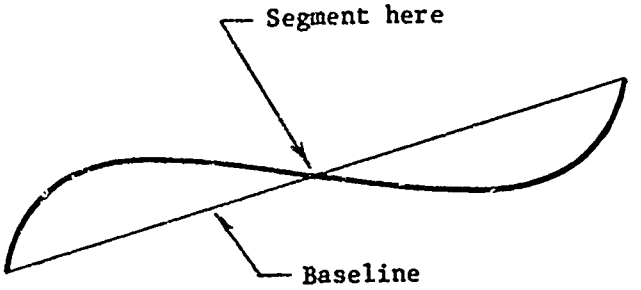


Figure 4.11
Segmentation at Inflection Point

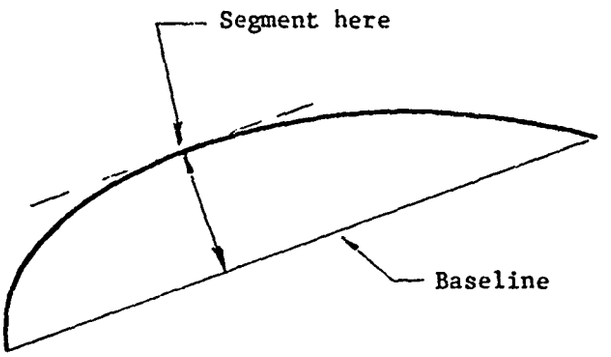


Figure 4,12
Segmentation at a Maximum Excursion

4,4 CURVE FITTING SUBROUTINES

In every case, an attempt is made to fit each line segment by a straight line before trying any higher-order curves. A straight line is represented by the equation

$$A\,x + B\,y + C = 0, \qquad \text{[Equation 4.1]}$$

For a given line segment, the constants A, B, and C are determined by the method of Section B,3 of Appendix B. The data structure for a straight line contains the values of the constants A, B, and C, and the coordinates of the two end points of the line segment.

The fit of a straight line is acceptable if the RMS error of the fit is below a certain threshold, if the individual error ($A\,x + B\,y + C$) for each point is below another threshold, and if there exists no systematic deviation from linearity of the points. A systematic deviation from linearity is detected whenever five consecutive points lie on the same side of the "best fit" line. (In the case where the number of points in a

line segment, N, is less than ten, the number of consecutive points necessary is N/2.)

A general second order curve may be represented either in the form

$$A x^2 + B x y + C y^2 + D x + E y + F = 0 \qquad \text{[Equation 4.2]}$$

or in the form

$$\frac{x'^2}{M} + \frac{y'^2}{m} = 1 \qquad \text{[Equation 4.3]}$$

where $x' = (x - X_c) \cos\theta + (y - Y_c) \sin\theta$

$y' = (y - Y_c) \cos\theta - (x - X_c) \sin\theta$

Equation 4.2 is useful for determining the goodness of fit of the curve to a data point or the set of data points. Equation 4.3 contains useful geometric information about the curve. If m is positive, Equation 4.3 represents an ellipse. Then M and m are the squares of the major and minor axis lengths, respectively, X_c and Y_c represent the coordinates of the center of the ellipse, and θ represents the rotation of the major axis clockwise from the horizontal. If m is negative, Equation 4.3 represents a hyperbola, and M, m, X_c, Y_c, and θ are interpreted similarly.

The method of Section B.1 finds five different curves in the form of Equation 4.2 which may or may not adequately represent the set of data points. These five curves are ranked in order of goodness of overall RMS fit, and considered one at a time. Curves are rejected as unacceptable if the overall error of the fit is too large or if any individual data point lies too far from the curve. In addition, checks are performed to prevent such misfits as those shown in Figure 4.13.

The data structure for a general second order curve contains the six constants of Equation 4.2, the five constants of Equation

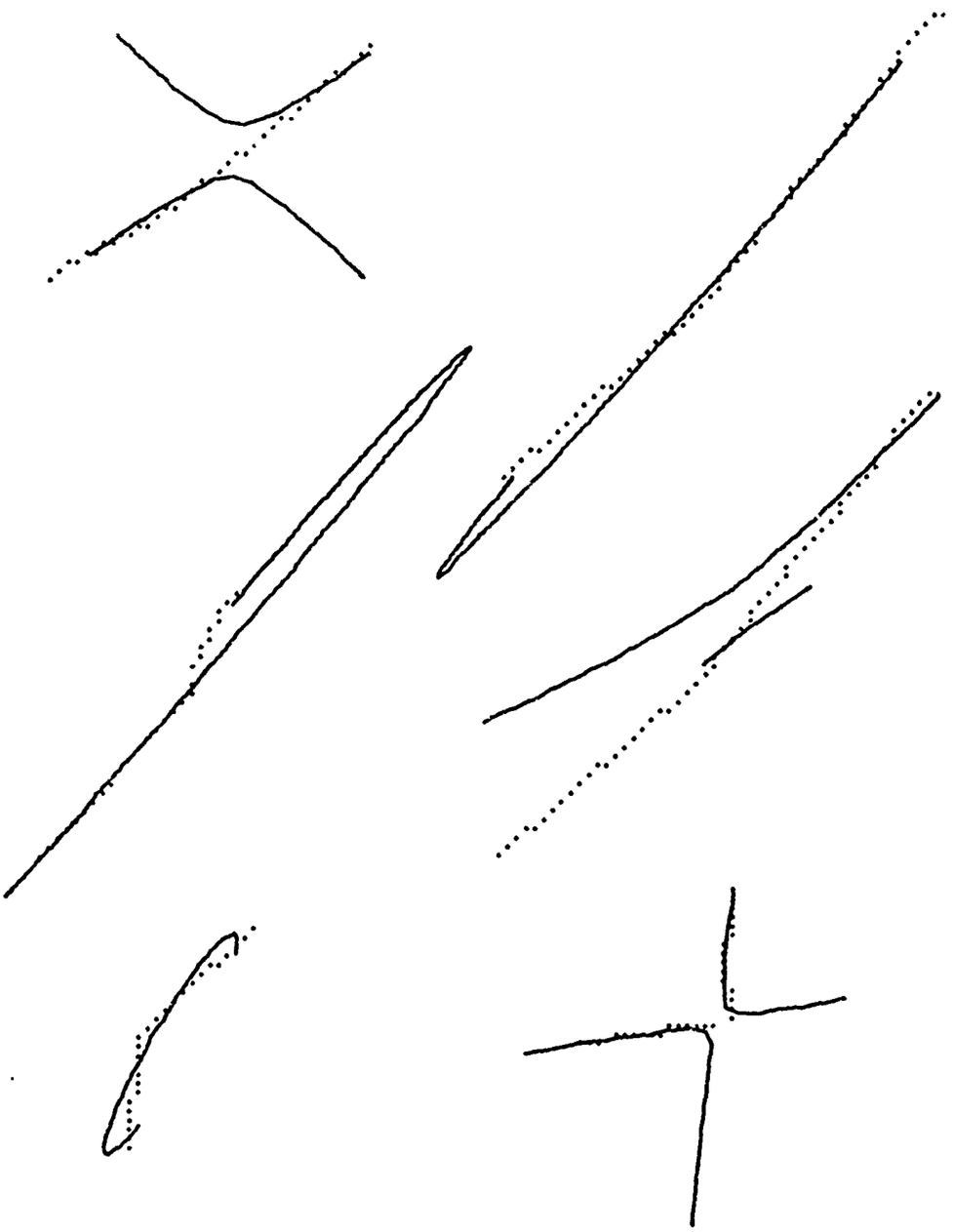


Figure 4.13
Some Examples of Unacceptable Curve Fits

4.3, the coordinates of the end points of the subsegment, and the polar coordinates, relative to the principal axes, of the end points, in clockwise order.

General second order curves have proven a not entirely satisfactory representation for curved lines. They are clumsy to manipulate, and the least squares procedure for evaluating their parameters tends to give solutions which exhibit the wrong curvature. The curvature problem is discussed in detail in Section B.5 of Appendix B. Some alternatives to general second order curves for representing curved lines are presented in Section B.6 in Appendix B.

In the case of the two segments shown in Figure 4.7, the following steps were necessary to fit them:

1. The segment on the left was fitted with a hyperbola.

2. The curve on the right could not be fitted with a straight line or a second-order curve, and was segmented near the middle of the vertical portion.

3. These two subsegments were fitted with a straight line and a hyperbola, without further segmentation. Figure 4.14 shows the state of affairs at this point.

4. The point of division between the two subsegments was shifted, to obtain a better fit. Figure 4.15 shows the final fit.

Figure 4.16 is a composite of all the laser traces of the Barbie Doll, after segmentation and curve fitting. A few of the shorter line segments, notably around the wrists and ankles, have been lost, because the point sorting routine did not find enough contiguous points to consider significant. At present, at least five contiguous points are necessary for a line segment to be retained. Many of the shorter line segments are represented as straight lines. At least six points are necessary for nontrivial solutions to the eigenvalue equations to exist. Some of the traces of the tabletop remain in the lower right corner.

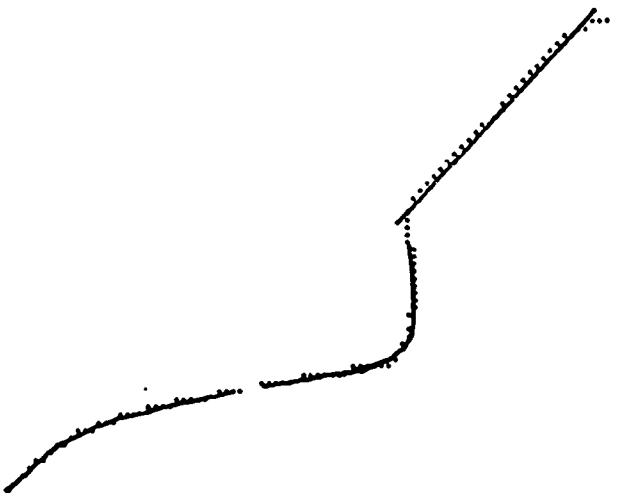


Figure 4.14
Curve Fitting Without Optimization of Division Point

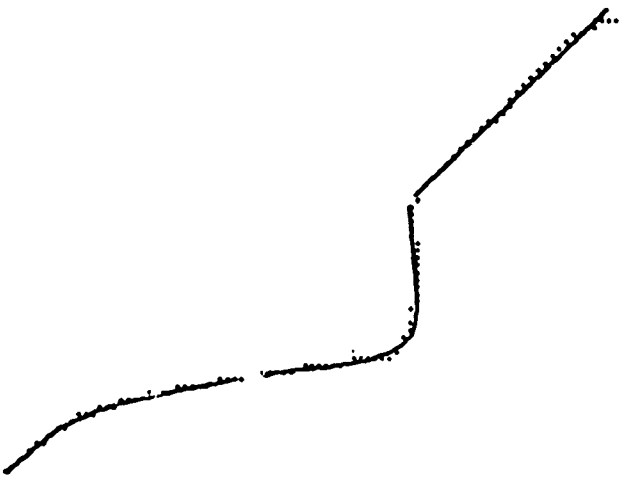


Figure 4.15
Final Fit of a Laser Trace

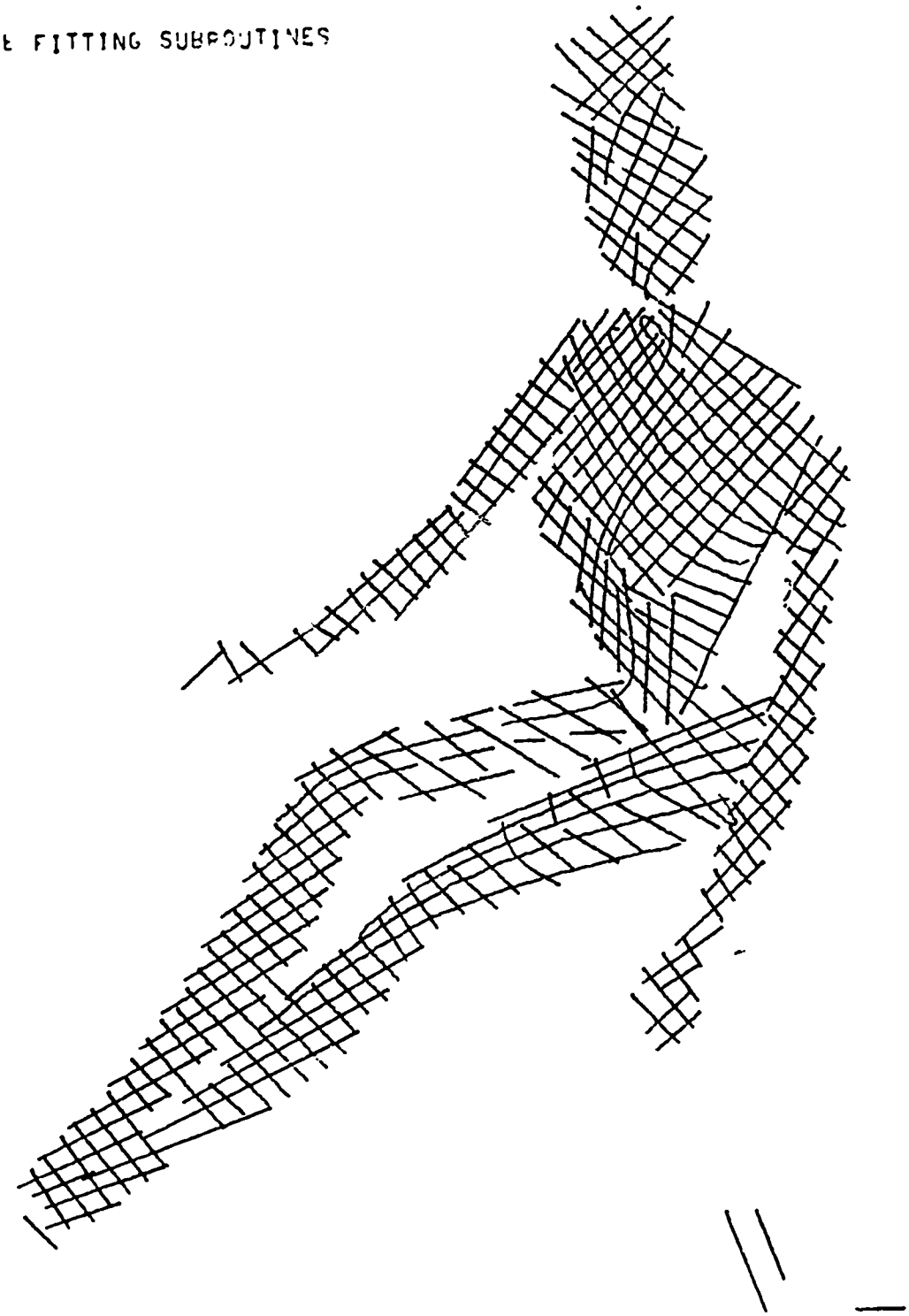


Figure 4.16
Multiple Traces of a Barbie Doll, After Curve Fitting

5 FITTING OF PRIMITIVES

An essential prerequisite of the description of objects is the ability to recognize and trace out the primitives of which the descriptions are composed. The identification of the ends of the primitives, the places where gross changes in cross section occur, aids in segmenting complex objects into their parts. This chapter describes our methods for identifying and tracing generalized cylinders.

Ideally, these routines would be used by a higher level program which makes hypotheses about the structure of an object, and calls the cylinder tracer to verify these assumptions. The structure and operation of such a higher level program is suggested in Chapter 6. In practice, an initial segmentation is carried out for the entire object, and the user of the program may specify which pieces will be traced.

The basic mode of operation of the cylinder tracer is to accept an initial estimate as to the location and orientation of the axis of a segment, and to either improve on the estimate, or reject the estimate as not leading to a plausible generalized cylinder. The initial guess may come from a two-dimensional analysis of the laser scans, as described in Section 5.3, or may be supplied by a higher level hypothesis generation program for verification.

An essential function of the cylinder tracer is segmentation of complex objects. Once an initial estimate is verified as belonging to part of an easily described cylinder, the routine tries to extend the cylinder in both directions. The extension is terminated where a gross change of cross section diameter occurs, suggesting the existence of a joint between segments.

Presently the routines assume all generalized cylinders may be approximated with circular cross sections. This is a limitation in the implementation, but not in the general method. It should be possible to characterize segments of objects using the circular assumption, and later to perform a more specific cross section determination. Alternatively, the routines which fit circles during the cylinder tracing could be modified and expanded to account for other cross sections. Section 6.2 discusses methods for and problems in description of arbitrary cross sections.

The method is an iterative one. Starting with some estimate of the axis of a segment, cross sections are determined

perperdicular to this axis. The centers of the cross sections will (hopefully) represent an improvement on the initial axis assumption. The method is usually convergent, but if divergent behavior is detected, steps may sometimes be taken to correct the divergence.

5.1 CYLINDER TRACING

The method we use for cylinder tracing requires an initial estimate of the axis of the generalized cylinder. It obtains an improved axis estimate by fitting circles in cross section planes perpendicular to the initial axis, and using the centers of the circles to define a new axis.

A generalized cylinder is represented in the program as a sequence of points in three dimensions, denoting the axis of the cylinder, and a linear radius function of the form,

$$RADIUS(n) = RADIUS(0) + M * n, \quad \text{[Equation 5.1]}$$

where n corresponds to the order of points on the axis.

To illustrate this Section and the next, we make use of the laser image of a cone. Figure 5.1 is a composite of the curve-fitted line segments obtained from the laser traces. Calibration data is used to convert each line segment into a space curve in three dimensions. We may compute how these segments will appear if viewed from some point in space other than the TV lens center, obtaining the side view of the cone shown in Figure 5.2. The remainder of the figures of the cone, in this Section and the next, are also side views.

Figure 5.3 shows an initial estimate of the axis of the cone, supplied by the preliminary segmenter (Section 5.2). The approximate outline of the cone is sketched on the figure for clarity.

To improve on the initial estimate, the following steps are executed:

1. For each point on the axis, a cross section plane is determined normal to the axis direction, as explained below.

2. For each plane, points on the surface of the object in the vicinity of the plane are found. Section 5.4 describes how these points are found.

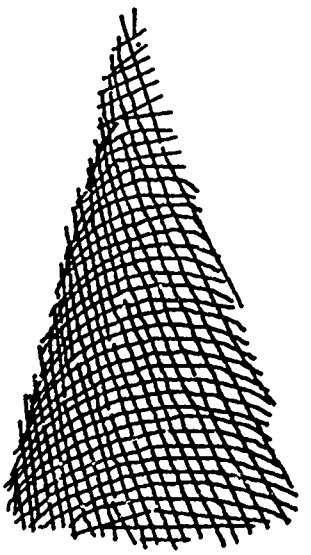


Figure 5.
Cone: Front View

Figure 5.2
Cone: Side View

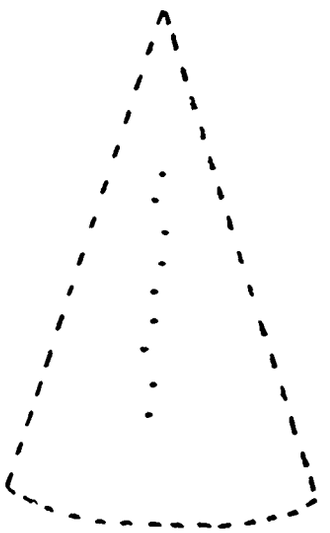


Figure 5.3
Initial Axis Estimate

3. From the points found in step 2, a preliminary estimate of the radius and center of the (circular) cross section are obtained, as described in Section 5.5. Figure 5.4 shows the circles obtained, shown in perspective on their respective planes.

4. The radius function, Equation 5.1, is determined from a least squares fit on the radii found in step 3.

5. Using the results of step 4, circles of proper radii are fitted to the points found in step 2. A steepest descent algorithm, augmented by Newton's method, finds the center coordinates which minimize the mean square distance of the surface points from the circle. Figure 5.5 shows the new circles fit.

6. Fixups are made, if necessary.

The centers of the circles fitted to the surface should represent an improved estimate of the axis of the generalized cylinder.

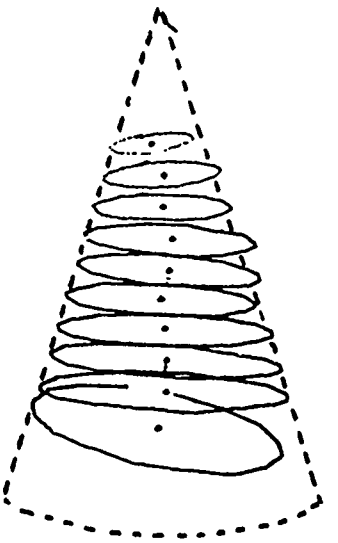


Figure 5.4
Preliminary Radius and Center Estimates

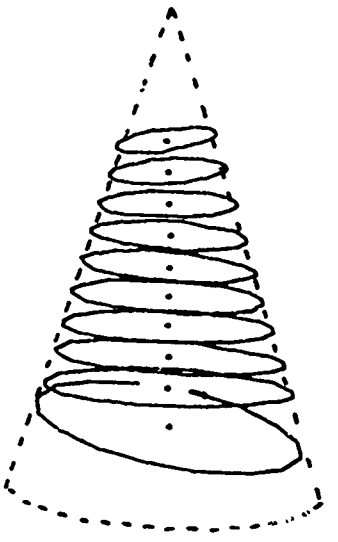


Figure 5.5
Cone with Smoothed Radius Function

A quadratic smoothing function is used to estimate the tangent to the axis, for orienting the cross section plane of step 1. Five sequential points are transformed into a two dimensional coordinate system for least squares fitting by a parabola. The tangent to the parabola, at the point through which we wish the cross section plane to pass, is transformed back into three dimensions. The cross section plane is taken normal to the tangent direction.

With a well defined generalized cylinder to trace, and in the absence of other nearby surfaces, the method usually converges rapidly. But when an error is made in fitting one cross section, the error must be detected and corrected, or subsequent applications of the curve tracer to the axis estimate will diverge to an incorrect or meaningless answer. Several types of errors are checked, and corrected when possible.

The ends of cylinders are checked first. The ends are susceptible to errors when extension of a cylinder has been terminated because of gross cross section changes. A new axis estimate at the end may cause part of the cross section at the end to include the wrong element. If the diameter of the end cross section does not agree (within a certain limit) with that predicted by the radius function Equation 5.1, or if the axis makes a sharp bend at the end, the end point of the axis is deleted from the cylinder.

For each interior point on the axis, the diameter of the cross section is checked against that predicted by the radius function. The angle made by that point and its two adjacent points is also checked; an error exists if this angle is less than 90 degrees. If an error is detected, a new cross section plane is specified midway between the two adjacent axis points. The parameter GAP, which controls the cross section determination (see Section 5.4), is either increased or decreased, depending on whether the diameter estimated by Section 5.5 is smaller or larger than the predicted diameter. The new cross section replaces the old, and the radius function is recomputed.

5.2 EXTENDING THE CYLINDER

Usually an initial axis estimate includes only a short section of a longer piece that may be described as a generalized cylinder. Extending the axis estimate finds the longest possible cylinder that may be conveniently described, and aids in segmentation of complex objects by locating gross changes in cross section diameter of an object.

Extension is from one end at a time. From one end, the axis is extended a fixed distance (determined by the average spacing between planes in the original axis estimate), and a single new circle added to the cylinder description, following steps 1 through 3 of Section 5.1. A circle of radius determined by Equation 5.1 is fitted to the cross section points, as in step 5. As long as the extension is compatible with the rest of the generalized cylinder, additional extensions are made.

Incompatibility of an extension with the rest of the cylinder may be signalled either by a sharp bend in the axis as a result of the extension, or by disagreement between the circle radius estimated by the method of Section 5.5 and the radius function Equation 5.1. (For our purposes, the radii agree if their ratio is between 0.66 and 1.5.)

When a disagreement or anomaly is detected, new constants are determined for the radius function of Equation 5.1, by repeating steps 4 and 5 of Section 5.1, and the extension attempted again. If the disagreement persists, the entire axis estimate obtained up till now is reprocessed by the cylinder tracer, and the extension attempted once again. (The finding of cross section points is time consuming. This is why a fixup to the radius function is tried before the entire axis is reprocessed.) If the attempt to extend the cylinder again fails, then extension is abandoned, and extension is started from the other end.

The cone used as an illustration for the previous Section was first extended toward the top. Figure 5.6 is the complete analysis of the cone after extension from both ends.

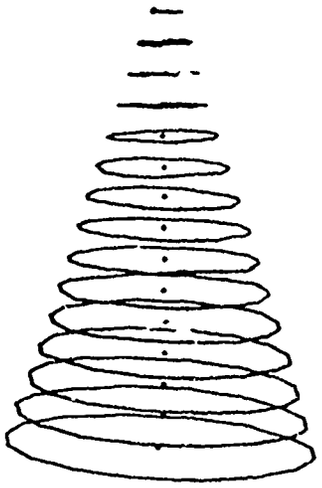


Figure 5.6
Complete Cone Analysis

5.3 PRELIMINARY GROUPING

An initial grouping and segmentation is carried out on an image to identify candidates for representation by generalized cylinders, and to supply an initial estimate to the cylinder tracer and extender.

This initial grouping is carried out in TV raster coordinates, before conversion of the fitted curves to three dimensions. Segments of laser traces which are adjacent and parallel are linked together to form "groups". The midpoints of the segments making up a group will be the preliminary axis estimate supplied to the cylinder tracer.

Figure 5.7 shows the groups extracted from the image of Figure 4.16. Only the depth discontinuities detected by the point linking routine (Section 4.2) are considered to segment laser traces; the segmentation performed for curve fitting (Section 4.3) is disregarded. For the purposes of preliminary grouping, each segment of a laser scan is represented as a straight line joining the end points of the segment.

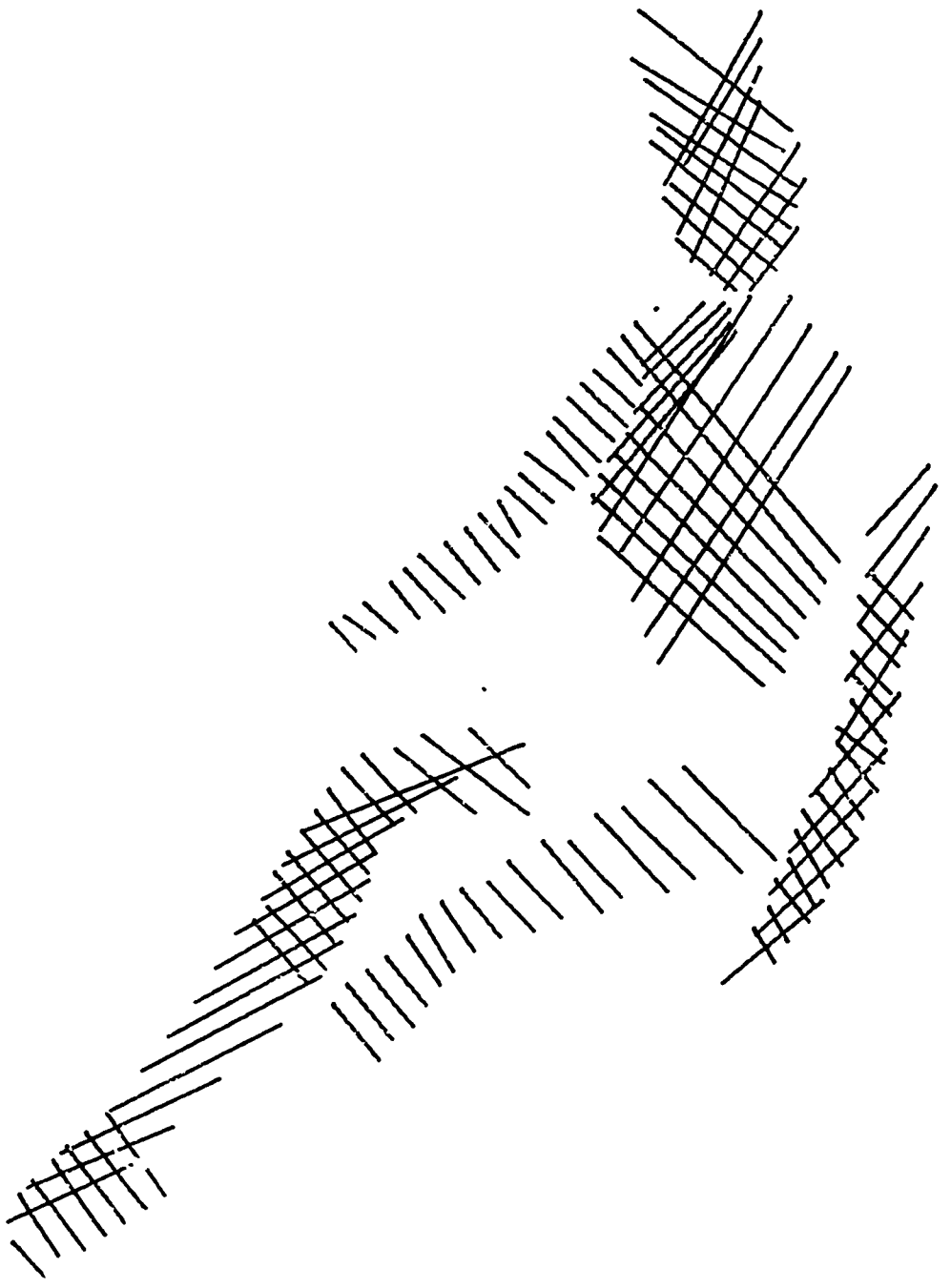


Figure 5.7
Preliminary Grouping on the Image of a Marble Doll

Figure 5.8
Preliminary Axis Estimates

Figure 5.8 plots the midpoints of the segments shown in Figure 5.7, which will be transformed into three dimensions and used as preliminary estimates for the cylinder tracer and extender.

The basic method of preliminary grouping is to link together line segments from consecutive laser scans on the basis of whether or not they are roughly parallel in their TV images. Segments from consecutive laser scans are linked together by the following test. The end points of each segment are joined by a straight line. If the perpendicular bisector of the first line intersects the second line between its end points, and the perpendicular bisector of the second similarly intersects the first, then the lines are "approximately parallel", and are linked together. If any line segment can be linked to either of two segments in any one laser scan, then no linkages are made. Figure 5.9 shows some examples. Lines 1, 3, and 5 will be linked together, as will lines 2, 4, and 6, and lines 7 and 8. No linkage may be made between lines 5 and 7, because the perpendicular bisector of line 7 does not intersect line 5.

A set of segments linked together is a "group". A check is made for gross changes in the length of segments making up the group. If such a change is detected, the group is divided at the discontinuity.

All groupings are extracted simultaneously in a single pass through the data structure.

A rough estimate of the radius of each cross section is required for the cross section point finder (Section 5.4). This initial estimate is computed from the length of each line segment, and the angle the segment makes with the line linking the midpoints.

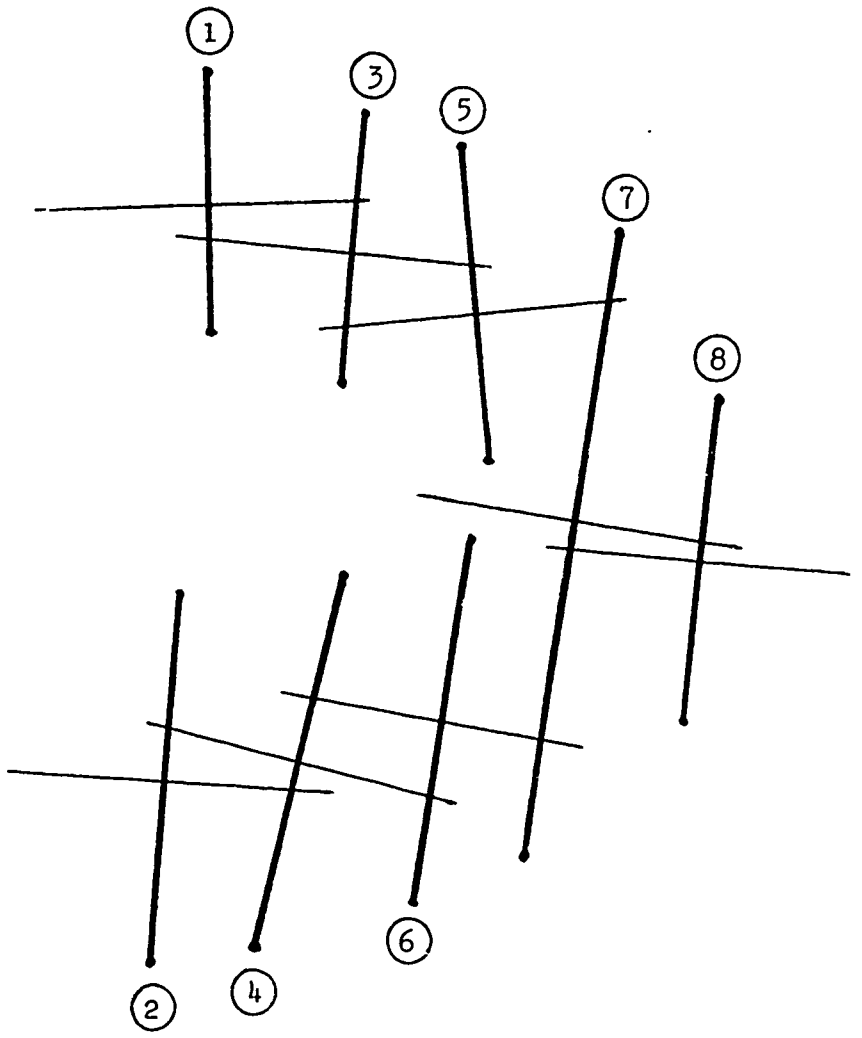


Figure 5.9
Some Scan Segments and their Perpendicular Bisectors

5.4 LOCATING CROSS SECTION POINTS

The cross section finder locates points on the surface of an object in the vicinity of a plane cutting the object. Since one plane may cut a complex object at several places, it is necessary to separate the points corresponding to the generalized cylinder being considered, from those that may belong to another part of the object.

The cross section finder requires the coordinates of a plane (in three dimensions) and an estimate of the center and radius of the circular cross section it is to find.

Two auxiliary planes are introduced, parallel to and on either side of the cross section plane. The distance between the cross section plane and each auxiliary plane is controlled by the distance between consecutive points on the axis being investigated. Now every curve segment obtained from the range data is examined to see if some portion of it lies between the two auxiliary planes. (Examining the coordinates of the end points of each segment eliminates most of the segments with little computation.) If some portion of a curve segment lies between the two auxiliary planes, the intersections of the segment with the auxiliary plane are entered into a list of cross section points, in a two dimensional coordinate system on the cross section plane. If the angle at which the curve segment intersects the auxiliary planes is small, intermediate points are also entered.

Figure 5.10 plots the cross section points found in a cross section across the body of the doll shown in Figure 5.25. (The points are viewed as if we were looking at the cross section plane from near the feet of the doll.) Note the group of points to the left of the main group of points. These correspond to the right arm of the doll. (The reason for the rather wide scattering of these points is discussed in Section 5.6.) The cross section finder must now separate these two groups of points, based on the supplied estimate of the radius and center of the circular cross section.

The cross section points are linked together into groups (equivalence classes) on the basis of proximity to one another. A threshold, GAP, determines the maximum distance between points to link them into the same group. The default value for GAP is determined by the mean distance between consecutive laser traces, but may be modified by the cylinder tracer (Section 5.1) to correct errors.

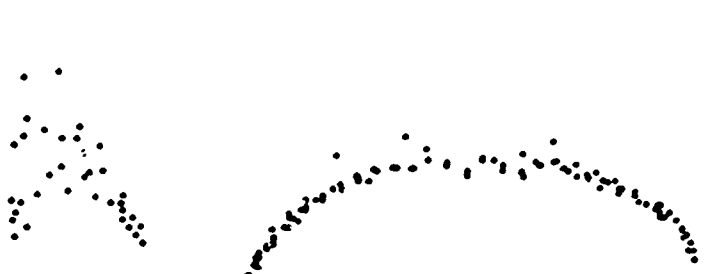


Figure 5.10
Cross Section Points

The estimated center of the cross section is transformed onto the two dimensional coordinate system on the cross section plane. The mean distance of each group from the estimated center is computed. Then all groups having a mean distance greater than the estimated radius are rejected.

In the points of Figure 5.10, two groups were identified. The group corresponding to the doll's arm was rejected by the distance criterion, leaving the points shown in Figure 5.11.

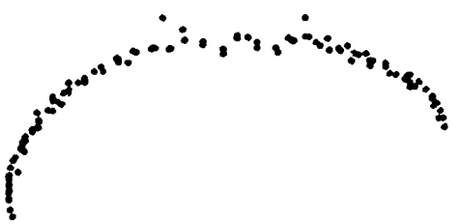


Figure 5.11
Cross Section Points Retained

5.5 RADIUS AND CENTER ESTIMATION

This Section describes how the known placement of the laser and TV camera may be used to estimate the radius and center of a particular cross section.

Given a circular cylinder with a straight axis, approximately half of its surface will be visible when viewed in ordinary illumination by a television camera. When the same cylinder is illuminated by a point source of light located at the laser center, approximately half of the surface will be illuminated. The radius and center estimation is based on the amount of overlap between the illuminated area and the visible area.

The estimation is based on the following assumptions:

1. That the surface we are exanining may be approximated by a circular cylinder,

2. That the cross section plane whose radius we are estimating is normal to that cylinder,

3. That the distance of both the laser and the TV camera to the surface is great compared to the radius we are to obtain, and

4. That laser scans are detected in the entire area which is both illuminated and visible.

We shall examine these assumptions later.

We shall proceed by assuming we have a cylinder whose radius is known (let us call it R), and calculate the degree of overlap. Figure 5.12 shows a normal cross section of the cylinder. In the figure, we plot the projections of the lines of sight from both the camera and the laser. The area from A counterclockwise to C is illuminated by the laser, and the area from B counterclockwise to D is visible from the TV camera. The area from B to C is the overlap. Let the angle between the projections of the laser and TV lines of sight be called Φ. Then the length of the line BC is given by

$$BC = 2 R \cos \Phi/2 \qquad \text{[Equation 5.2]}$$

furthermore, the length of the line OH (the normal distance from the center O to the line BC) is

$$OH = R \sin \Phi/2 \qquad \text{[Equation 5.3]}$$

In practice, a collection of cross section points is supplied, on a given cross section plane. To estimate the radius of the cross section, the extreme points of the collection are identified as the end points. The laser and TV line of sight projections on the cross section plane are calculated, and the angle between them. Equation 5.2 is solved to give the radius of the cross section, and Equation 5.3 is used to estimate the distance of the center behind the line BC.

The assumption that the surface may be approximated by a circular cylinder introduces some small error when the surface is conical. Equations 5.2 and 5.3 assume the camera and laser to be in the plane of the cross section. In the cylindrical case, the area of the visible surface is unaffected by moving the camera or laser in a direction perpendicular to the the cross section plane. In the conical case this is not necessarily so. For most "normal" viewing angles, the error introduced is small, but it should be borne in mind that in certain views the estimate will be off by a significant amount.

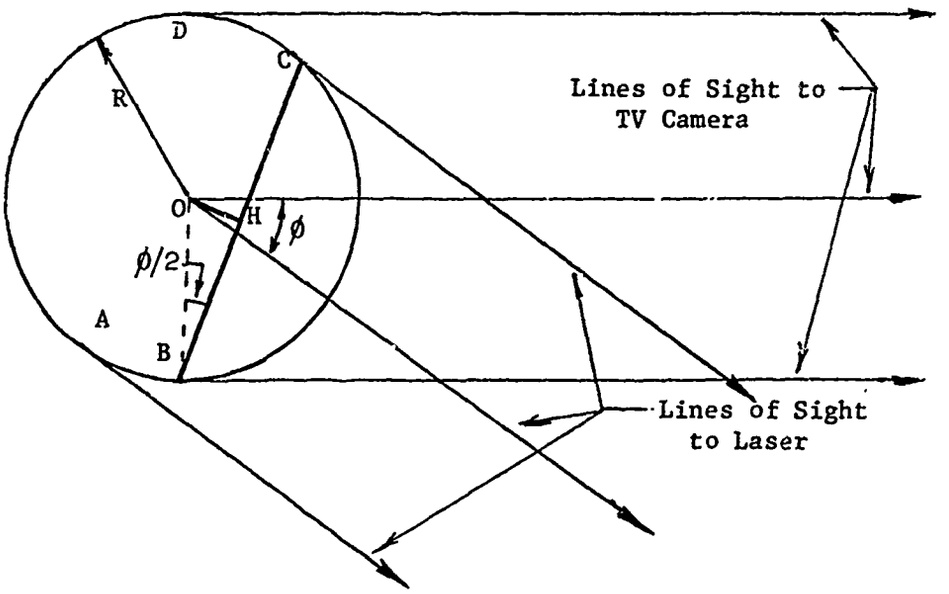


Figure 5.12
Radius and Center Estimation

The iterative nature of the cylinder finding operation guarantees that if cylinder finding converges, then the cross section planes will approach being normal to the axes of cylinders.

The distance from the laser to the object begin scanned is typically about 30 inches. Cross sections of cylinders seldom exceed 1 inch in radius. The error introduced by assumption 3 is not more than

sin (1 inch / 30 inches)

or about three percent.

The error introduced by assumption 4, that laser lines are detected in the entire area which is visible and illuminated, is the most difficult to estimate. When a laser trace illuminates a cylinder, the end of the laser trace becomes dimmer and dimmer as it approaches the limit of illumination. There must always exist some area of the cylinder in which laser traces may fall, but will be below the brightness threshold of detection. The width of this undetectable area will depend, among other things, on the

brightness of the laser beam, the color of the object, the sensitivity of the camera, and the orientation of the cylinder with respect to the laser. In one test, a detected laser trace was twelve percent shorter than it should have been if the entire trace were uniformly bright. But because of the variable nature of this error, it remains the largest single contributor to uncertainty of radius estimation.

5.6 SOME RESULTS

Our routines tend to do well on objects describable as a single generalized cylinder, and on portions of complex objects which possess considerable elongation. They give marginal results in the neighborhood of T-joints, and where nearby parts confuse the cross section finder.

The results pictured in this Section were obtained with some degree of manual direction of the program's execution. It was necessary for the operator to specify which areas of the picture to analyze, and to interpret the results as to success or failure.

This procedure could be carried out completely under program control, given a routine to detect overlap of proposed axis segments with previously identified cylinders, and a routine to determine success or failure of an analysis. Detecting overlap requires only a two-dimensional map of the image. An incorrect analysis is indicated by circle diameters many times larger or smaller than those initially predicted, and axis points which do not occur along any well-defined curve.

In order to present an unbiased picture of the capabilities and limitations of our program, the results presented here were all obtained with the same version of the analysis program. Some improvement in the results might be expected by adjusting parameters to give the best results for each particular picture. For each image we have analyzed, we have at least once obtained a better analysis. But the results shown here were obtained in the same evening, with no adjustments between one picture and the next.

Figure 5.13 is the processed image of a snake made of modelling clay. From an initial axis estimate comprising perhaps the middle third of the figure, the complete analysis of Figure 5.14 was obtained by extension of the ends.

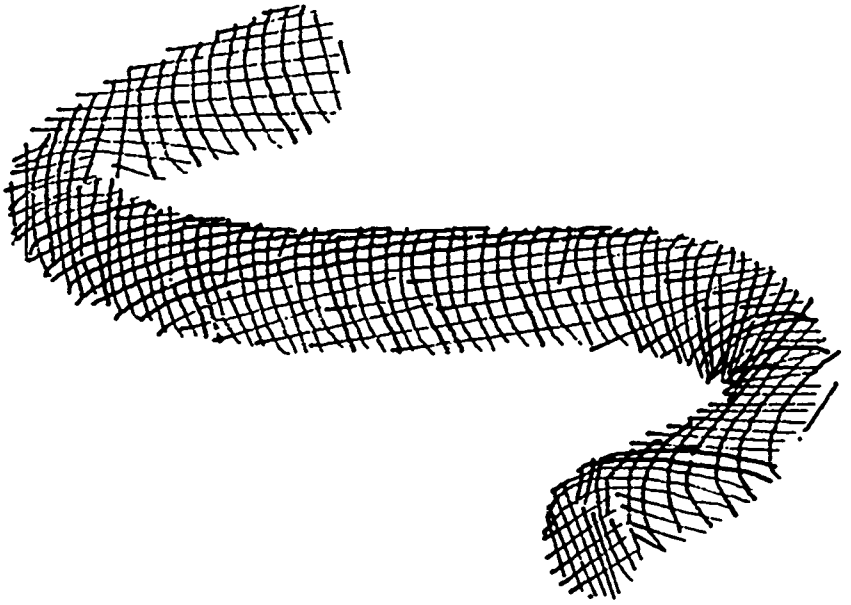


Figure 5.13
Snake

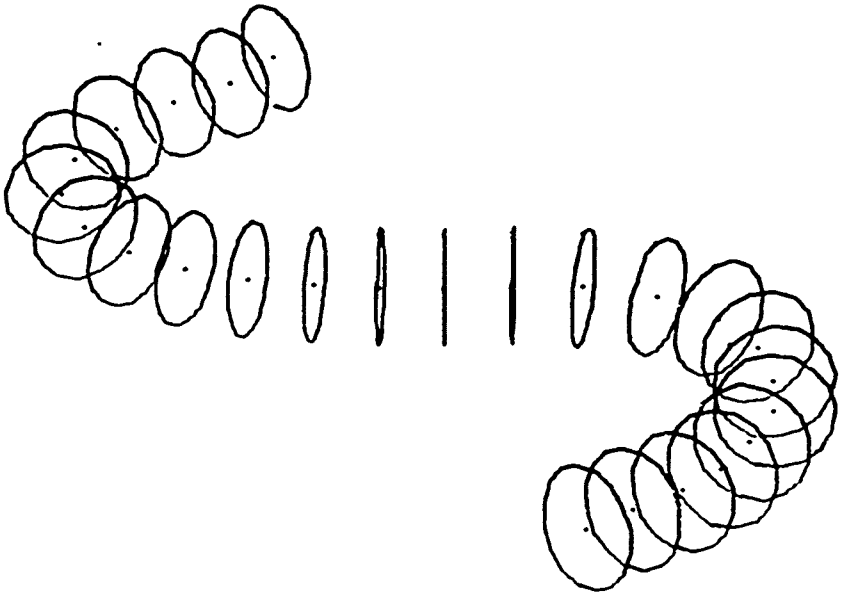


Figure 5.14
Analysis of snake

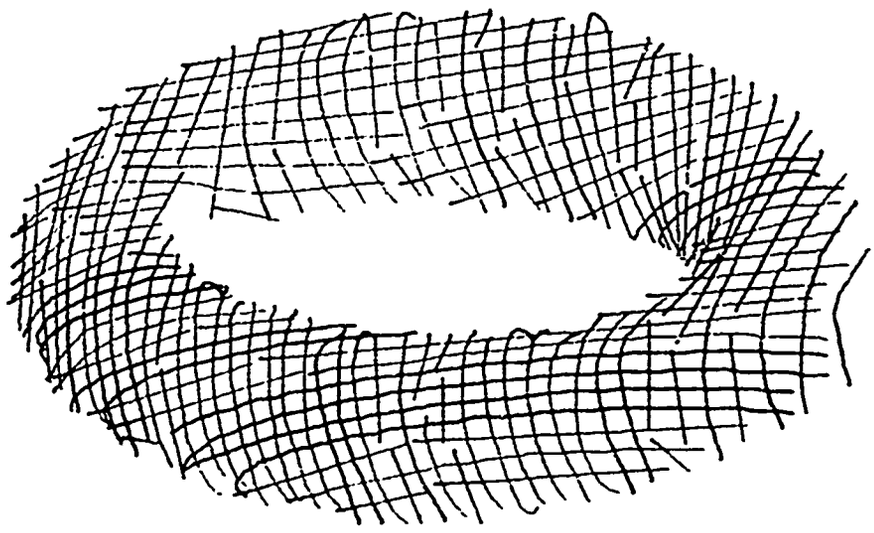


Figure 5.15
Torus

A torus (actually, a bathtub toy) is depicted in Figure 5.15. The initial axis segment which was assigned the highest value was in the lower part of the figure. Extension was necessary from one end only, resulting in the representation shown in Figure 5.16. The program detected that a closed curve had been traced, and that extension from the other end was not necessary.

When the analysis was tried on the toy hammer of Figure 5.17 the result was the description shown in Figure 5.18. Figure 5.19 shows the same description from a different viewing angle. Three axis segments were necessary to complete the description, one for the handle, and one on each end of the head of the hammer. The description of the bottom part of the head (as viewed in Figure 5.19) leaves much to be desired. The initial axis estimate for that part of the image follower for the most part the flat end of the head. Another pass of the cylinder tracer would most likely improve the description. If a T-joint were detected in the figure, and the handle deleted from the data structure as suggested in Section 4.1, an accepate description would almost certainly result.

Figure 5.16
Analysis of Torus

Figure 5.17
Hammer

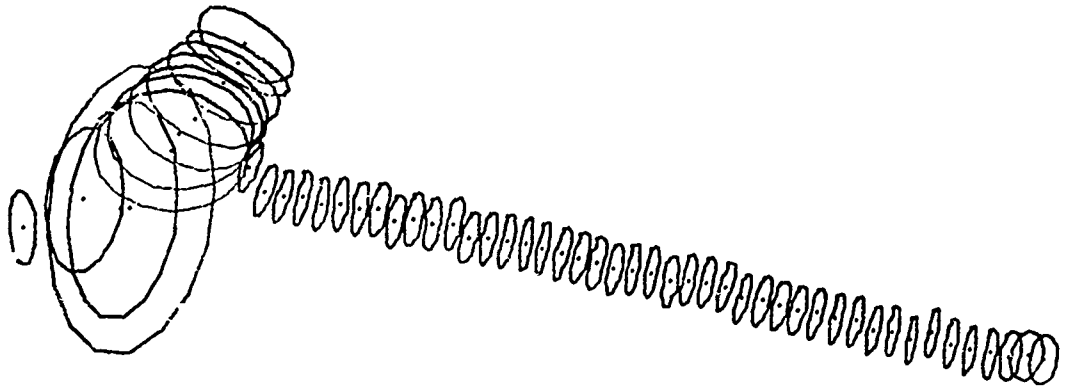


Figure 5.18
Analysis of Hammer

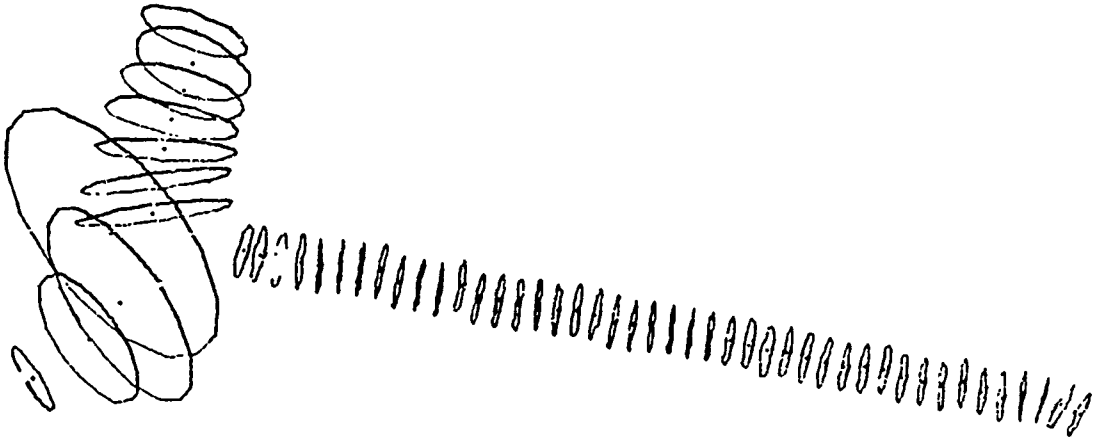


Figure 5.19
Analysis of Hammer, Top View

Frequently nearby parts of a complex object cause the analysis of an axis segment to go awry. Figure 5.20 is the image of a glove stuffed with tissue paper, and Figure 5.21 is the description obtained for the glove. In the analysis of the fourth finger, pieces of the third finger were included in the cross sections found. The situation we see here is due to too small an initial value for the threshold GAP for the cross section finder. When the fourth finger was analyzed, The small value for GAP allowed the axis of the finger to be extended into the palm of the glove, because cross sections across the palm were broken into small segments. As the analysis proceeded, the cylinder's radius function gradually became rather conical. When the cone became wide enough to include the base of the third finger, segments from that part were included in the cross sections.

The palm of the hand is not well represented by a circular cross section. In addition, the brightness of the laser was not adjusted properly, so that some of the laser scans near the wrist were not bright enough to be detected. Nonetheless, the analysis does give a description of the palm which is surprisingly good under the circumstances.

A toy horse is depicted in Figure 5.22. Unfortunately, when the laser was being set up to scan this particular object, it was not noticed until too late that the head of the horse was outside the bright portion of the laser line. Consequently part of the head is missing in the image. Figure 5.23 shows the cylinder analysis was able to identify seven generalized cylinders, comprising the body, the neck, the tail, and the four legs. Figure 5.24 is a different view of the same analysis which shows the neck, body, and tail to better advantage. Both left legs of the horse are partially occluded by the shadows of the right legs. For reasons that are presently unclear, the analysis of the body failed to stop at the rear legs, but continues to the tail. (In terms of a higher level representation, the extension of the body is desirable, but the cylinder extender is supposed to stop where a gross change in cross section takes place, and let a higher level routine decide if it wants the cylinder extended.)

In this analysis of the horse, both right legs were identified satisfactorily. But this is one of the better analyses of this image; usually only some small length of the legs is identified before the cylinder extender exits with a mismatch. Our program is poor at finding thin cylinders. Some indication of this may be seen in the bend in the axis of the right foreleg. There are several reasons for this deficiency.

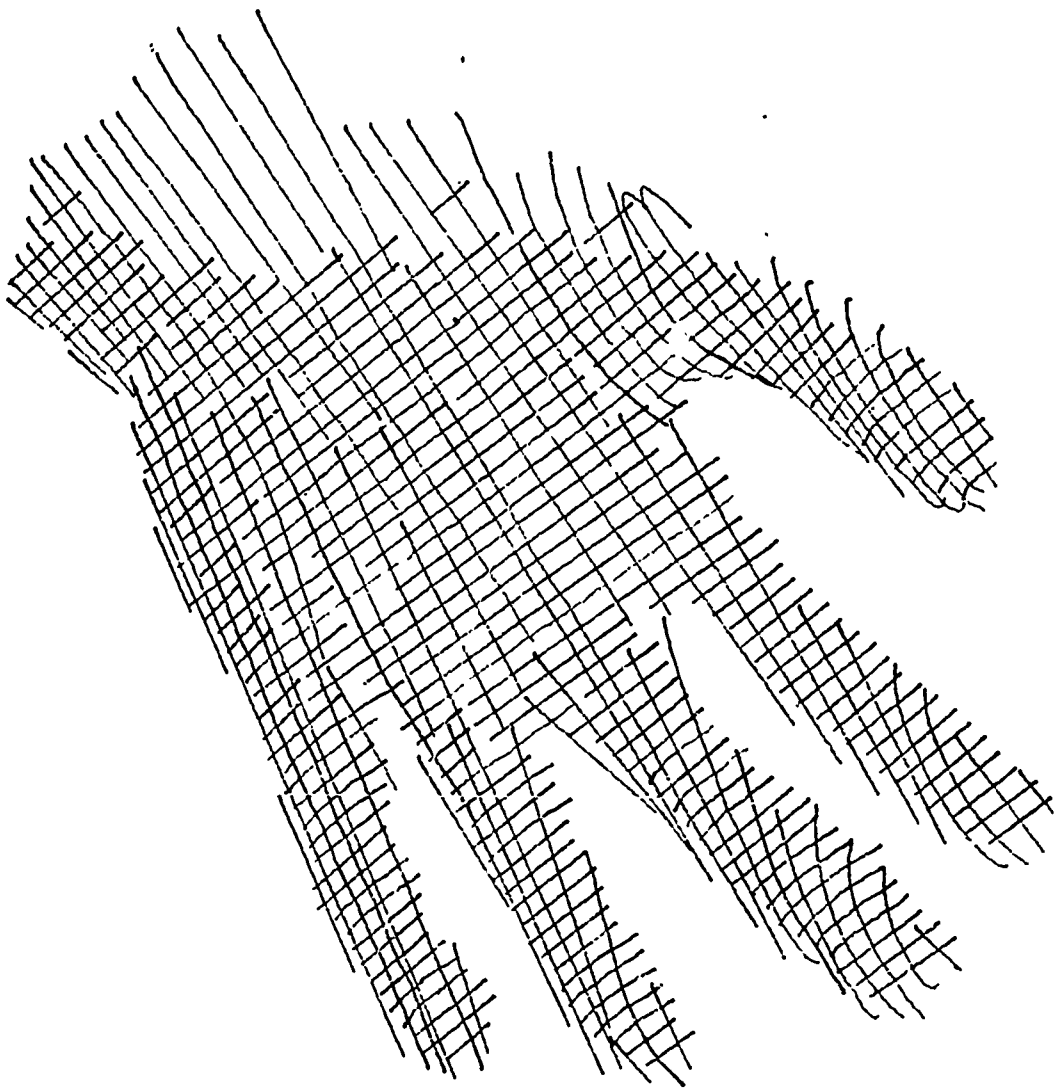


Figure 5.20
Glove

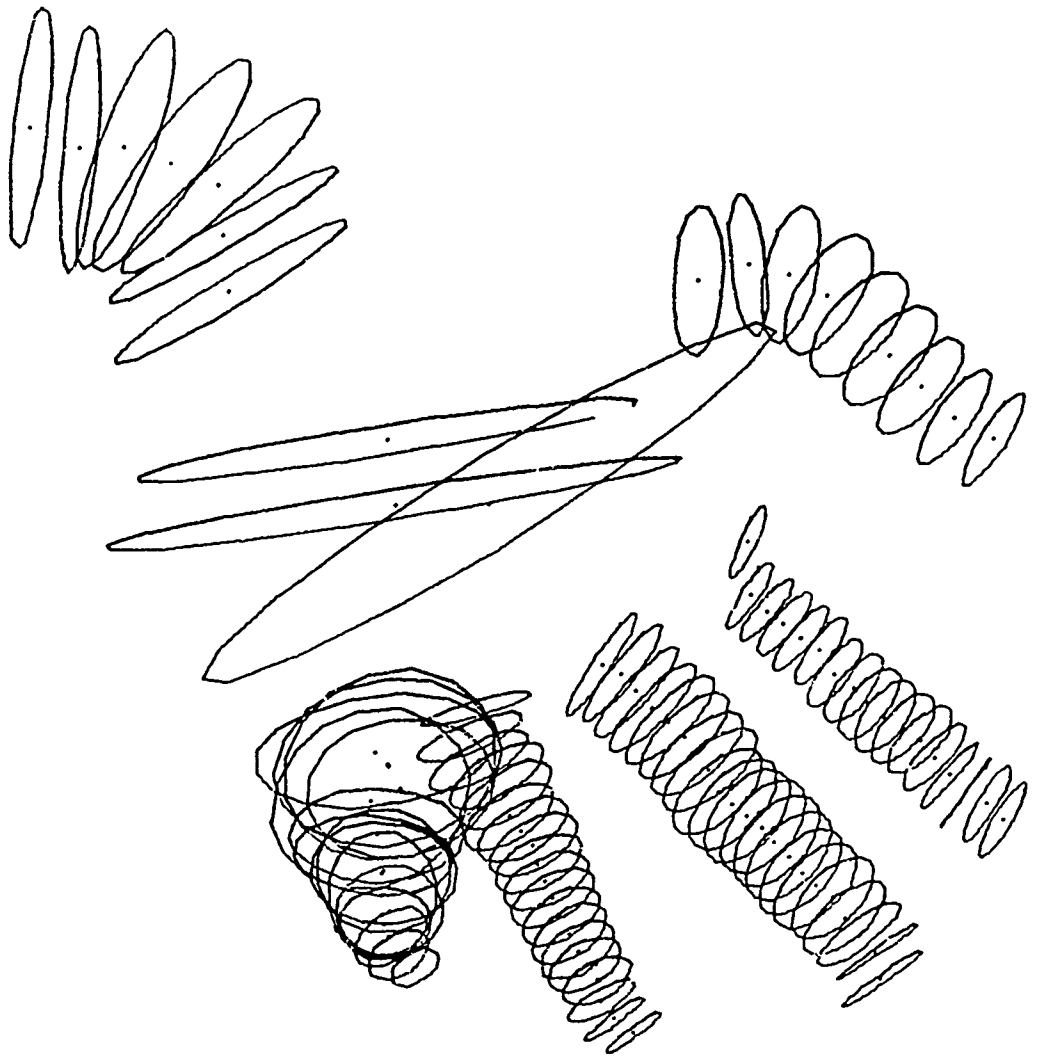


Figure 5.21
Analysis of Glove

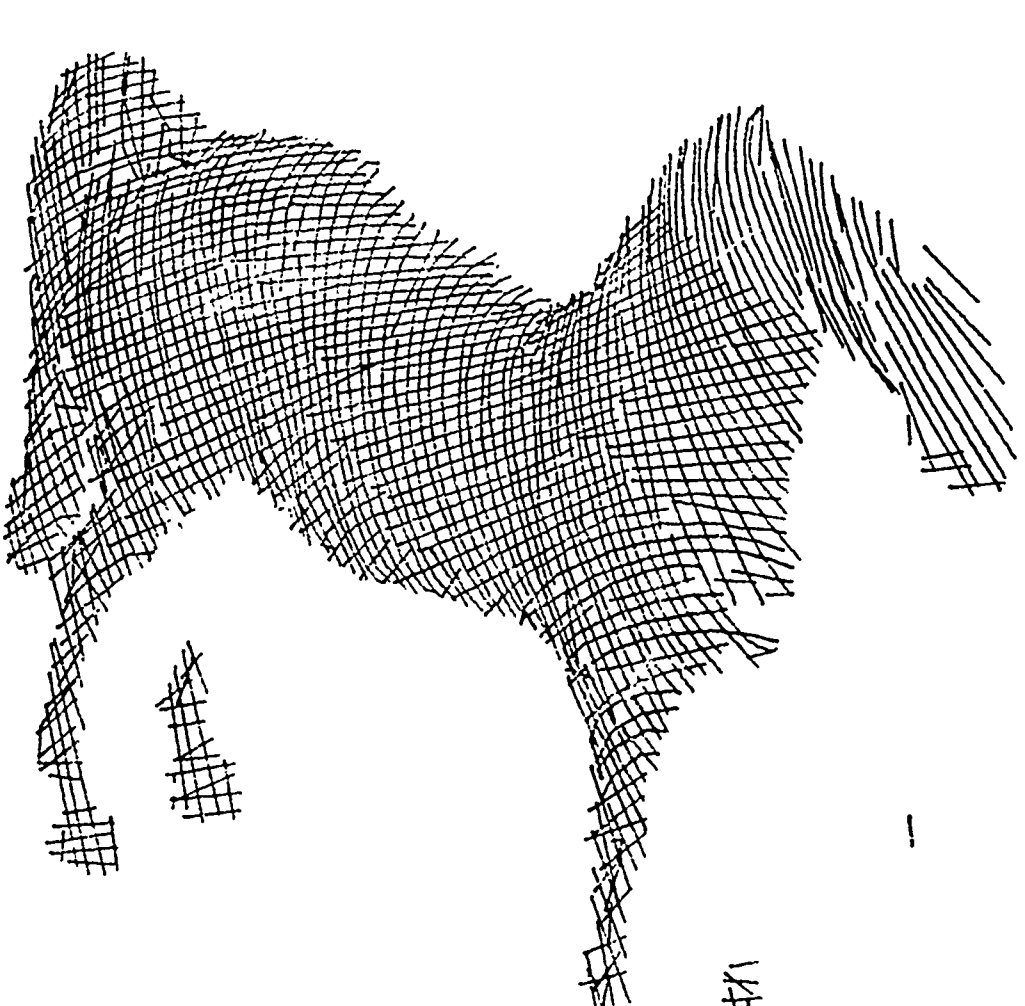


Figure 5.22
Horse

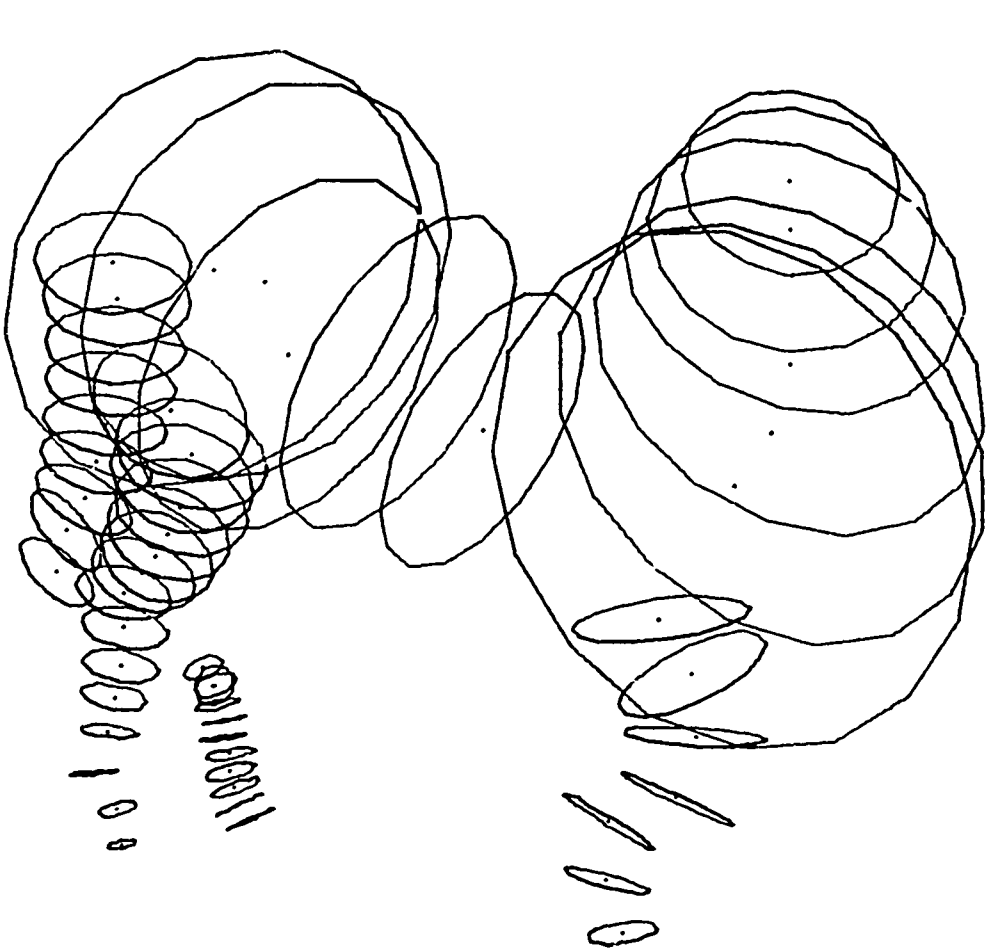


Figure 5.23
Analysis of Horse

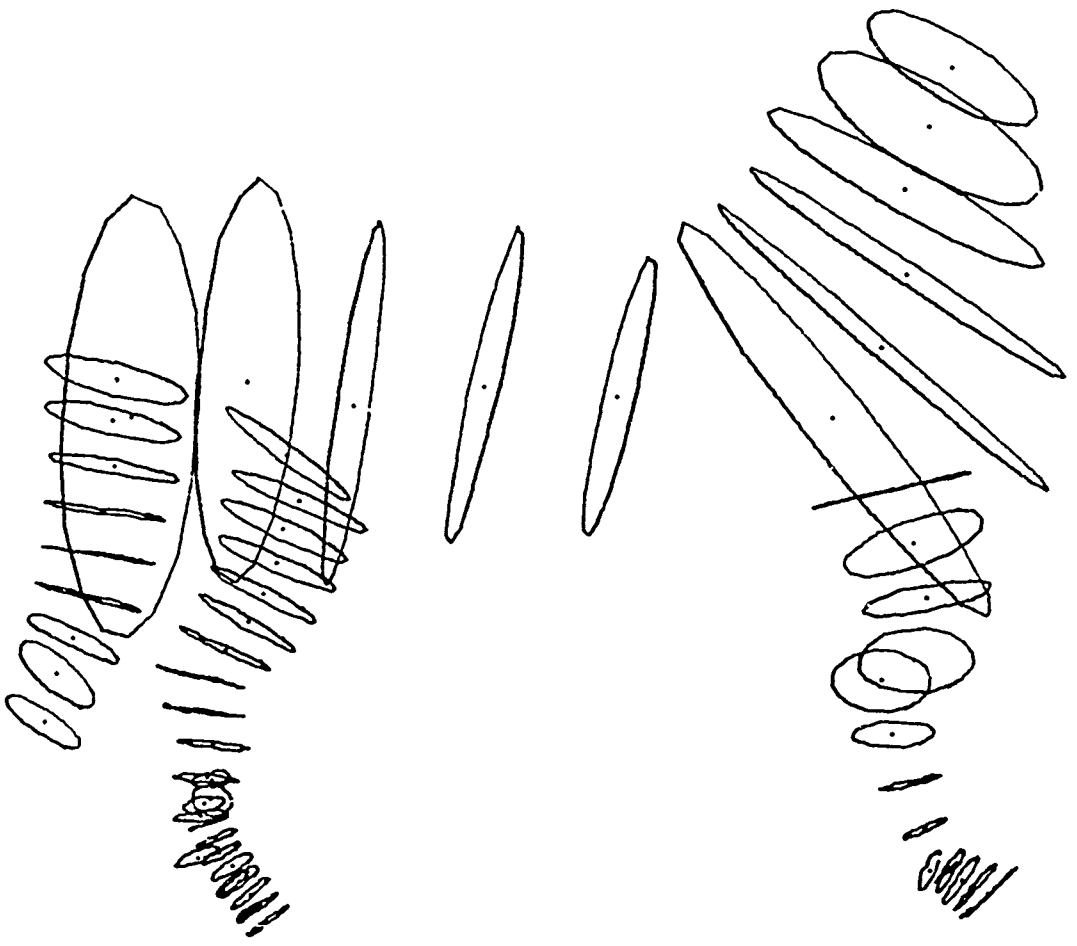


Figure 5.24
Analysis of Horse, Side View

The line linking routine (Section 4.2) rejects short line segments. The curve fitter cannot fit curved lines to segments of fewer than six points. The short lines tend to be relatively noisier, and tend to be fit badly by the curve fitter. Examining the legs of Figure 5.22, there is much less regularity of the curve segments than in the broader areas such as the body or the neck. And when the cylinder tracer searches for small cross sections, there is less margin for error in guesses of the centers of cross sections. The result of these factors is that the cylinder finder usually fails to trace the complete legs of the horse.

An example of the noisiness of fitted line segments to thin segments may be seen in Figure 5.10, where the points representing the arm of the doll are much more widely scattered than those of the body.

To fix this sort of error, it will be necessary to fix the point linking routine to accept shorter line segments (and to put up with more noise in the rest of the image), perhaps to modify the curve fitter, and to allow for greater errors in the cross section finder when the cross sections being sought are small.

Analysis of the baby doll shown in Figure 5.25 resulted in the description of Figure 5.26. The routine failed to make a segmentation between the right leg of the doll, and its body. This is the same sort of error as that which caused the body of the horse to extend toward the rear. The cylinder extender managed to continue the analysis of the right leg to include the foot. Separation between the right arm and the body is good. The head is not well described.

Figure 4.16 is reproduced as Figure 5.27. The results of the analysis of this picture are shown in Figure 5.28. The description of the body is poor, because of its non-cylindrical shape and because the presence of the left arm nearby confused the cross section finder. There is little separation between the legs, but the program found both knees, where the separation is greatest. In analyzing the right calf, separation between the legs was small enough to cause the cylinder finder to regard both legs together as a single cylinder. This cylinder was extended upward toward the body, overlapping the previous analyses of the knees. The short cylinder representing the left arm is evidence of difficulty with thin cylinders first mentioned in connection with the horse.

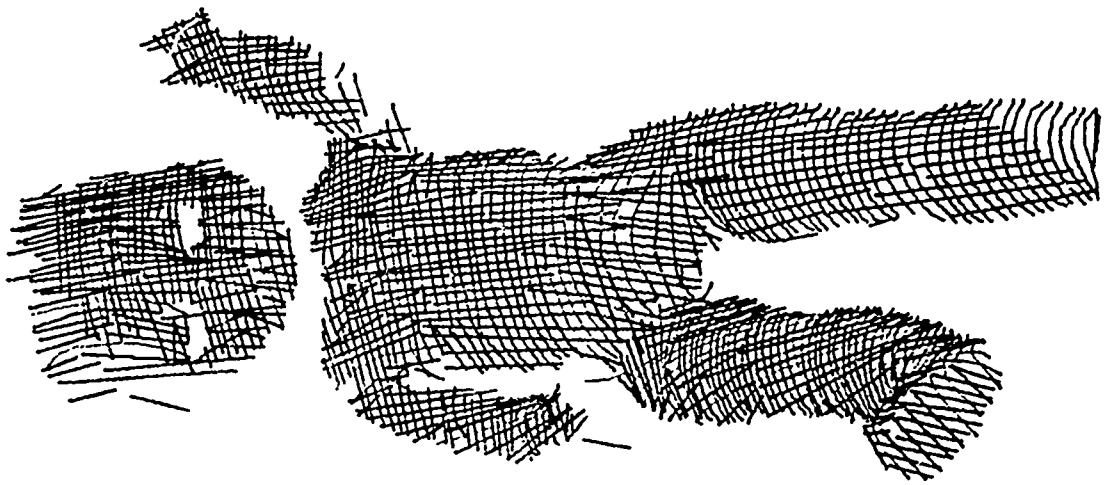


Figure 5.25
Doll

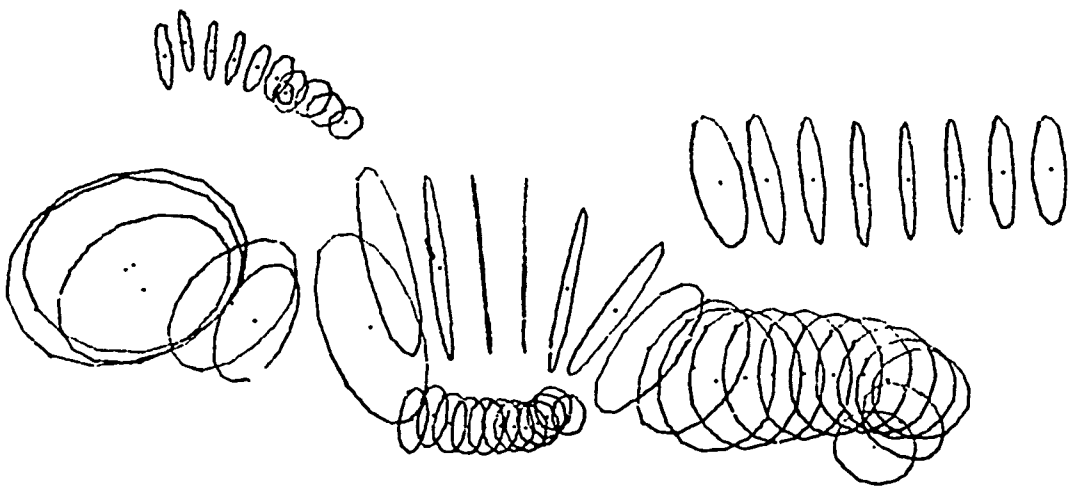


Figure 5.26
Analysis of Doll

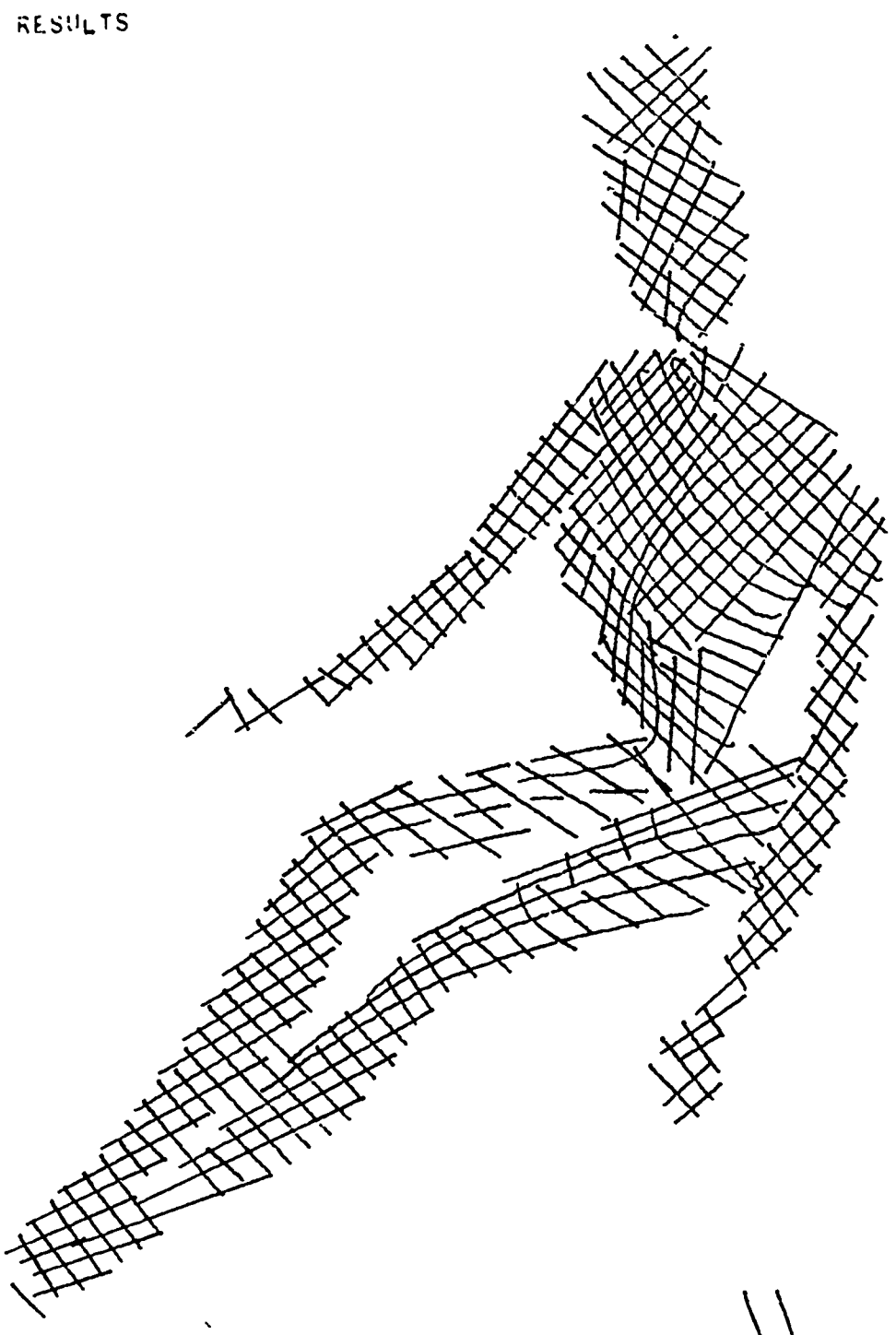


Figure 5.27
Barbie Doll

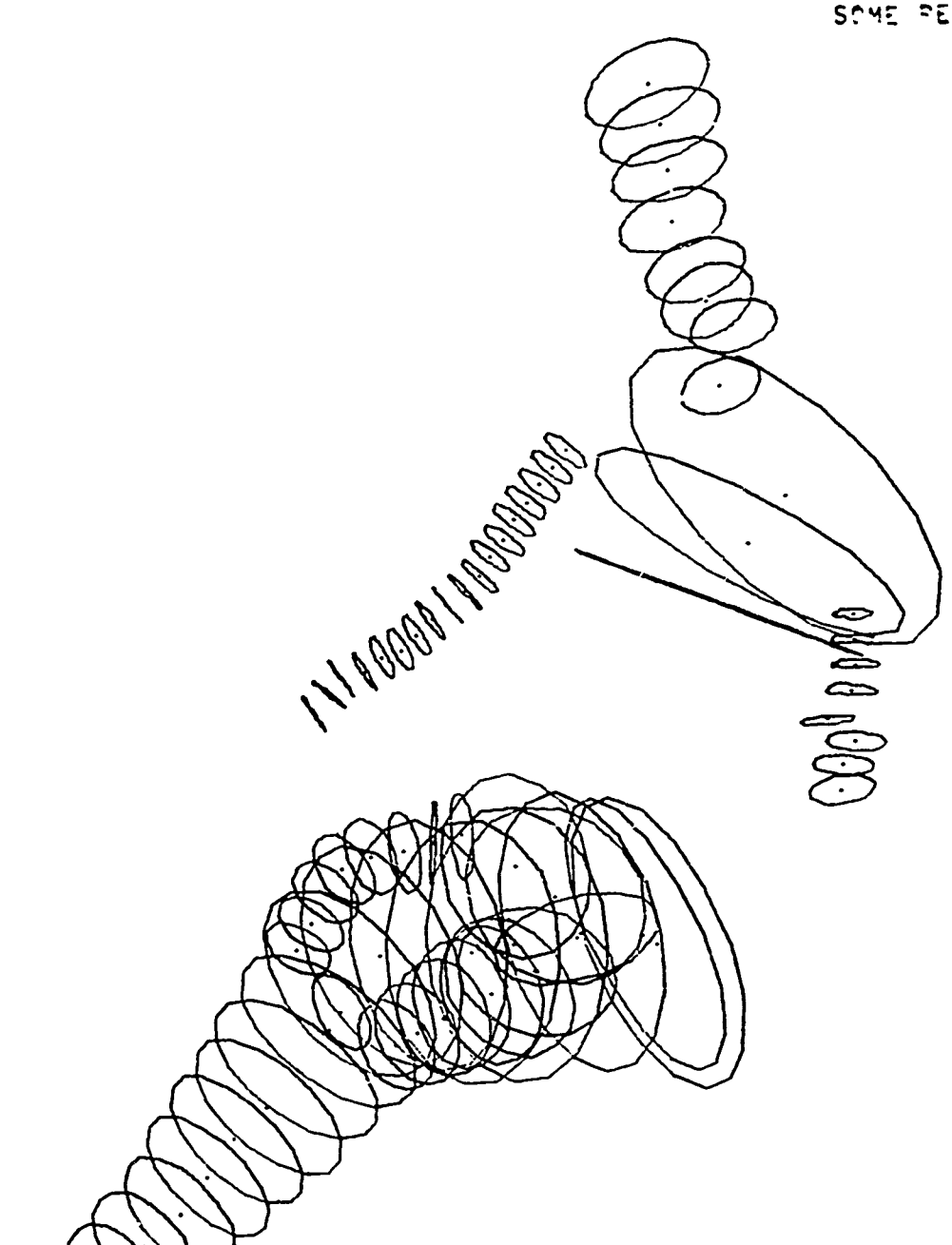


Figure 5.28
Analysis of Barbie Doll

The program which performs the preliminary analysis and the cylinder tracing and extending requires between 42 and 56 K of core on the PDP-10 computer. This includes from 3 to 17 K of data arrays, 11 K of display buffers, and 7 K for an interactive debugging package, RAID. (See [Petit] for a description of RAID.) Simple figures such as the ring or the torus may be processed in about one minute of computer time. The hammer required two minutes to process, while the horse, the glove, and both dolls each required between five and seven minutes.

6 DESCRIPTION OF OBJECTS

The state of the experimental portion of our research may be seen from the examples in the preceding chapter. A great deal of work remains before we may proceed from laser data from an arbitrary real object, to a full description in terms of our representation. The material in this chapter contains some suggestions as to how this might be accomplished in the near future, and some of the problems which will probably be encountered.

The principal mode of operation of a description program should be to build a hypothesis on the available data, test the hypothesis, and use the results of the test to build a better hypothesis. The hypothesis generated will depend on the data. Special purpose routines may examine the data to generate specific types of hypotheses. For example, a routine might postulate the existence of "missing" joints on an object based on structural relationships and on proximity. Verification of a hypothesis may be a simple "go - no go" test, such as surface continuity in the present example, or might yield additional data, such as the existence of previously unnoticed details. The results of the test, in either case, should be placed in the data structure to enable better hypothesis generation in succeeding iterations.

6.1 LINKING SEGMENTS TOGETHER INTO A SKELETON

The most logical next extension of our research is the construction of a program which will make use of the cylinder-finding routines to generate complete descriptions of complex objects. As a first pass, some joints between segments may be discovered immediately, but higher-level considerations concerning the structure and connectivity of skeletons will eventually have to be considered.

A useful routine in this and later phases will be one to check surface continuity between two or more segments. The analysis routine should select points on the surface of each segment, and the continuity routine should search in a narrow band between the points for the existence and spacing of laser traces. If a continuous path on the surface is not found, the continuity routine should be able to make small deviations from the initial path. If continuity is not indicated for this particular pair of points, the analysis routine should be prepared to suggest other point pairs. Heuristic considerations

will dictate the number of joint pairs to be analyzed before continuity is rejected.

The easiest joints to treat will be the end-to-end joints. Candidates for this type of joint will have the end points of their principal axes located close to one another, and linked by surface continuity. More than two axis segments with surface continuity to the joint candidate will indicate that the joint is not of the end-to-end type.

Bifurcations and multifurcations will probably also be easy to recognize. The conditions for a bifurcation to exist are similar to those for an end-to-end joint. One end of one skeleton segment will have surface continuity to each of two other segments, the "active" end of the principal segment lying between the "active" ends of the subordinate segments. The subordinate branches should have cross sections of comparable size.

T-joints will present a more difficult case. The initial pass of the cylinder finder will not usually trace the principal segment (the "crossbar") through the area of the joint, because of the abrupt change in apparent cross section where the subordinate segment (the "upright") joins it. Sometimes the preliminary grouping analysis will miss one or both halves of the principal segment on either side of the joint we wish to find. The analysis of the hammer, Figure 5.19 provides a good example of what might be expected. Cases where the crossbar has been analyzed as a single continuous cylinder will be easy to detect. But candidates for T-joints will also come from cases where at least two skeleton segments are linked by surface continuity to an area that is not part of any known skeleton segment. What will probably be necessary is to temporarily delete from the data structure any skeleton segment which is suspected of being the "upright" of a T-joint, and request another analysis of the suspected "crossbar" from the cylinder finder.

Perhaps these steps will provide all the links necessary to describe the complete skeleton of an object. More likely, there will be areas of the picture which have no corresponding skeleton segments, segments with substantial areas of overlap, disconnected fragments of skeletons, and joints with ambiguous interpretations.

Of the results presented in the previous chapter, the snake and the torus will need no further identification. The hammer and the glove might be successfully linked into objects by the

simple methods we have proposed. The horse and the two dolls will need further analysis. In the case of the baby doll, we might expect the preliminary linking to correctly identify an end-to-end joint between the head and the body, and a T-joint between the left arm and the shoulder. Because of the failure of segmentation between the doll's right leg and its body, the left leg would be incorrectly joined to the body/right leg in a T-joint. The right arm would not be linked to any other part.

The next likely step will be to call into play a routine to suggest plausible interpretations for the data. Many of these suggestions will be subject to verification by the cylinder finder and the surface continuity routine. This routine might suggest a T-joint between the doll's right arm and the hip, and reject that on the lack of surface continuity. It might suggest a bifurcation with the right arm, head, and body, but reject that hypothesis because of the disparity between cross section sizes. It should also suggest, and settle for, a T-joint between the right arm and the shoulder.

If at the end of this analysis we have more than one possible analysis of the object, another routine should rank order the choices on the basis of simplicity and agreement with the data, and either choose the best description, or submit to the next higher level of control the ordered list.

6.2 DEALING WITH NON-CIRCULAR CROSS SECTIONS

The present description program assumes all cross sections it encounters are circular. Obviously, the program must deal with more general cross sections if it is to deal with a wider class of objects.

Many of the cross sections we will be dealing with will be small with respect to the resolution of our data. Shape recognition may be performed on large cross sections one at a time. But to properly identify or classify the smaller cross sections it will be necessary to consider several neighboring planes simultaneously.

The cylinder fitting routines already implemented take many cross section planes into account by fitting each profile individually, then smoothing the radius function of axis length by a linear least squares fit. When dealing with arbitrary cross sections shapes, it may still be possible in some cases to fit each profile individually, and to merge or check the results

against neighboring cross sections. But to handle cross section determination correctly, in general it will be necessary to merge the raw profile data before performing fitting or recognition. This will necessitate finding a transformation or set of transformations to bring the data from different cross section planes into correspondence. This is direct application of generalized translational invariance to find the primitives it generates!

Two basic cross-section shapes will account for most of the common objects a recognition system might have to deal with. Ellipses alone will probably model quite successfully most animate objects, and with the addition of rectangles, will handle a large number of manufactured objects. Both ellipses and rectangles may be characterized by their major and minor dimensions, and their angular orientations with respect to the skeleton. The angular orientation may be specified by the orientation of the auxiliary coordinate system mentioned in Section 2.3. (This is the first time we have needed to use this auxiliary coordinate system. Circles are indifferent to the angular orientation of the axes on which they are drawn.)

Differentiating between ellipses and rectangles will not be difficult. For each rectangular cross section, either one face or two faces of the profile will be visible simultaneously to the laser and the TV camera. The discriminating routine will attempt to fit one or two straight lines to the points of the cross section. If this fails, the cross section will be designated an ellipse. This type of differentiation is error prone in individual cases, but comparison with the analyses of neighboring cross sections should cancel most errors.

The dimensions of rectangles, where two faces are visible, may be obtained directly from the cross section profile. Where only one face is visible, higher level considerations will have to estimate the missing dimension. For ellipses, a method similar to that used for estimating the radii of circular cross sections (Section 5.5), based on the known positions of the laser and TV camera, will give the principal dimensions. Comparison with neighboring cross sections will reduce the errors involved.

Where yet more general cross sections must be recognized, there are two possible directions for the analysis to proceed. Either the cross section analysis must choose among some finite set of possible shapes, or assumptions must be made regarding the regularity of shapes involved. The former case is a classical pattern recognition problem. The best approach to this type of

analysis would be to write special-purpose routines to isolate features of the curves, on which recognition may be based. In the latter case, some possible regularity assumptions are N-fold rotational symmetry, and bilateral symmetry about some predetermined axis. If more than one regularity assumption may hold, the program must choose which one to apply, based on measurable features of the shape of the profile.

Recognition of cross section shapes need not be done in a single pass. A preliminary analysis could assume circular cross sections to find the skeleton of an object. Higher level considerations based on analysis of the skeleton would then influence a subsequent detailed cross section examination.

6.3 SEPARATION OF OBJECTS IN A SCENE

Where there are multiple objects in a scene, the objects must be separated one from another, so they may be dealt with separately. Our existing implementation makes no attempt to do this, and will deal only with single objects, with the background (the tabletop) suppressed.

If the scene to be analyzed is on a tabletop, calibration information will give the homogeneous coordinates of the plane of the table. The line segments corresponding to the tabletop are those that lie in or near this plane, and may be removed from the scene with no difficulty.

Body separation may be done utilizing depth information. We shall use the hypothetical scene of Figure 6.1 to illustrate. An edge drawing is made of the scene, as viewed from the TV camera. The edge drawing will include only the depth discontinuities and concave edges in the scene. A depth discontinuity exists at each of the end points of the line segments isolated by the point linking algorithm (Section 4.2). A concave edge point exists at those points where recursive segmentation (Section 4.3) divided a laser trace, and the tangents of the ends of the segments form a concave corner (as seen from the TV camera). (It may also be necessary to consider all second order curves which are concave from the TV camera, and denote a concave edge point at the point of highest curvature.) Figure 6.2 shows the depth discontinuities and concave edges to be found in Figure 6.1.

The areas inside closed curves may be isolated as bodies. Figure 6.2 contains three closed curves which isolate the ell-block, the cylinder, and the cone.

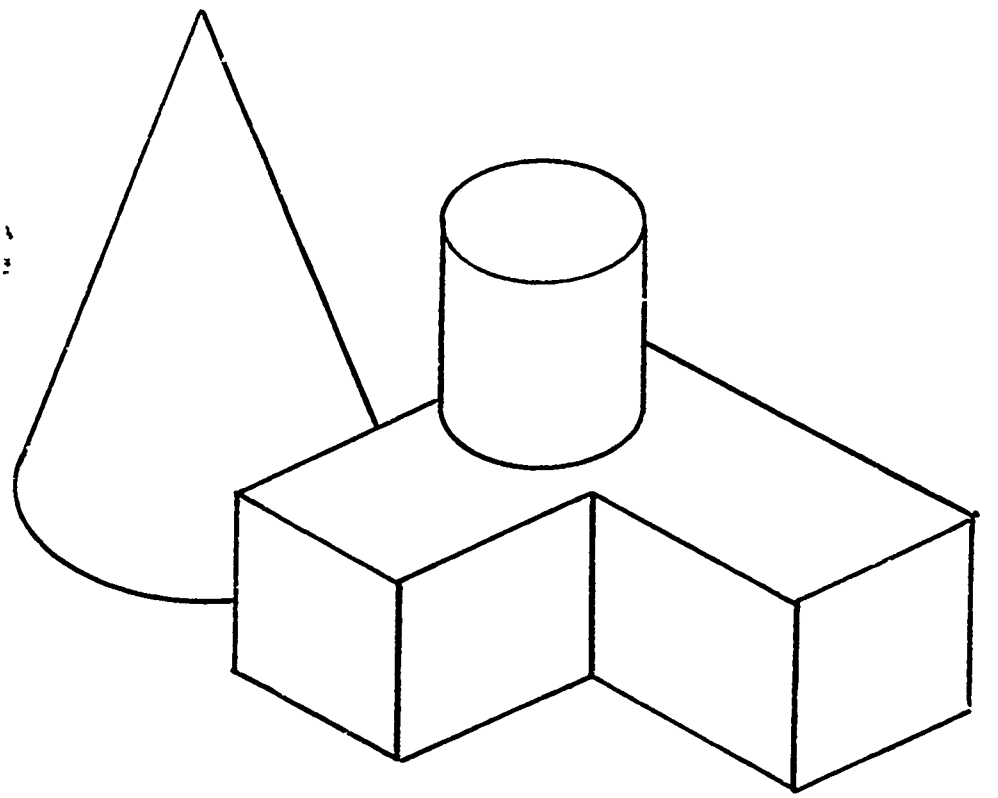


Figure 6.1
Scene with Multiple Objects

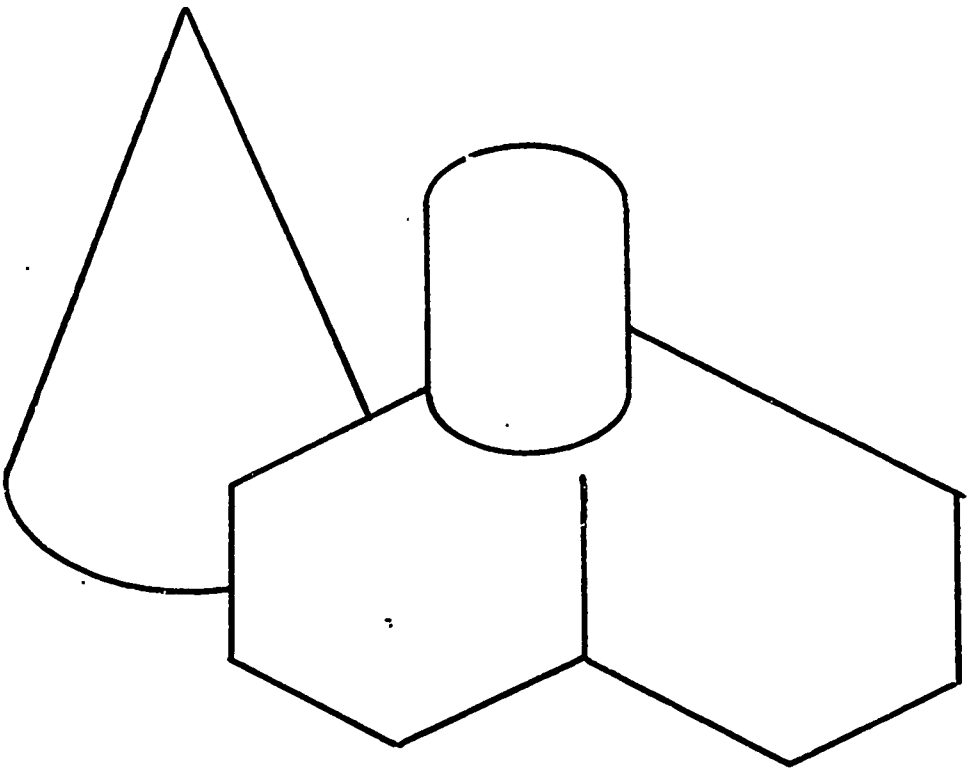


Figure 6.2
Depth Discontinuities and Concave Edges

Frequently this method will segment a scene into discrete bodies that are part of the same object. The hammer of Figure 5.17 would be segmented into two parts by this method. The segmentation will be useful in identifying primitives, but the higher level subroutines should have the option of recombining parts. This will be possible only when the program has some a priori knowledge about the context of the scene.

6.4 OCCLUSION

Occlusion in a scene may arise in two ways. The more familiar case is where one object hides a portion of another, as seen from the point of view of the camera. The other case is where the shadow of one object falls on another object. These two cases are manifestations of the same phenomenon: when the scene is viewed from the point of view of the laser deflection assembly, the two cases reverse.

The depth discontinuity edges drawn for body separation (Section 6.3) may be used to identify edges that arise from occlusion from the camera. Repeating the procedure, where the TAB takes the laser point of view, will identify the laser's occlusion edges. Figure 6.3 shows a narrow cone in front of a snake-like object, in front of a wide cone. Figure 6.4 shows the same scene, viewed from the laser. The duality between types of occlusion edges should be apparent from these two views. Where two contiguous areas (in either view) are separated by a depth discontinuity edge, that edge is labelled an occlusion edge on the object farther from the point of view.

Where occlusion edges are perpendicular to the axis of an object, as where the tip of the cone hides a portion of the snake in Figure 6.3, the pieces may be joined by looking for the continuation of an axis segment on the other side of the hiding piece. A more difficult case to cover is where the occluding edge is parallel to the axis of the body we are examining. Such is the case where the snake hides part of the wide cone in Figures 6.3 and 6.4. In a context of circular cross sections, we might estimate the radii of curvature of the visible portions of the wide cone, and unite the two portions on that basis. In a sufficiently limited context, other forms of surface fitting may work.

In general, dealing with occlusion is extremely difficult. But the use of three-dimensional information gives us some additional tools for dealing with the problem.

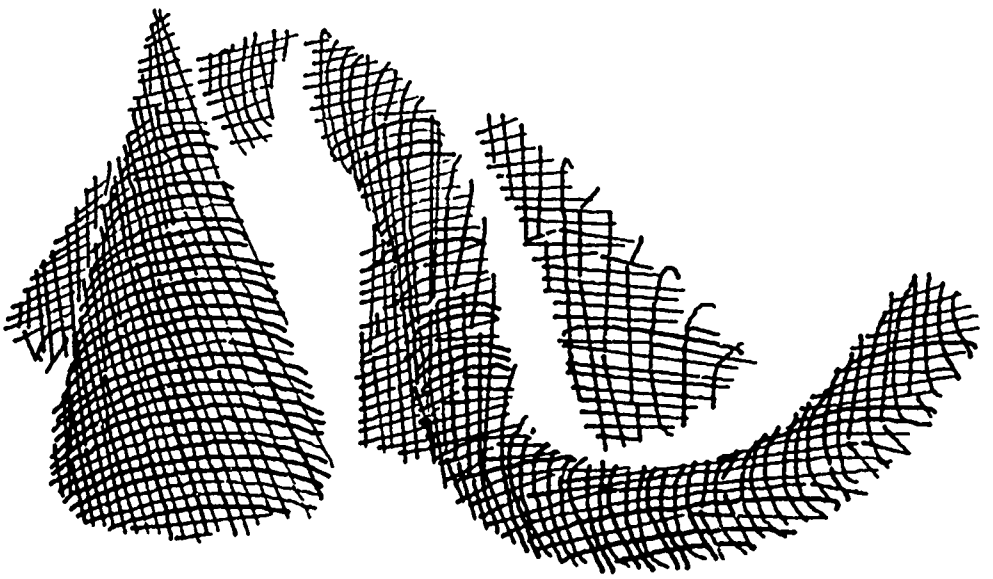


Figure 6.3
Scene with Occlusion, from Camera's Viewpoint

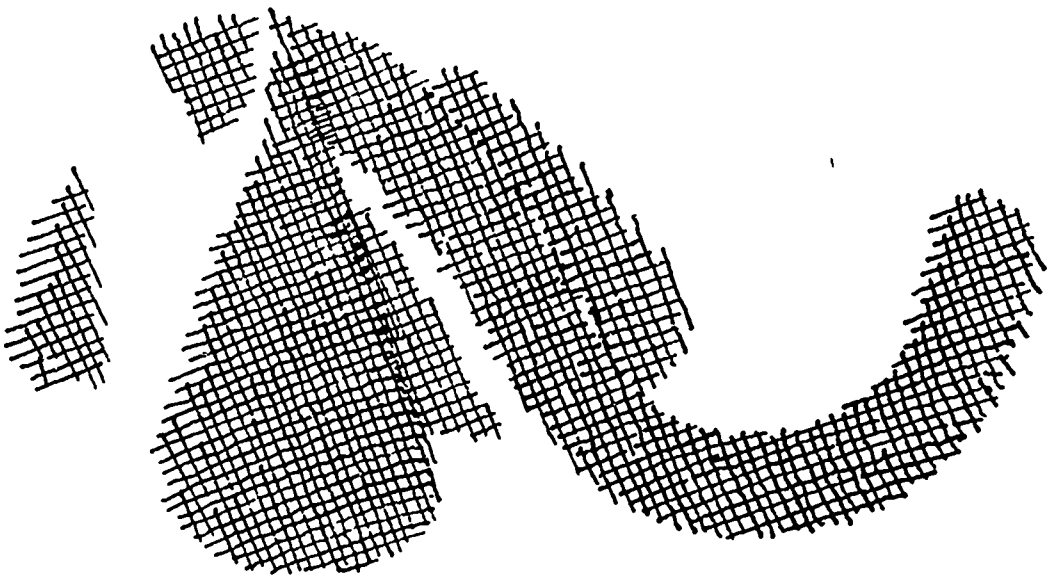


Figure 6.4
Scene with occlusion, from Laser's viewpoint

6.5 RECOGNITION OF OBJECTS

A test of the effectiveness of our description program will be to put the descriptions to use in recognition. This involves comparing our generated descriptions against stored descriptions of the objects among which we are to choose for identification.

The non-uniqueness of our representation, as well as errors to be expected in the description, make it unlikely that the generated description of any object will exactly match its corresponding stored description. This means it will be necessary to develop measures of similarity between descriptions.

If the number of possible identifications is small, similarity might be measured by special-purpose subroutines. The recognition phase of our program would then involve calling the subroutine corresponding to each possible response, and reporting which subroutine gave the best degree of match. But as the number of possible objects increases, this becomes inefficient.

Classification of skeletons based on their topological and geometric properties will do much to reduce the search space. We must not place too much reliance on the classification of joints made by the description program. The same articulated joint, in different attitudes, might look like a T-joint one time, a bifurcation another, and an unclassified joint joining three skeleton segments a third time. The topological classification must depend only on the number of segments at each joint, and not any designation of hierarchical organization or principal axes for any part. While this classification will not always give the correct answer the first time, searching under similar classifications for matches should find the right match for a given description.

Another technique for recognition is a decision tree. Specific properties of a description are tested for, and the results determine further testing until a terminal node is reached. Proper design of the tree can result in efficient testing. But construction of the tree is tedious, especially for a large class of objects. Where parts of the generated description are liable to be incorrect or even missing, multiple paths must lead to the same terminal node.

7 CONCLUSIONS

We have seen how representation by generalized cylinders may model many complex curved objects, we have seen how a practical description program can isolate the parts of objects and describe them in terms of generalized cylinders. We have also seen some examples not well handled by our representation, and some examples where our description program breaks down.

Regardless of its shortcomings, we feel that the performance we have demonstrated proves the feasibility of the method. With improvements along the lines of the suggestions in Chapter 6, it has the potential as the nucleus of a powerful vision system.

7.1 ADVANTAGES OF ACTIVE RANGING

The presence and availability of depth information to a description program increases its capability immensely. Our experimental work on description of curved objects would not be possible without depth information from which to form models. Shirai and Suwa [Shirai] report they are able to describe the class of right prisms using a ranging system essentially similar to ours, and some rather simple plane-recognizing routines.

Incorporation of depth information into the Stanford Hand-Eye system could speed low-level processing and provide additional information for higher level routines. Work is presently under way to provide a one-point-at-a-time ranging capability, using either the laser triangulation system or stereo correlation, to disambiguate between alternative representations at the highest level. Figure 7.1 shows what might be accomplished if depth information were used at a lower level. To generate this figure, the second-order curve fitting was disabled in the curve fitter, and only straight lines were fit. The recursive segmenter locates the corners of the objects very well. Figure 7.2 plots only the end points of each straight line segment. This compares rather favorably with output from conventional two-dimensional edge followers.

A drawback to the present system of ranging is the long time it takes to scan a scene. For each laser scan the TV camera must be read and the line detected, and it may take anywhere from 20 to several hundred scans to cover an entire scene. But more sophisticated hardware can reduce the data acquisition time by one or two orders of magnitude. A coded grid of the sort mentioned in the last paragraph of Section 3.2 would require only

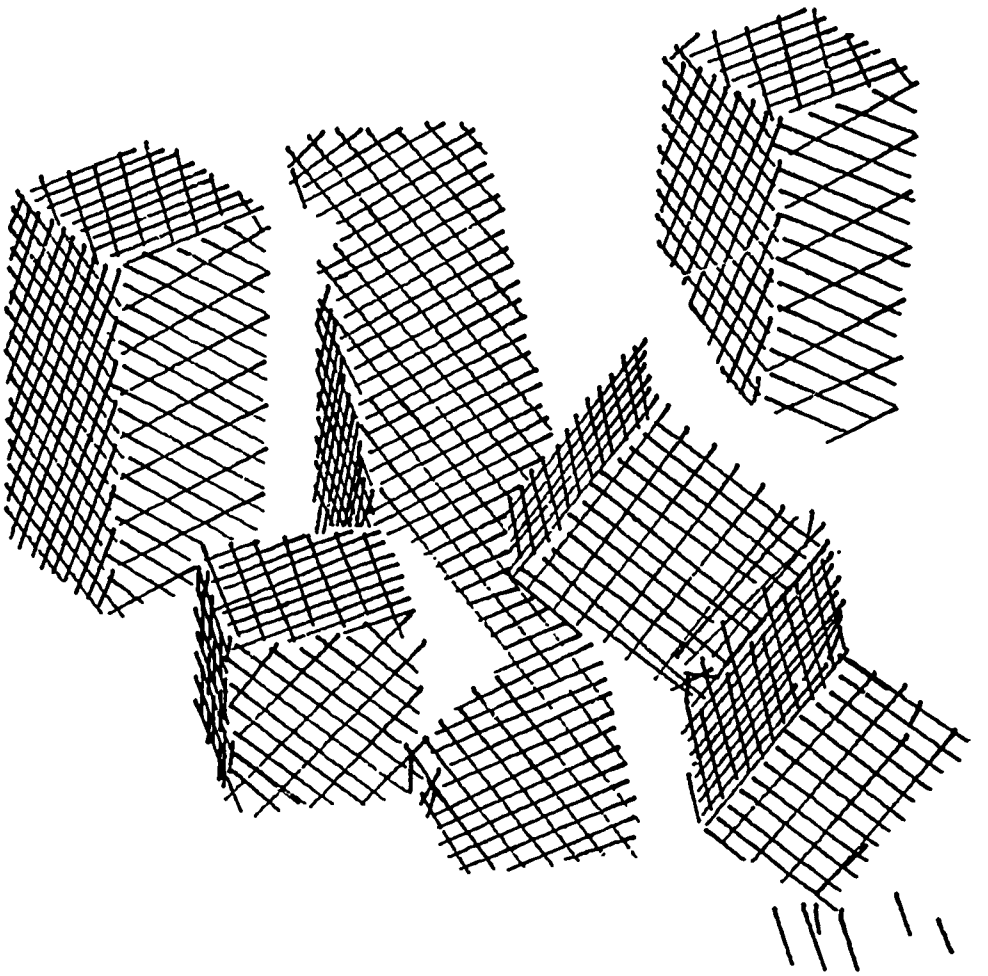


Figure 7.1
Plane-Faced Solids

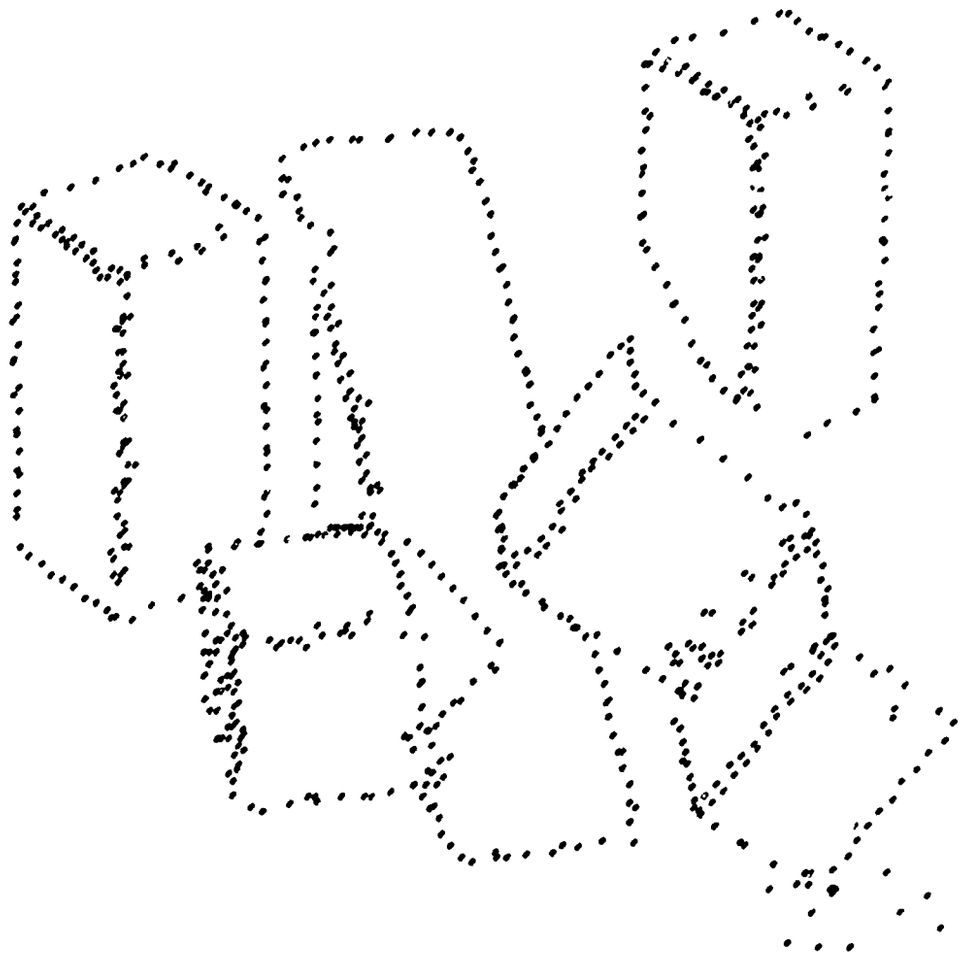


Figure 7.2
Edges Obtained from Three-Dimensional Data

ore exposure of the TV camera to view a scene, but would require some fairly sophisticated processing to decode the grid.

In the future, direct ranging schemes such as time-of flight measurement may provide a full depth grid in as little time as it takes to read the TV camera. These would also eliminate the drawback of the present method, that planes and surfaces visible to the camera but occluded from the laser cannot be detected. Another possible technique might be stereo television with hardware correlation of images.

We do not propose the immediate application of any presently available ranging devices to replace existing two-dimensional techniques, but we predict that active ranging will be utilized more and more in vision research in the near future. Active ranging will speed the day when visual capabilities are feasible in practical robots. The additional capabilities provided by depth information make these predictions virtual certainties.

7.2 WHERE DO WE GO FROM HERE?

We are at the verge of a new epoch in the science of computer vision. The recent interest in curved objects is only a part of this new era. There are many plans for research and development of practical robots for industrial and extraterrestrial applications, and vision is sure to be a major contributer to the new technology.

To exploit the full capabilities of vision in general-purpose intelligent machines, much research must be done in the areas of image acquisition, object description, and representation theory.

The advantages of active ranging have been discussed in the previous Section, as well as some possible candidates for fast and accurate three-dimensional image acquisition.

Methods of description from actual data of primitives and objects can benefit from innovation. Axes are frequently suggested by the direction of minimum curvature of visible surfaces, but sensitivity of this method to noise in the data must be considered. Outlines of objects may be constructed from the depth discontinuities in a scene (see Section 6.3). We have made no attempt to utilize these outlines in detecting cylinders, except in the preliminary axis determination. These outlines are of obvious value in guiding a primitive detector or higher-level object recognizer. And very little work has yet been done in

detailed strategies for a hypothesis-verify paradigm, or for recognition.

The representation we propose and use throughout this research has been selected and developed with the idea of general utility in mind. The representation we believe to be useful in recognizing and dealing with everyday objects. But our ideas of utility are based mainly on mental exercises. We asked, "If we were to write a program to recognize hand tools, what kinds of information would be useful?" The final test of our efforts will be the application of our representation and methods in a practical robot. We have formed a hypothesis about how to deal with the visual problems to be encountered by robots; verification of this hypothesis is yet to be performed. When the loop is closed, a truly practical system of representation and description can evolve.

The first practical robots will deal with a limited universe. The representations they use will probably model adequately the objects with which they are to deal, but not have wide applicability outside their limited areas. Nonetheless, they will provide valuable experience. A general representation theory must deal with all the objects handled by these special purpose systems.

There is a need, too, to understand the requirements of a representation in the related areas of graphics and man-machine communications. Progress has been made independently in these areas, but as the capabilities of intelligent machines improve, there will be a need to unify the representations they use. Experience, again, is the key.

The next decade will see great strides in the utilization of intelligent machines. We are proud to have been a part of their development.

APPENDIX A: ERROR ANALYSIS OF RANGING SYSTEM

Two assumptions were made in Section 3.4 which will affect the accuracy of ranging. These were:

1. That the area on the surface of the rotating mirror through which all laser planes pass may be approximated by a point, and

2. That the undiverged laser beam is reflected from the rotating mirror at right angles to the axis of rotation of the mirror.

The validity of these assumptions will depend on the initial alignment of the laser deflection assembly with the laser beam.

For convenience in exposition, let us denote the laser ray which passes through the cylindrical lens in a straight line the "principal ray" of the laser plane. The principal ray follows the same path the laser beam would follow if the cylindrical lens were not present.

If the principal ray strikes the rotating mirror at its center of rotation, then all planes of light must pass through the point where it strikes. If the principal ray misses the center of rotation, then the size of the area through which all planes of light must pass is dependent on how close to the mirror's axis the principal ray strikes, on the angle the mirror makes with the beam, and on the angle through which the mirror will rotate. Figure A.1 illustrates the geometry of the situation.

Call the distance from the principal ray to the center of the mirror A, and the angle made by the principal ray and the mirror (in its "centered" position) θ. Suppose the mirror normal rotates over an angle of Φ, and let us compute the apparent size of of the area in question, the distance B measured parallel to the reflected principle ray in the "centered" position.

The misalignment A is certainly not more than 0.2 inches; let us call it 0.2, and see how much error it introduces. Typical values for θ and Φ are 60 degrees and 5 degrees, respectively. Simple trigonometry taking into account that the angle of reflection of the principal ray is equal to the angle of incidence) gives a value of 2.215 inch for the distance B.

APPENDIX A: ERROR ANALYSIS OF RANGING SYSTEM

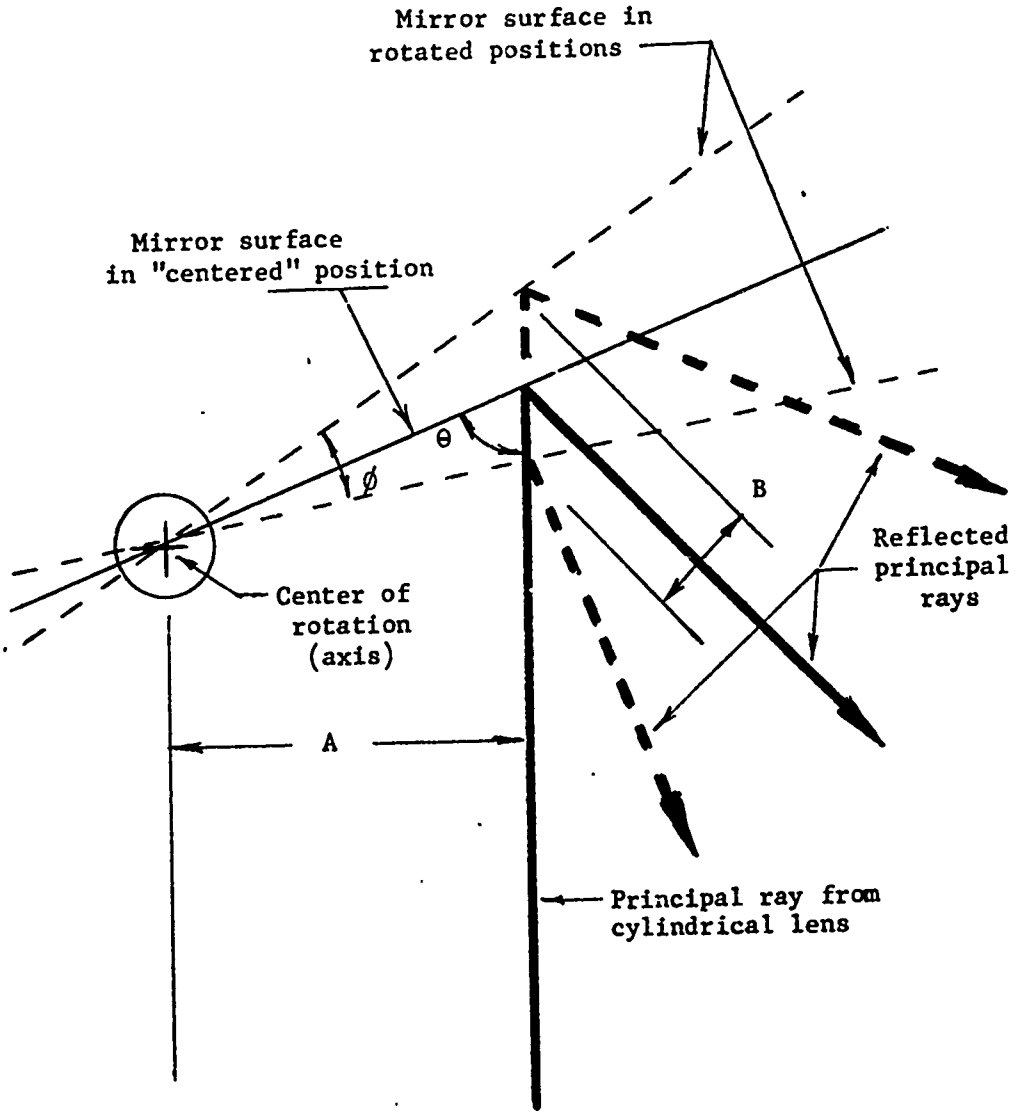


Figure A.1
Error in Laser Center Position

APPENDIX A: ERROR ANALYSIS OF RANGING SYSTEM

The plane of light passes through a point which may be as far as $s/2$, or 0.0075 inches away from where the calibration information thinks it is. Figure A.2 shows how to calculate the range error from this uncertainty. X marks a point on the laser plane whose depth is known from the calibration pattern. We wish to know the depth of some other point in the scene located a distance D away from X. The distance from the laser center to the point X is typically about 24 inches. The scenes we scan are usually not more than 6 inches deep, so the distance D is not more than three inches. Solution of similar triangles gives a distance of 0.001 inch for the error, C, in the position of the plane. This must be divided by the tangent of the 30 degree angle between the principal ray and the line to the camera, giving an error of about 0.002 inches.

The error caused by the reflected principal ray not being perpendicular to the axis of rotation of the mirror is somewhat larger. With the magnitude of the angles involved, the predominant effect of non-perpendicularity will be a deviation from straightness of the locus of the principal ray across the table as the mirror rotates. The initial alignment procedure guarantees that the laser beam will be within 0.3 degrees of being perpendicular to the mirror surface. (The cylindrical lens and focussing lens are removed, then the mirror is rotated and the deflection assembly positioned so that the beam is reflected back into the laser.) The mounting of the mirror to the motor shaft is to usual mechanical tolerances, and should not be oriented at an angle of more than 0.5 degrees to its axis. We shall assume an angular deviation of one degree.

The principal ray will sweep out a cone in space as the mirror rotates. This cone will intersect the tabletop in a hyperbola, as shown in Figure A.3. Over a baseline of 12 inches, the deviation from linearity will be 0.026 inches. Since the laser planes are oriented at $\pm$ 45 degrees to the baseline, the error in their position will be 0.707 times the deviation from linearity, or 0.019 inches. This we divide by the tangent of the angle between the laser ray and the line to the camera, to obtain an error in depth of 0.037 inches.

There are some other known sources of error in the calibration. Errors in measuring the intersections of the laser scans on the table top in the calibration pattern are about 0.020 inch. This, too, must be divided by the tangent of the angle between the laser and the camera, to yield a depth error of 0.010 inch. Other errors in the camera calibration are probably less than one raster unit, and may be included in the 0.035 to 0.070

APPENDIX A: ERROR ANALYSIS OF RANGING SYSTEM

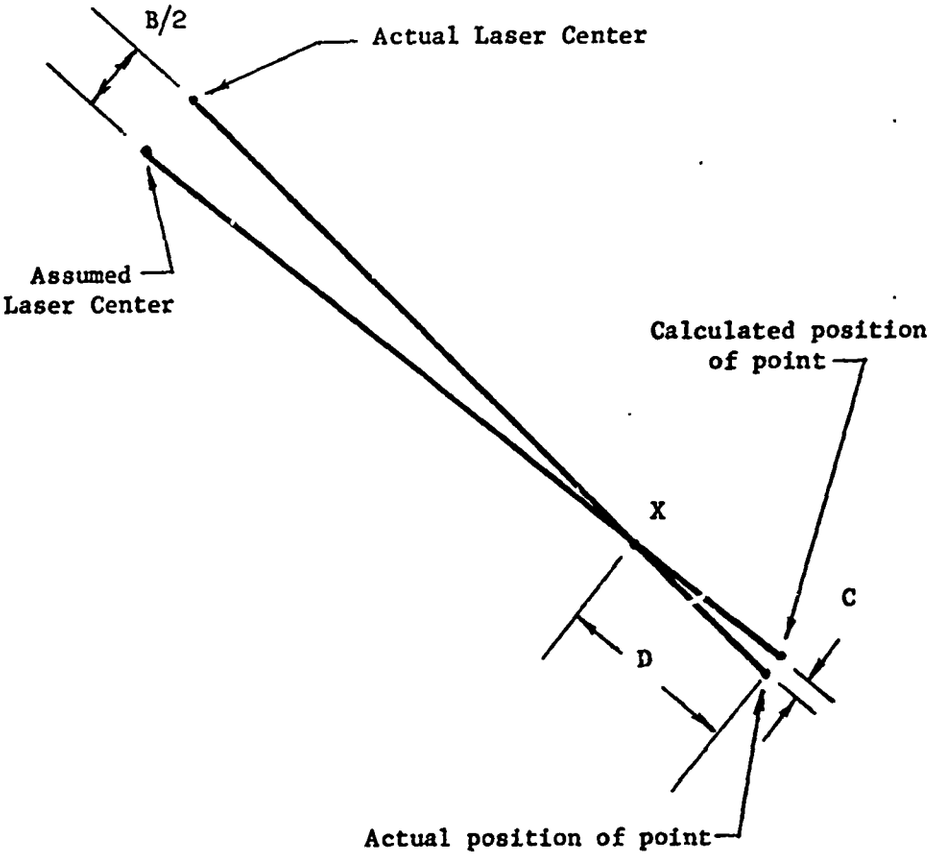


Figure A.2
Error In Depth Due to Motion of Laser Center

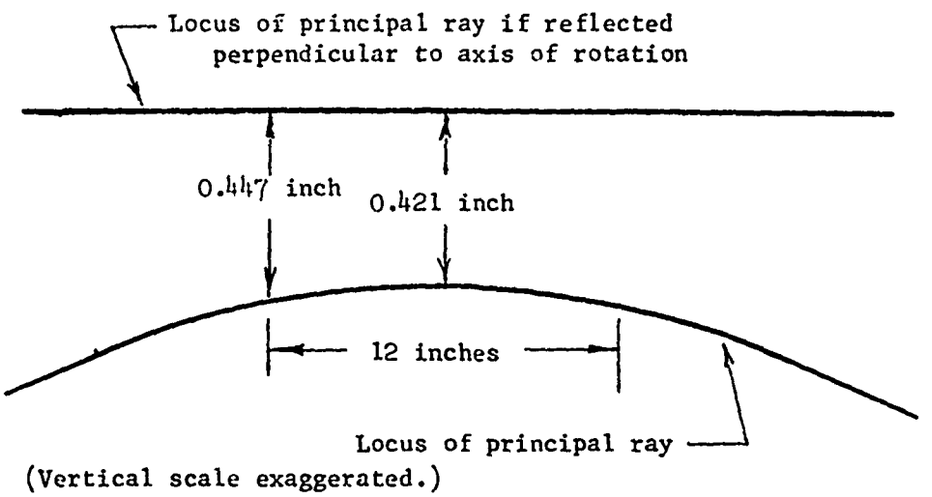


Figure A.3
Error due to Non-Perpendicularity of Laser Ray

inch figure given in Section 3.2 as the error due to resolution in the TV camera.

Each of the errors mentioned here are maximum errors. Adding them together gives a total maximum absolute error in range of 0.294 to 0.129 inch, or about a tenth of an inch. But to obtain the average expected error of any depth measurement, the individual errors must be divided by the square root of two, and they must be added in root-sum-square fashion. By this reckoning, the average absolute range error is 0.045 inch to 0.056 inch, or about 1/20 inch.

APPENDIX B: CURVE FITTING

We represent two dimensional curves by equations of the form

$$G(x,y) = \sum_J \sum_K C_{JK} x^J y^K = 0. \quad \text{[Equation B.1]}$$

Curve fitting is the process of determining a set of values for the coefficients C_{JK} which minimize the mean square error,

$$E = \sum_i (G(x_i, y_i))^2 / N, \quad \text{[Equation B.2]}$$

over some set of data points (x_i, y_i), $1 \le i \le N$, under certain constraints.

That constraints are required should be obvious from the form of the function $G(x,y)$. If each of the C_{IJ}'s are zero, $G(x,y)$ will be everywhere identically zero, and the minimum E will be also zero.

The constraint we have chosen to use is the "average gradient" constraint described in Section B.1. One of the principal advantages of this method is that the result of the minimization process is independent of the coordinate system, a result which we shall show in Section B.4. We know of no other constraint or set of constraints on the coefficients which possess this property.

A problem with our method of curve fitting is a tendency to produce curves which "curl in" at the ends. An analysis of this phenomenon is contained in Section B.5.

B.1 FITTING OF GENERAL SECOND-ORDER CURVES

A general second-order curve is represented by the equation

$$G(x,y) = A x^2 + B x y + C y^2 + D x + E y + F = 0 \quad \text{[Equation B.3]}$$

The solutions to this equation for fixed coefficients A through F are the conic sections. Finding the "best fit" of this equation to a set of data points (x_i, y_i) is equivalent to minimizing the error measure $E = \sum_i G(x_i, y_i)^2$ over the coefficients A, B, C, D, E, and F.

The error measure E may be concisely represented in matrix representation. Define the vectors V and X as

$$X = \begin{matrix} x^2 \\ xy \\ y^2 \\ x \\ y \\ 1 \end{matrix} \quad \text{and} \quad V = \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} \qquad \text{[Equation B.4] and [Equation B.5]}$$

Then

$$G(x,y) = V^T X = X^T V \qquad \text{[Equation B.6]}$$

$$G^2 = V^T X X^T V \qquad \text{[Equation B.7]}$$

$$E = \sum G^2 = \sum V^T X X^T V = V^T \sum (X X^T) V \qquad \text{[Equation B.8]}$$

$$E = V^T P V \qquad \text{[Equation B.9]}$$

where $P = X X^T$ is a matrix of sums of powers of x and y.

The coefficients which specify the "best fit" are contained in the vector V which minimizes $V^T P V$. Clearly a constraint is needed. For if V is the zero vector then G(x,y) will be everywhere identically zero.

Before choosing a constraint, let us ask what it is we really wish to minimize. Certainly there are constraints which will yield a low value of the error measure, which we could not accept intuitively as a good fit. The intuitive judgment of goodness of fit depends on the distance of the data points from the line which approximates them. This suggests we search for a constraint which will cause the error function G(x,y) to be proportional to the distance from the point (x_i, y_i) to the line G(x,y)=0.

To a first approximation, the distance from the point to the line is the value of G(x,y) divided by the gradient of G, ∇G, in the neighborhood of the point. Ideally, we should wish to minimize $\sum (G / |\nabla G|)^2$. However we can approximate this by constraining the average or root-mean-square gradient of G to be unity over the set of data points, and minimizing $\sum G^2$ under that constraint.

The magnitude of the gradient of G, ∇G, is the square root of $(V^T X_x)^2 + (V^T X_y)^2$, where

$$X_x = \begin{matrix} 2x \\ y \\ 0 \\ 1 \\ 3 \\ 0 \end{matrix} \quad \text{and} \quad X_y = \begin{matrix} 0 \\ x \\ 2y \\ 0 \\ 1 \\ 0 \end{matrix}$$

[Equation B.10] and [Equation B.11]

are the derivatives of X with respect to x and y, respectively. Then

$$(\nabla G)^2 = (V^T X_x)^2 + (V^T X_y)^2 = V^T (X_x X_x^T + X_y X_y^T) V$$

[Equation B.12]

$$\Sigma(\nabla G)^2 = V^T \Sigma (X_x X_x^T + X_y X_y^T) V = V^T Q V,$$

[Equation B.13]

where $Q = (X_x X_x^T + X_y X_y^T)$ is another matrix of sums and powers of x and y. Requiring the root-mean-square gradient of G to be unity is equivalent to requiring that $V^T Q V = N$, where N is the number of data points.

We wish to minimize the matrix product $V^T P V$, under the constraint that $V^T Q V = N$. It is well known from LaGrange

multiplier theory [Courant] that the constrained minimum is a solution to the generalized eigenvalue equation

$$P\,V = \lambda\,Q\,V \qquad \text{[Equation B.14]}$$

for some value of λ.

The solution of this equation is described in Section B.2. Each of the five eigenvectors found are normalized to make $V^T Q V = N$. The mean square error of each solution is equal to its corresponding eigenvalue, since

$$\frac{\sum G^2}{N} = \frac{V^T P V}{N} = \frac{\lambda V^T Q V}{N} = \lambda \qquad \text{[Equation B.15]}$$

B.2 SOLUTION OF THE GENERALIZED EIGENVALUE EQUATION

To implement the method of Appendix B.1 requires solution of the generalized eigenvalue equation

$$P\,V = \lambda\,Q\,V\,, \qquad \text{[Equation B.16]}$$

However, the matrix Q is singular and the matrix P is nearly singular: the closer the data points are to lying on a conic section curve, the closer P is to being singular. We are indebted to Mr. Richard Underwood for supplying the following numerical method for solving the equation.

The last row and the last column of the matrix Q are zero. The matrix Q may be represented by the partitioned matrix

$$Q = \begin{array}{c|c} Q^{*} & \emptyset \\ \hline \emptyset & \emptyset \end{array} , \qquad \text{[Equation B.17]}$$

We may usually expect the five-by-five matrix Q^{*} to be positive definite. A Cholesky decomposition will factor Q into a lower diagonal matrix L, and its transpose, $\bar{L}$, so that

$$Q = L L^{T} = \begin{array}{c|c} L^{*} & \emptyset \\ \hline \emptyset & \emptyset \end{array} \begin{array}{c|c} L^{*T} & \emptyset \\ \hline \emptyset & \emptyset \end{array} \qquad \text{[Equation B.18]}$$

If we let $\bar{L}$ represent the augmented matrix

$$\bar{L} = \begin{array}{c|c} L^{*} & \emptyset \\ \hline \emptyset & 1 \end{array} , \qquad \text{[Equation B.19]}$$

then we have the result

$$\bar{L}^{-1} Q \bar{L}^{-T} = \begin{array}{c|c} I & \emptyset \\ \hline \emptyset & \emptyset \end{array} , \qquad \text{[Equation B.20]}$$

where I denotes the five-by-five unit matrix. (The notation $\underline{L}^{-T}$ denotes the transpose of $\underline{L}^{-1}$).

The original generalized eigenvalue equation, Equation B.16, may be transformed into

$$\underline{L}^{-1}\, P\, \underline{L}^{-T}\ \underline{L}^{T}\, V = \underline{L}^{-1}\, Q\, \underline{L}^{-T}\ \underline{L}^{T}\, \lambda\, V. \qquad \text{[Equation B.21]}$$

Applying the substitution

$$C = \underline{L}^{-1}\, P\, \underline{L}^{-T} = \begin{array}{c|c} C' & U \\ \hline T & \\ U & \alpha \end{array} \qquad \text{[Equation B.22]}$$

and letting Y be the partitioned column vector

$$Y = \underline{L}^{T}\, V = \begin{array}{c} Z \\ \hline W \end{array} \qquad \text{[Equation B.23]}$$

yields the representation

$$\begin{array}{c|c|c} C' & U & Z \\ \hline T & & \\ U & \alpha & W \end{array} = \lambda \begin{array}{c|c|c} I & \emptyset & Z \\ \hline & & \\ \emptyset & \emptyset & W \end{array} . \qquad \text{[Equation B.24]}$$

This may be factored into the two eigenvalue equations

$$C'\, Z + V\, W = \lambda\, Z \qquad \text{[Equation B.25]}$$

and

$$V^T Z + \alpha W = 0 \quad , \qquad \text{[Equation B.26]}$$

Solution of Equation B.26 yields

$$W = - (V^T Z) / \alpha \qquad \text{[Equation B.27]}$$

which when substituted into Equation B.25 gives the form

$$(C' - V V^T / \alpha) Z = \lambda Z \quad , \qquad \text{[Equation B.28]}$$

Equation B.28 may be solved by usual eigenvalue methods, such as the Q-R algorithm [Isaacson]. The five original eigenvectors of Equation B.16 may be recovered by applying Equation B.27 and Equation B.23.

B.3 FITTING OF STRAIGHT LINES

The fitting of straight lines follows the same procedure as the fitting of curved lines, Section B.1.

A straight line is represented by the equation

$$G(x,y) = A x + B y + C = 0, \qquad \text{[Equation B.29]}$$

Defining

$$V = \begin{matrix} A \\ B \\ C \end{matrix} \quad \text{and} \quad X = \begin{matrix} x \\ y \\ 1 \end{matrix} \qquad \begin{matrix} \text{[Equation B.30]} \\ \text{and} \\ \text{[Equation B.31]} \end{matrix}$$

gives

$$G(x,y) = V^T X = X^T V \qquad \text{[Equation B.32]}$$

$$\Sigma G^2 = \Sigma V^T X X^T V = V^T \Sigma (X X^T) V = V^T P V,$$

[Equation 8.33]

where

$$P = \Sigma X X^T = \begin{matrix} \Sigma x^2 & \Sigma xy & \Sigma x \\ \Sigma xy & \Sigma y^2 & \Sigma y \\ \Sigma x & \Sigma y & N \end{matrix}$$

[Equation 8.34]

The gradient of G, ∇G, is the square root of $A^2 + B^2$ for all x and y. Thus

$$(\nabla G)^2 = V^T \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix} V = V^T Q V$$

[Equation 8.35]

The constrained minimum ΣG^2 subject to the constraint $\nabla G = 1$ is a solution to the generalized eigenvalue equation

$$P V = \lambda Q V$$

[Equation 8.36]

for some value of λ. Using the method of Section 8.2, this equation may be solved symbolically with the following results.

Let $$r = \sum x^2 - \frac{(\sum x)^2}{N}$$ [Equation B,37]

$$s = \sum xy - \frac{\sum x \sum y}{N}$$ [Equation B,38]

and $$t = \sum y^2 - \frac{(\sum x)^2}{N},$$ [Equation B,39]

Then the eigenvalues become

$$\lambda = (r + t \pm [(r - t)^2 + 4 s^2]^{1/2}) / 2,$$ [Equation B,40]

And the eigenvectors are given by

$$V = \begin{matrix} -s \\ r - \lambda \end{matrix} \qquad s \sum x - (r - \lambda) \sum y$$ [Equation B,41]

B,4 ROTATIONAL, TRANSLATIONAL, AND SCALING INVARIANCE OF EIGENVECTOR SOLUTION

In this Section we shall show that the results of curve fitting by the method of Section B,1 are independent of the coordinate system in which fitting takes place, provided the coordinate system is an orthogonal one.

To show why this property is desirable, let us consider a rather obvious example of curve fitting under a constraint other than the average gradient. A representation of general second order curves commonly used in curve fitting is

$$G(x,y) = A x^2 + B x y + C y^2 + D x + E y + 1 = 0,$$ [Equation B,42]

This is equivalent to Equation 1 of Section B,1 with the coefficient F equal to unity. The minimization of this function

is frequently treated in college calculus courses, and involves solution of a set of five simultaneous linear equations. We observe that G(0,0) is identically 1; hence the curve cannot pass through the origin of coordinates! If some of the data points happen to lie near the origin, the resulting curve will be grossly different from what might be expected otherwise.

Using the notation of Section B.1, suppose that for a set of data points there exists a vector V and a scalar λ such that

$$\sum X X^T V = \lambda \sum (X_x X_x^T + X_y X_y^T) V,$$

[Equation B.43]

Let there be some u-v coordinate system related to the x-y coordinate system by the transformation [h], such that

$$\begin{matrix} u \\ v \\ 1 \end{matrix} = [h] \begin{matrix} x \\ y \\ 1 \end{matrix} = \begin{matrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{matrix} \begin{matrix} x \\ y \\ 1 \end{matrix},$$

[Equation B.44]

We may derive from the above that

$$U = \begin{matrix} u\,u \\ u\,v \\ v\,v \\ u \\ v \\ 1 \end{matrix} = \begin{matrix} aa & 2ab & bb & 2ac & 2bc & cc \\ ad & ae+bd & be & af+cd & bf+ce & cf \\ dd & 2de & ee & 2df & 2ef & ff \\ 0 & 0 & 0 & a & b & c \\ 0 & 0 & 0 & d & e & f \\ 0 & 0 & 0 & 0 & 0 & 1 \end{matrix} * \begin{matrix} x\,x \\ x\,y \\ y\,y \\ x \\ y \\ 1 \end{matrix}$$

[Equation B.45]

$$U = H X$$

[Equation B.46]

where H is an expanded transformation matrix relating U and X. Let U_u, and U_v be the derivatives of U with respect to u and v, respectively. If we can show that there exists a vector V' and a scalar λ' such that for all x and y,

$$V^T X = V'^T U, \qquad \text{[Equation 8.47]}$$

and such that

$$\Sigma U U^T V' = \lambda' \Sigma (U_u U_u^T + U_v U_v^T) V', \qquad \text{[Equation 8.48]}$$

then invariance with coordinate system will have been proven.

The derivatives of U with respect to x and y are evaluated as follows.

$$U_x = \begin{matrix} 2aax + 2aby + 2ac \\ 2adx + (ae+bd)y + af + cd \\ 2ddx + 2dey + 2df \end{matrix} \begin{matrix} a \\ d \\ \emptyset \end{matrix}$$

$$= \begin{matrix} 2a\,(ax+by+c) \\ a\,(dx+ey+f) + d\,(ax+by+c) \\ 2d\,(dx+ey+f) \end{matrix} \begin{matrix} a \\ d \\ \emptyset \end{matrix} = \begin{matrix} 2au \\ av + du \\ 2dv \end{matrix} \begin{matrix} a \\ d \\ \emptyset \end{matrix}$$

$$= a U_u + d U_v \qquad \text{[Equation 8.49]}$$

Similarly,

$$U_y = \begin{matrix} 2abx + 2bby + 2bc \\ (ae+bd)x + 2bey + bf + ce \\ 2dex + 2eey + 2ef \end{matrix} \begin{matrix} b \\ e \\ \emptyset \end{matrix} = b U_u + e U_v, \qquad \text{[Equation 8.50]}$$

In addition, we shall need the result that for [h] to be an orthogonal transformation requires that

$$a^2 + b^2 = d^2 + e^2 \qquad \text{[Equation B.51]}$$

and

$$ad + be = 0. \qquad \text{[Equation B.52]}$$

We are now prepared to prove the main result by deriving Equations 3 and 4. The only V' which will satisfy Equation B.47 is

$$V' = H^{-T} V \qquad \text{[Equation B.53]}$$

or

$$V = H^{T} V' \qquad \text{[Equation B.54]}$$

Substituting the above in Equation B.43 gives

$$\sum X X^{T} H^{T} V' + \lambda \sum (X_x X_x^{T} H^{T} V' + X_y X_y^{T} H^{T} V') = 0. \qquad \text{[Equation B.55]}$$

We premultiply by H to get

$$\sum H X X^{T} H^{T} V' + \lambda \sum (H X_x X_x^{T} H^{T} V' + H X_y X_y^{T} H^{T} V') = 0. \qquad \text{[Equation B.56]}$$

Differentiating Equation B.46 with respect to x, or y yields

$$U_x = H X_x \quad \text{and} \quad U_y = H X_y \qquad \text{[Equations B.57 and B.58]}$$

which we use to write

$$\Sigma\, U\, U^T\, V' + \lambda \Sigma\, (U_x\, U_x^T\, V' + U_y\, U_y^T\, V') = \emptyset$$

[Equation B.59]

or $$\Sigma\, U\, U^T\, V' + \lambda \Sigma\, (U_x\, U_x^T + U_y\, U_y^T)\, V' = \emptyset,$$

[Equation B.60]

Now, by Equations B.49, B.50, B.51, and B.52, we may write

$$\begin{aligned} U_x\, U_x^T + U_y\, U_y^T &= a^2\, U_u\, U_u^T + a\,d\, U_u\, U_v^T + a\,d\, U_v\, U_u^T \\ &\quad + d^2\, U_v\, U_v^T + b^2\, U_u\, U_u^T + b\,e\, U_u\, U_v^T \\ &\quad + b\,e\, U_v\, U_u^T + e^2\, U_v\, U_v^T \\ &= (a^2 + b^2)\, U_u\, U_u^T + (d^2 + e^2)\, U_v\, U_v^T \\ &\quad + (a\,d + b\,e)\, (U_u\, U_v^T + U_v\, U_u^T) \\ &= (a^2 + b^2)\, (U_u\, U_u^T + U_v\, U_v^T), \end{aligned}$$

[Equation B.61]

Substituting the above into Equation B.60 gives

$$\Sigma\, U\, U^T\, V' + \lambda\, (a^2 + b^2)\, \Sigma\, (U_u\, U_u^T + U_v\, U_v^T)\, V' = \emptyset,$$

[Equation B.62]

which is Equation B.48, with $\lambda' = \lambda\, (a^2 + b^2)$.

B.5 THE CURVATURE PROBLEM

There is a tendency for our curve fitter to fit to the data curves which are very much more flattened than we would like to see. Figure 4.13 shows some examples of this type of "error", especially the upper right, middle left, and lower left curves in the illustration. The curve fitting subroutines check for this kind of error, and reject curves where the elongation is obviously too great. But the tendency to fit flattened ellipses affects all curves, and introduces an undesirable amount of systematic error into the fitted data.

The problem is accentuated by the fact that we are dealing with only portions of ellipses (or hyperbolas). With a given short arc, a large number of curves may be construed to fit. When dealing with complete ellipses, the principal dimensions of the curves necessary to fit them are immediately apparent, and only minor variations in the fit could be considered reasonable.

We shall show that this tendency arises from the fact that, all other things being equal, the error of a distribution of points about a curve depends inversely on the magnitude of the second derivative of the error function. That is, a curve whose gradient varies rapidly tends to "fit" better, in the least squares sense, than one with a constant gradient.

The problem is not limited to fitting with the average gradient constraint. Lyle Smith [Smith] noted the same phenomenon, fitting curves with the constant term, F, of Equation B.3 constrained to unity. We believe, but have been unable to prove, that no linear or quadratic constraint on the coefficients of Equation B.3 can eliminate the problem.

An illustration will make clear the dependence of the average error measure on the derivative of the (magnitude of the) gradient. Suppose that in the x-y plane there exist n points uniformly distributed near the y-axis between the limits $x = -1$ and $x = 1$, as shown in Figure B.1. A one-dimensional error function

$$F(x) = A x^2 + B x + C \qquad \text{[Equation B.63]}$$

gives the error of each point. The average error measure

$$G = \sum_{1}^{n} (F(x_1))^2 / n \qquad \text{[Equation B.64]}$$

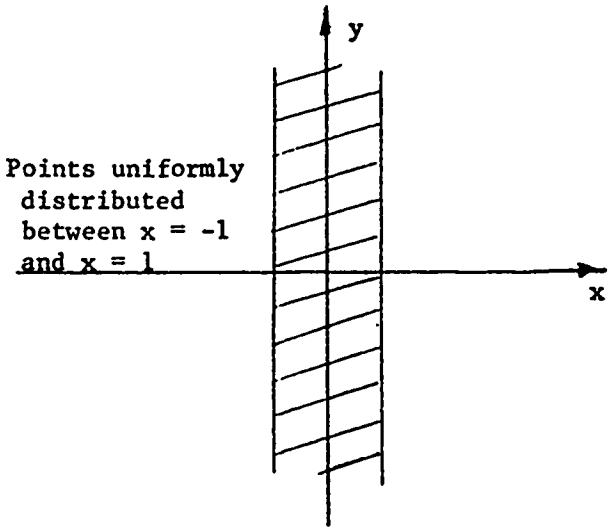


Figure 8.1
A Distribution of Points in the x-y Plane

is to be minimized over the distribution of points. We may approximate the error measure F by an integral:

$$\bar{G} = \int_{-1}^{1} (A x^2 + B x + C)^2 \, dx. \qquad \text{[Equation 8.65]}$$

The average gradient constraint requires that

$$\frac{1}{2} \int_{-1}^{1} \left(\frac{dF}{dx} \right)^2 dx = 1 \qquad \text{[Equation 8.66]}$$

or

$$\int_{-1}^{1} (2 A x + B)^2 \, dx = 8/3 A^2 + 2 B^2 = 2. \qquad \text{[Equation 8.67]}$$

The parameter A must not be greater than the square root of 3/4 in order that this constraint be satisfiable.

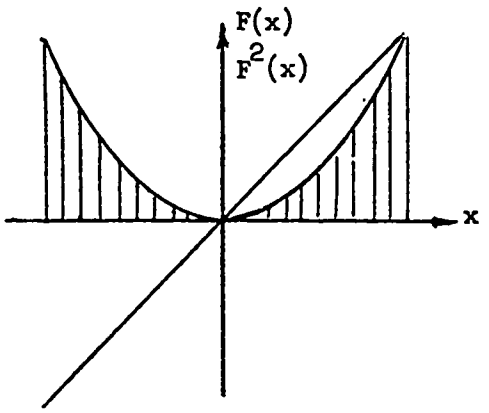
The second derivative of F(x), (which in the two dimensional case would be the derivative of the magnitude of the gradient of F), is proportional to A. For illustrative purposes, we may choose arbitrary values for A, subject to the limitation above, solve for B in Equation B.67, then find the value for C which minimizes the error measure $\bar{G}$.

Figure B.2 plots F(x) and $F(x)^2$ against x for three different values of A. (This plot would require three dimensions in the two dimensional case.) The integrals of the shaded areas under the curves for $F(x)^2$ are $\bar{G}$, the error measure. The minimum error occurs when A is equal to the square root of 3/4.

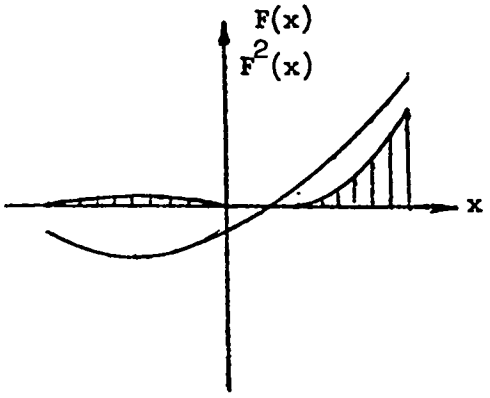
Elongated ellipses, and hyperbolas whose asymptotes cross at narrow angles, are characterized by a high derivative of the magnitude of the gradient of their defining function. The curve fitter chooses these flattened curves over the more intuitive curves we would prefer.

A possible solution to the problem involves a weighted least squares approach and an iterative method. Lyle Smith used a method in which, after solving the least squares equation to find an ellipse, he would weight each point by the curvature of the ellipse near that point. After several iterations, the method converged to the intuitive solution. But our attempts to apply this method to our average gradient solution failed. It was difficult to distinguish among the five eigenvectors to determine which one to use in determining the weights. Perhaps a smaller weighting, so that the solutions varied slowly with each iteration, might have given acceptable results, but this was not tried.

A = 0
F(x) = x
G = 0.333



A = $\sqrt{3}/4$
F(x) = $\sqrt{3}/4(x^2+x-1/30)$
G = 0.158



A = $\sqrt{3}/2$
F(x) = $\sqrt{3}/2(x^2-1/2)$
G = 0.087

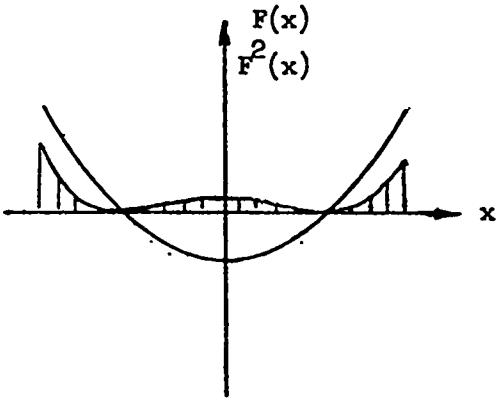


Figure 8.2
Error Measure with Different Curvatures Imposed

8.6 ALTERNATIVES TO GENERAL SECOND-ORDER CURVES

Curve fitting by general second-order curves leaves much to be desired. Solution of the least squares equations requires much computation. Choosing among the three solutions returned is tricky. A practical solution to the end-curling problem has not been found. The parameters of the curve are not easy to manipulate. We discuss in this Section some alternatives to general second-order curves which minimize some or all of these problems.

We are committed to general second order curves mainly for historical reasons. The curve fitting routines were originally written for an attempt to detect cylinders, cones, and spheres in conventional TV images processed by an edge finder, much in the same way cubes are detected in the present Stanford Hand-Eye system [Feldman]. In such an application, it was important that ellipses be represented simply. In the present application, description of three-dimensional curved objects, what we require mainly is an efficient smoothing operation, and a representation capable of carrying curvature information. But at this point in time, the parameters and data structures of general second order curves are so firmly embedded in our programs that it would take several weeks to convert to another representation.

Nevertheless, we suggest some alternatives.

The easiest type of function to compute would be quadratic polynomials. For each segment, we draw a baseline between the endpoints, and rotate the segment so that the transformed baseline is the x-axis of the new coordinate system. This method would tend to isolate segments of uniform curvature. Because a segment may have uniform curvature when viewed perpendicularly in 3-space, but not appear uniform when viewed in perspective from the TV camera, it would be advisable when using this method to transform the points to three dimensions before curve fitting.

The data structure for quadratic polynomials would need to contain only the coordinates of the two end points, and the three coefficients A, B, and C of the function $y = A x^2 + B x + C$. Since the coordinates of the end points uniquely determine the rotation of the points, there would be no need to separately include it.

A version of the curve fitting program using quadratic polynomials has been implemented, and seems to give somewhat more

consistent results than fitting by general second-order curves. Figure B.3 shows the torus as it was with quadratic polynomials. Compare this with Figure 5.15. Fitting is in TV raster coordinates in this version. Some further improvement might possibly be expected were conversion to three dimensions to take place before curve fitting.

Quadratic splines are very much like quadratic polynomials, with an additional stipulation that the end points of the functions representing each segment must coincide.

A spline is a function defined on an interval divided up into a finite number of subintervals. The end points and the points where two subintervals join are called nodes. For an n-th order spline, the function on each subinterval is an n-th order polynomial. Usually the function is constrained to be continuous, and continuous in its n-1 derivatives, at its nodes, but other constraints are sometimes applied. The spline functions which we suggest are second-order, and continuous in zero-th derivative (the function itself must be continuous.)

If we draw a baseline between the end points of the line and rotate the curve so that the transformed baseline is the x-axis of the new coordinate system, we may fit the line with such a spline. For a given placement of the nodes, the function $F_{min}(x)$ which minimizes the mean square error $\Sigma(F(x)-x_i)^2$ may be found as the solution to a (2n+1)th order system of linear equations, where n is the number of subintervals.

Instead of recursive segmentation, we need a procedure to systematically introduce nodes until an acceptable fit is obtained. Introducing new nodes at the points where the pointwise error is greatest, and perturbing the nodes to improve the overall fit, in a manner similar to the method of DeBoor and Rice [DeBoor], may give good results, but many details remain to be worked out. Some care must be taken not to let nodes lie too close together, to prevent obtaining functions which oscillate wildly. Some means of distinguishing between straight lines and curves should be looked into, since straight lines may not be fitted separately from quadratics.

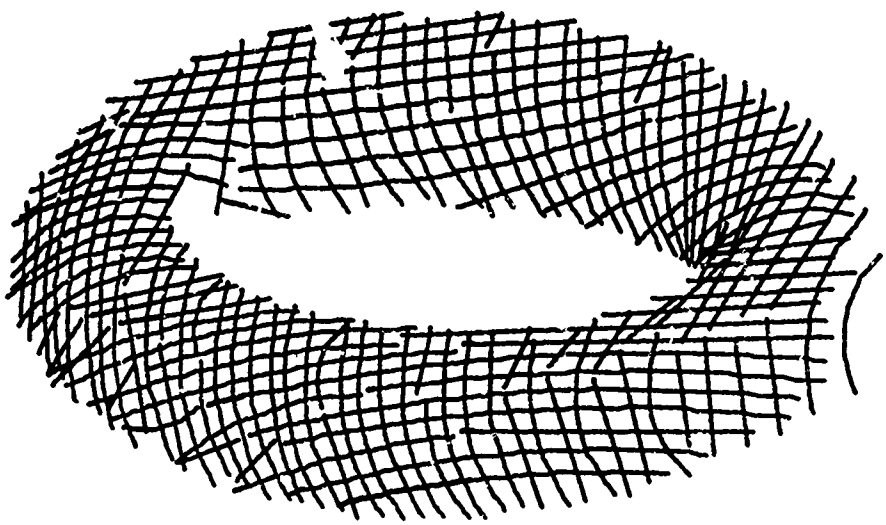


Figure B.3
Torus Fit With Parabolas

REFERENCES

[Baumgart] Bruce G. Baumgart, "GEOMED, a Geometric Editor" Stanford Artificial Intelligence Laboratory Operating Note Number 62, May, 1972.

[Blum] Harry Blum, "A Transformation for Extracting New Descriptors of Shape", Symposium on Models for Perception of Speech and Visual Form, Boston, November 11-14, 1964.

[Binford 70] Thomas O. Binford, "Triangulation by Laser", December, 1970, unpublished.

[Binford 71] Thomas O. Binford, "Visual Perception by Computer", presented at the IEEE Conference on Systems and Control, Miami, December, 1971.

[Coons] S. A. Coons and B. Herzog, "Surfaces for Computer-Aided Aircraft Design", J. Aircraft, Vol 1, No. 4 (July-Aug. 1968), pp 402-406.

[Courant] R. Courant, Differential and Integral Calculus, Interscience, 1936, Volume 2, pp. 190-199.

[Coxeter] H. S. M. Coxeter, Introduction to Geometry, John Wiley and Sons, 1961, pp321-326.

[DeBoor] Carl de Boor and John R. Rice, "Least Squares Cubic Spline Approximation I - Fixed Knots", Purdue University Report No. CSD TR 20, April 1968. "Least Squares Cubic Spline Approximation II - Variable Knots", Purdue University Report No. CSD TR 21, April 1968.

[Earnest] Lester D. Earnest, "Choosing an Eye for a Computer", Stanford Artificial Intelligence Project Memo AIM-51, April, 1967.

[Falk] Gilbert Falk, "Computer Interpretation of Imperfect Line Data as a Three-Dimensional Scene", Stanford Artificial Intelligence Laboratory Memo AIM-132, August, 1970.

[Feldman] J. Feldman, K. Pingle, T. Binford, G. Falk, A. Kay, R. Paul, R. Sproull, and J. Tenenbaum, "The Use of Vision and Manipulation to Solve the 'Instant Insanity' Puzzle", Second International Joint Conference on Artificial Intelligence, London, September 1-3, 1971.

REFERENCES

[Gill] "Visual Feedback and Related Problems in Computer Controlled Hand-Eye Coordination", forthcoming Ph.D thesis, Stanford University, 1972.

[Horn] Berthold Klaus Paul Horn, "Shape from Shading: A Method for Finding the Shape of a Smooth Opaque Object from One View", Ph.D. Thesis, Massachusetts Institute of Technology, June, 1970.

[Isaacson] Eugene Isaacson, and Herbert Bishop Keller, Analysis of Numerical Methods, John Wiley and Sons, 1966, pages 173-174.

[Krakauer] L. Krakauer, Ph.D. Thesis, MIT, 1970.

[Mott-Smith] John Mott-Smith, unpublished.

[Petit] Phil Petit, "RAID", Stanford Artificial Intelligence Laboratory Operating Note No. 58.1, February, 1971.

[Pingle] Karl K. Pingle, "Hand/Eye Library", Stanford Artificial Intelligence Laboratory Operating Note 35.1, January, 1972.

[Roberts 63] L. G. Roberts, "Machine Perception of Three-Dimensional Solids", Technical Report No. 315, Lincoln Laboratory, Massachusetts Institute of Technology, May, 1963.

[Roberts 65] L. G. Roberts, "Homogeneous Matrix Representation and Manipulation of N-Dimensional Constructs", Document MS1045, Lincoln Laboratory, Massachusetts Institute of Technology, May, 1965.

[Shirai] Yoshiaki Shirai and Motoi Suwa, "Recognition of Polyhedrons with a Range Finder", Second International Joint Conference on Artificial Intelligence, London, September 1-3, 1971.

[Smith] Lyle B. Smith, "The Use of Man-Machine Interaction in Data-Fitting Problems", Stanford Linear Accelerator Center Report No. 96, March, 1969.

[Sobel] Irwin Sobel, "Camera Models and Machine Perception", Stanford Artificial Intelligence Project Memo AIM-121, May, 1970.

REFERENCES

[Will] P. M. Will and K. S. Pennington, "Grid Coding: A Preprocessing Technique for Robot and Machine Vision", Second International Joint Conference on Artificial Intelligence, London, September 1-3, 1971.