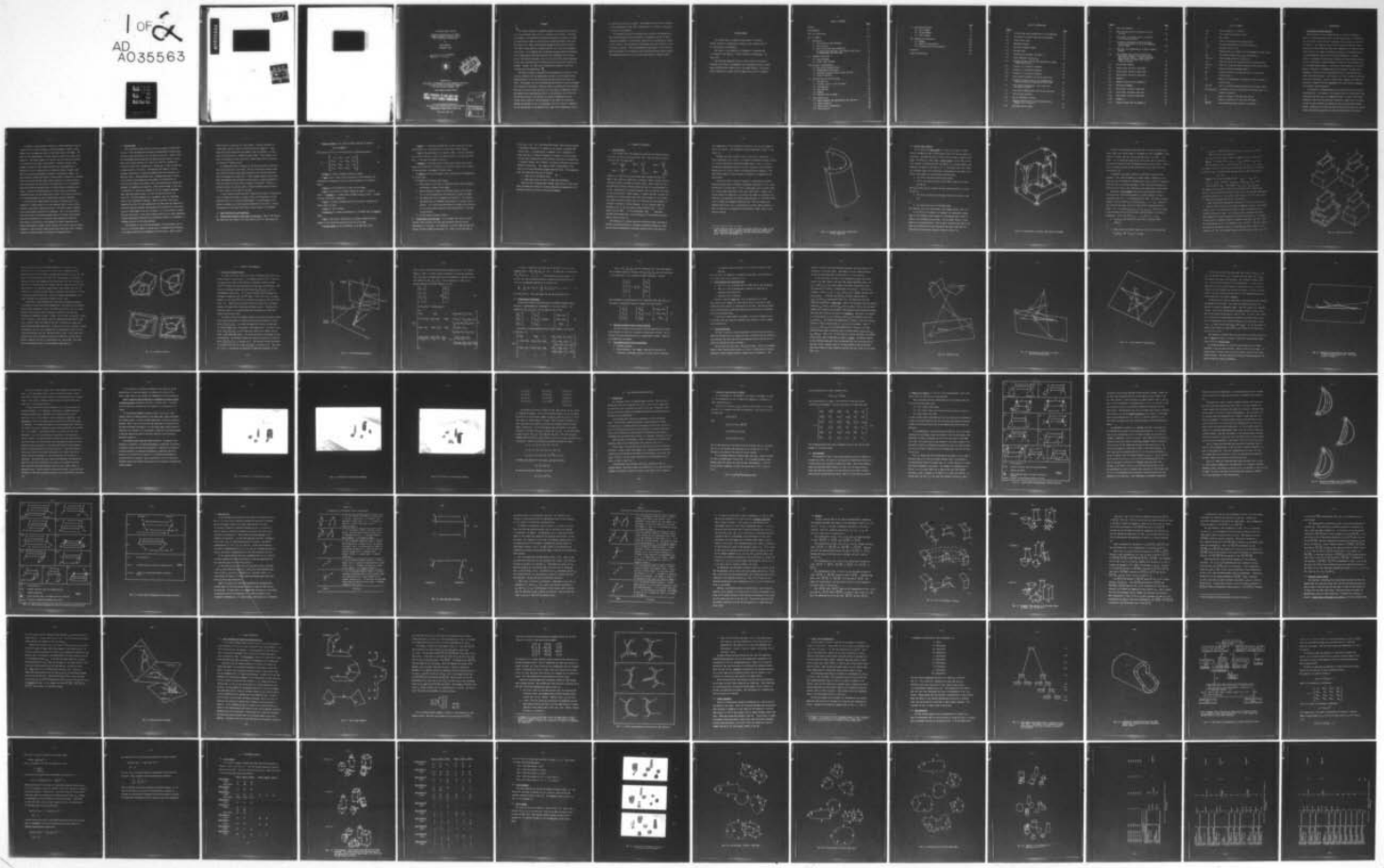
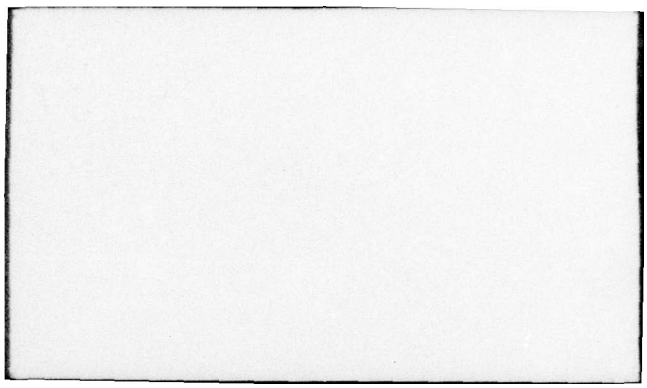
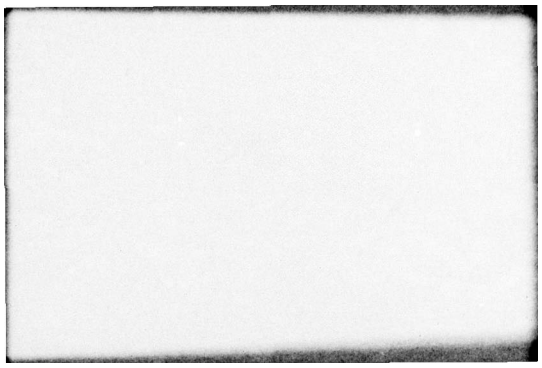
AD-A035 563 RENSSELAER POLYTECHNIC INST TROY N Y COMPUTER RESEAR--ETC F/G 9/2
COMPUTER RECONSTRUCTION OF BODIES BOUNDED BY QUADRIC SURFACES F--ETC(U)
SEP 76 R SHAPIRA AF-AFOSR-2937-76
UNCLASSIFIED CRL-48 AFOSR-TR-77-0051 NL







Technical Report CRL-48

Computer Reconstruction of Bodies Bounded by Quadric Surfaces from a Set of Imperfect Projections

by

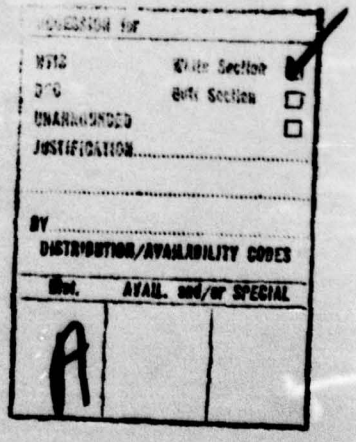
Ruth Shapira

September 1976

Prepared for

Directorate of Mathematical and Information Sciences
Air Force Office of Scientific Research
Air Force Systems Command, USAF

Grant Number AFOSR 76-2937



Computer Research Laboratory
Electrical and Systems Engineering Department
Rensselaer Polytechnic Institute

TROY, NEW YORK 12181

ABSTRACT

This thesis describes a computer program for constructing a description of solid bodies from a set of n pictures of the bodies. The bodies are assumed to be bounded by faces which are quadric or planar, and they are restricted to have all their vertices formed by exactly three faces. The pictures are taken from different vantage points, with the restriction that a slight shift in vantage point will not alter the topology of the picture. It is assumed that the program receives outline information from a preprocessor which has extracted this information from the pictures. The outline information (set of line structures) may be imperfect in that some junctions may be erroneously reported and some lines may be missing. However, all lines due to shadows are assumed to have been eliminated by the preprocessor.

The thesis includes a technique for establishing the validity of the junctions presented by the preprocessor as well as for matching corresponding features in the line structures derived from the different pictures. New grammar rules for line-drawing projections of curved and planar solid bodies are developed. These are useful in parsing the line drawings. They have also led to the definition of a new family of impossible objects. The program works simultaneously with all the available line structures. The parsing of every line structure is supported dynamically by the results gotten thus far from the parsing of the other line structures. Through the parsing of the line structures, the use of picture comparison and the application of the grammar rules, many of the preprocessor errors

are detected and partly corrected. The program also can provide feedback to the preprocessor in the form of suggestions as to where to look again for lines in the pictures.

The program utilizes the extracted line structures corresponding to the different bodies in all the pictures to determine the set of faces (insofar as possible) for every body. Every face is defined by an ordered set of n-tuples. The n-tuples are the matched lines and junctions in the n different pictures. The three-dimensional coordinates of the vertices and the equations of the faces can then be determined from these n-tuples. The program was written in PL/I and has been tested on several scenes.

ACKNOWLEDGMENT

The author wishes to express her gratitude to Professor Herbert Freeman for suggesting the topic of this research and for his guidance and encouragement.

The author is also grateful to Professors I. Cederbaum and A. Ginzburg of the Technion - Israel Institute of Technology - for their help.

This work was supported by the Air Force Office of Scientific Research, Directorate of Mathematical and Information Sciences, under grants AFOSR 70-1854, AFOSR 75-2755, and AFOSR 76-2937. It was also partly supported by a grant from the American Society for Technion.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF SYMBOLS

$A_{i,s}$	Junction number s in picture i
AB	Line connecting junction A and B
$\underline{a}$	The vector a
C_i	Center of projection for picture plane i
C_{ijq}	Intersection between line C_iC_j and plane P_q
F_i	Distance from C_i to P_i
G_i	Distance from C_i to center of corresponding film plane frame
J_{iq}	Intersection between line $A_{i,J}C_i$ and plane P_q
i=1,2,..n	Picture plane index (for n pictures)
i=0	Index for points not associated with any picture plane
j=1,2,..n	Index for coordinate system associated with picture plane j
j=0	Index for global coordinate system
$\underline{M}_i$	Projection matrix for projecting onto picture plane i
P_i	Picture plane i
$\underline{T}_i$	Matrix of transformation from global system to system of picture plane i
X^j,Y^j,Z^j	Axes of coordinate system associated with picture plane i
$X^j_{i,k},Y^j_{i,k},Z^j_{i,k}$	Coordinates of point k associated with picture plane i in coordinate system j
θ_i	Angle of rotation of X^i axis from X^0 axis
ϕ_i	Angle of rotation of Y^i axis from X^0Y^0 plane
$\overline{16_1,12_1}$	Line joining junctions 16 and 12 in picture 1

I. INTRODUCTION

1.1 The Problem of Scene Analysis

When we look at a good quality picture of a scene containing a number of three-dimensional objects, we are usually able to "understand" the scene; that is, we are able to perceive the real physical nature of the objects. The reason for this is that we have seen similar scenes before and at those times were able, by a combination of touch and additional views of the scene, to develop the ability of relating pictures of three-dimensional objects to the objects themselves. The same applies -- perhaps with the need for slightly more specialized learning -- to the understanding of scenes depicted in terms of line drawings, such as a draftsman might prepare. A picture (or line drawing) of a scene is merely a sentence in a language that we have learned. As with other languages, difficulties with understanding arise if the sentence contains words whose meaning is not known to us or if it contains unfamiliar syntactical structures. Further, sudden severe changes in context will disorient us because "picture languages," like other natural languages, are strongly context-dependent.

A scene may be incomprehensible from a picture because the picture contains errors. For example a line drawing (which is a specialized type of picture) may have some line segments missing. In such a case understanding of the scene can sometimes be gained by using a set of pictures (or set of line drawings) describing the scene from different vantage points. Although the pictures may each be defective in some way, collectively they may make possible the unique interpretation of the scene.

In computer scene analysis an image of a three-dimensional scene is converted into an array of pixels, with each assigned a number corresponding to the average grey value in that local area of the image. The image is then preprocessed to extract features suitable for scene interpretation. One common scheme is to extract the lines that are believed to correspond to the edges of the objects. Because of glare, haze, shadows, texture and other lighting conditions, some edges may be partially or wholly obscured and, for basically similar reasons, some spurious lines may be mistakenly thought to correspond to edges. The scene analysis program must extract as much useful information as possible from each picture and then relate this information to similar information devised from other pictures of the same scene (i.e., from pictures taken from different vantage points or under different lighting conditions). The use of multiple pictures is useful both for obtaining information about the three-dimensional nature of the objects as well as for resolving ambiguities due to imperfectly extracted edge data.

We can describe complex bodies in terms of volumes, that is, as assemblies of simple bodies, or we can describe them in terms of their surfaces, using faces, edges and vertices. The second type of description is particularly effective for polyhedra. But it is also suitable for curved bodies, especially man-made bodies, for which every distinct face can be described by a relatively simple mathematical equation. This thesis is restricted to the analysis of scenes containing only bodies with planar or quadric faces; however, the results could be easily modified to deal also with bodies whose curved faces are close to quadric surfaces, and this includes a big subset of man-made bodies.

1.2 Previous Work

The description scheme based on the use of volumes is exemplified by Agin (1) who described curved bodies as combinations of generalized cylinders, each described by its axis and cross section. Schemes using the faces-edges-vertices approach are almost entirely limited to polyhedra; they include Roberts (11), Guzman (7), Falk (6), Huffman (8), Clowes (4), Waltz (18), and many more. Nearly all of them utilized only a single picture of the scene. Although all were aware of the preprocessor limitations in extracting complete and valid edge data, the approach varied. Some maintained that a program analyzing an (unrealistic) perfect line drawing also contributes to understanding the process of perception, and indeed, their work provided much insight and guidance for computer scene analysis. They include Guzman (7) with his classification of junctions, Huffman (8) and his elegantly expressed edge labelling scheme, Clowes (4) with his special notation and Waltz (18), who made extensive use of grammatic rules for analyzing perfect line drawings with shadows. There is also the other group, exemplified by Falk (6), who assumed that the input data was imperfectly extracted from real pictures and imposed severe restrictions on the scene (such as preknowledge of the set of bodies). It is this writer's opinion that the restriction to data from a single picture by all these researchers created unnecessary difficulties and caused much effort to be expended on finding solutions to the wrong problems.

The extension of the work from polyhedra to curved bodies, even if only to a restricted family of curved bodies, increases the difficulties to an extent which may not be apparent at first glance. Some of these

difficulties are pointed out in this thesis. A partial extension to curved bodies can be found in the work by Chien and Chang (3). They dealt with bodies whose faces are planar, conic or cylindric, in which every curved face must be bounded by planar faces. They assumed a perfect line drawing for their input data. Their thesis can be considered an extension of Guzman's work.

As already mentioned, virtually all of the early work utilized data from only one picture of the scene. The use of multiple pictures taken from different vantage points was described in the recently published paper by Underwood and Coates (19). Their program analyzes convex polyhedra photographed from different vantage points, a single polyhedron in every set, where the input data is assumed to be perfect. The set of multiple pictures is thus not used to verify unreliable data but merely "to have a look" on all sides of the body.

In this thesis we shall show how a program can be made to "understand" a set of bodies with planar or quadric faces, utilizing defective data extracted from a set of multiple photographs. No preknowledge of the bodies is assumed, but certain general properties of the scene - such as the existence of precisely three faces in every vertex - are assumed.

1.3 Basic Definitions and Assumptions

a) Definitions relating to the bodies in the scene. Many of the following terms have been adapted from those used by Woon (20) and Clowes (4).

A quadric surface is the locus of points obeying the equation

$$Q(x,y,z) \equiv \underline{u}^T \underline{A} \underline{u} = 0 \tag{1}$$

where $\underline{u}=(x,y,z,1)^T$ is a vector and $\underline{A}$ is the coefficients matrix

$$\underline{A} = \begin{bmatrix} a_{11} & \frac{1}{2}a_{12} & \frac{1}{2}a_{13} & \frac{1}{2}a_{14} \\ \frac{1}{2}a_{12} & a_{22} & \frac{1}{2}a_{23} & \frac{1}{2}a_{24} \\ \frac{1}{2}a_{13} & \frac{1}{2}a_{23} & a_{33} & \frac{1}{2}a_{34} \\ \frac{1}{2}a_{14} & \frac{1}{2}a_{24} & \frac{1}{2}a_{34} & a_{44} \end{bmatrix} \tag{2}$$

A surface is either a quadric surface or a plane.

An edge is all or part of the intersection of two surfaces; the intersection is bounded by one or two other surfaces or is closed on itself.

A vertex is the intersection of three or more edges.

A face is a portion of a surface bounded by edges or closed on itself. A boundary is a closed chain of edges bounding a face. A single face may have several boundaries.

A body is a closed, connected part of the 3D space, delimited by a finite number of faces.

A scene is a set of bodies.

A projection is a central projection of a 3D entity onto the picture plane.

A limb is the locus of the points on a quadric surface that are tangent to the projecting rays and do not lie on an edge.

A virtual vertex is the intersection of an edge and a limb.

A region is a connected, visible part of the projection of a face. The projection of a face may result in zero, one, or more regions.

A line is the projection of an edge or a limb. The line projected by a limb will also be called a limb when this will not cause any confusion. A line may be straight or curved.

A junction is the intersection of two or more lines, and is thus either the projection of a vertex or of a virtual vertex, or is a result of obscuring part of an edge or limb by a face.

An object is the set of regions, lines, and junctions corresponding to a single body.

We impose the following restrictions:

1. Every vertex in the scene is formed by exactly three surfaces, and belongs to exactly three edges.
2. Smooth transition between two different faces is not allowed (i.e., the derivative must be discontinuous across the edge).
3. The camera position is assumed to be "general." For example, the projections of different vertices may not coincide in any picture. If a vertex does not belong to an edge, the projection of the vertex may not coincide with the projection of the edge in any picture.
4. No limb passes through a vertex.

b) Preprocessor and input data. It is assumed that there is a preprocessor which extracts the lines and junctions from the picture (photograph) of the scene. The extraction of curved lines has been the subject of work by Ramer (10) and Agin (1). There is also the work of

Shirai and Tsuji (15), who extracted straight lines from photographs of polyhedra, getting rid of shadows at the expense of missing some visible lines. Our preprocessor is assumed to be a combination of Shirai and Tsuji's (ORing and ANDing of a sequence of pictures taken from the same vantage point under different conditions of illumination to eliminate shadows) and of Ramer's (curved-line extraction and determination of junctions). The output of such a preprocessor is the input to the scene description program described here. The assumptions about the input data thus are as follows:

1. There are no extraneous lines.
2. Some valid lines (or parts of them) may be missing.

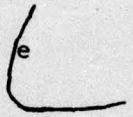
Since the kind of preprocessor assumed here was not in fact available, the data for testing of the scene description program had to be extracted visually from sets of actual photographs.

II. GRAMMATIC PROPERTIES

2.1 Junction Types

In accordance with the assumptions stated in the previous section, junctions can have no more than three lines. They can thus be classified into the following types:

Type W  Type V  Type Y 

Type T  Type A  Type S 

Types W, V, Y and T are the well-known types, generalized here (7). Types Y and W must be projections of vertices. Type T results from the covering of part of an edge or limb by a face. Types A and S are the projections of virtual vertices, where the line e is the projection of a limb. Characteristic to junction types A and S is that all the lines in the junction have a common tangent. A type-V junction is either the projection of a vertex whose third line is not visible or it is a junction which could be of either type Y or W but has one line missing, or of type T with part of the bar missing. If an S junction should be missing a line, it can look like a type-A junction  , like a type-V junction  , or be undetectable  . (The dots indicate the missing line.) An illustration of the different junction types is given in Fig. 2.1.

The detection of a type-A junction is difficult since this type of junction is characterized by a continuous transition between two lines. The two lines are governed by different equations and on the basis of

this segmentation of the continuous contour into the two line segments should be possible. Such segmentation was accomplished by Agin (1) and Albano (2).

Through junctions of types S and A, a limb can be identified. A limb's importance lies in that it is projection-dependent and has no match in a picture taken from a different vantage point. Also, it conveys information regarding the type of the face (cylinder, cone, ellipsoid, hyperboloid, etc.) and is useful in finding the face's equation. The type identification of each junction is assumed to be supplied by the preprocessor.

It would be useful to have an extended (or generalized) junction labelling scheme based on Huffman's polyhedral labelling ("-" for a line projected by a concave edge, "+" for a convex edge, and a directed arrow for a convex edge created by a front and a back face*). However, for curved objects the labelling of a line may not always be consistent and, therefore, the usefulness of junction labelling here is of doubtful value since it cannot be propagated along the line. (See for example the labelled line in Fig. 2.1). The distinction in this thesis between a Y and a W junction is in keeping with past practice rather than for any practical reason.

---

* A front face is a face for which the outward normal (at least at the edge in question) has a component directed toward the vantage point. For a back face, the component is directed away from the vantage point. See Woon and Freeman (21).

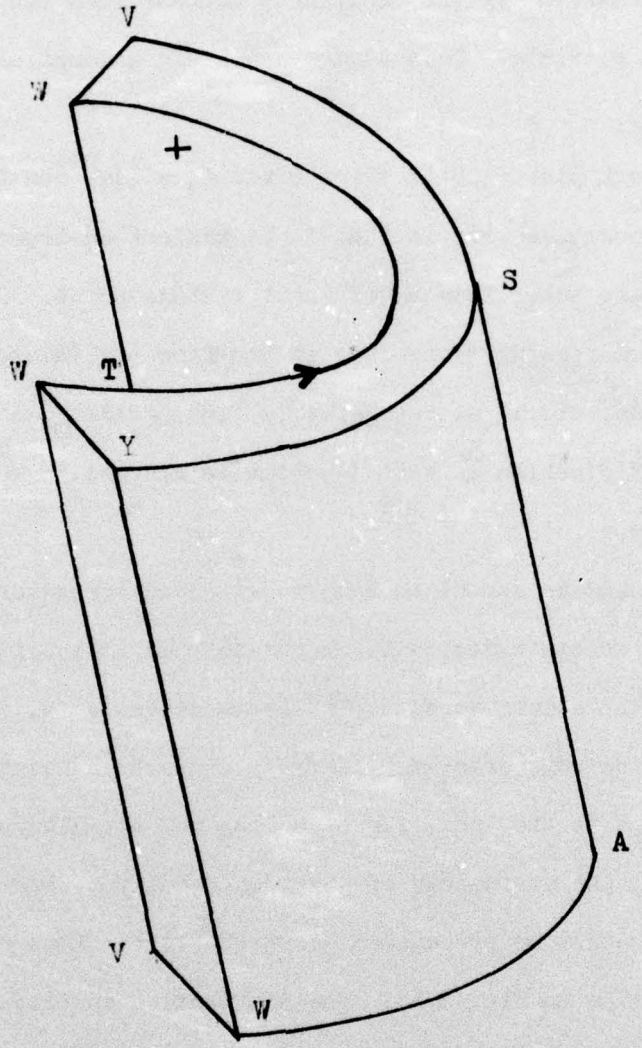


Fig. 2.1 Junction Types and Inconsistency of Line Labelling

2.2 Cyclic-Order Property

Let us define the depth index of a line as the number of faces modulo-2 between the edge corresponding to this line and the center of projection. As mentioned before, every vertex belongs to exactly three edges. We define for these edges a cyclic order in the vertex, as the order induced by "walking around" the vertex in a clockwise manner, and numbering the edges in the order 1, 2, 3. The cyclic order will retain its clockwise sense in any projection, irrespective of the shape of the vertex, under either of the following two conditions:

1. All three lines have an even depth index.
2. One of the three lines has an even depth index and two have an odd one.

The cyclic order will be reversed (become counter-clockwise) in a projection if:

1. One of the lines has an odd depth index and two have an even one.

or

2. All three lines have an odd depth index.

The different cases are illustrated by the example shown in Fig. 2.2.

If we trace out the edges of a boundary, we consistently change edges at the vertices, either always in a decreasing cyclic order or always in an increasing cyclic order. Let us always choose to trace edges in an increasing order. Then we shall traverse every edge in two different directions as we walk around the two faces that share it. (See the two arrowed lines along the line AB in Fig. 2.2).

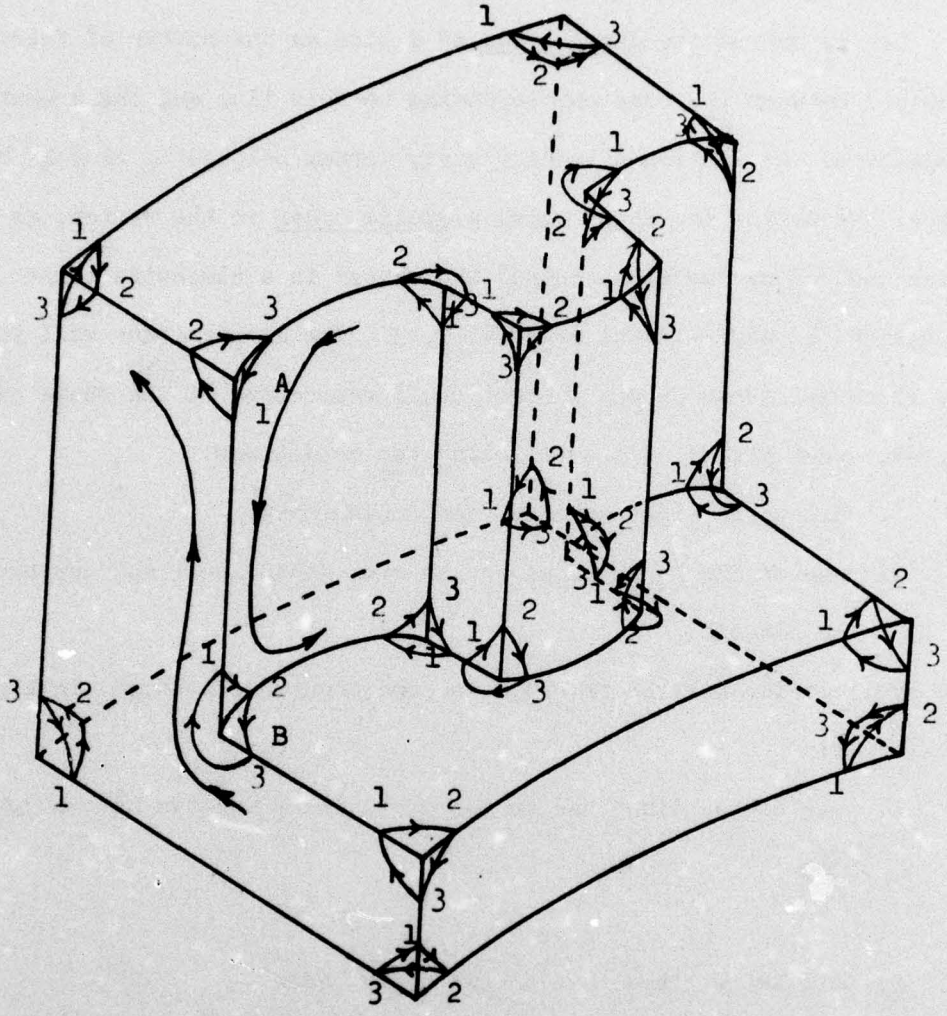


Fig. 2.2 Illustration of Cyclic Order and Line Assembly

From the foregoing we conclude that when a vertex's projection has three lines, their cyclic order is determined by their arrangement in the picture. However, when one line is missing, the cyclic order is not known. Let us define the relation AB < AC to mean that in junction A the line AC follows the line AB immediately in the cyclic order. Then if A is a two-line junction, we have either the relation AB < AC or the relation AC < AB.

Knowledge of the cyclic order at a junction is crucial to the assembly of the lines belonging to a single region. Therefore, it is important to have a means for determining the cyclic order in a junction when the order is not given naturally. Some rules and techniques are stated in this section, with additional rules given later in the thesis.

We define a line assembly (LA) as the directed path followed in tracing out the lines corresponding to a single boundary - or any continuous part of it - in increasing cyclic order. We denote an LA in picture i by the ordered set of junctions $A_{i,1}$ ..... $A_{i,n}$ visited in the course of the trace. Now it follows from the definition that for every $1 < k < n$ we have $A_{i,k}\,A_{i,(k-1)} < A_{i,k}\,A_{i,(k+1)}$. Two LA's are said to be distinct if they trace out lines corresponding to two distinct boundaries. If for two LA's, $A_{i,1} \ldots A_{i,n}$ and $A_{j,1} \ldots A_{j,m}$

1. $A_{i,1}=A_{i,n}$, and there is at least one A_j different from every A_i

or

2. There are two successive junctions in the two LA's such that $A_{i,k}=A_{j,\ell}$ and $A_{i,(k-1)} = A_{j,(\ell+1)}$

or

3. $A_{i,n} = A_{j,1}$ and the relation $A_{j,1} A_{j,2} < A_{i,(n-1)} A_{i,n}$ exists. Then the two LA's are distinct.

Two rules forcing a cyclic order in a junction can be stated now:

Rule 1: If we have an LA, $A_{i,1} \ldots A_{i,n}$, in which $A_{i,1} \equiv A_{i,n}$ then we must have in $A_{i,1}$ the relation $A_{i,1} A_{i,(n-1)} < A_{i,1} A_{i,2}$.

For example, in Fig. 2.3(a) we have the LA B,C,D,A,B, (The LA is shown as an arrowed line). The lines in B are thus forced to have the relation

$$BA < BC$$

Rule 2: If we have two distinct LA's, $A_{i,1} \ldots A_{i,n}$ and $A_{j,1} \ldots A_{j,m}$, such that $A_{j,m} \equiv A_{i,1}$ and $A_{j,(m-1)} \not\equiv A_{i,2}$, then we must have in $A_{i,1}$ the relation $A_{i,1} A_{i,2} < A_{j,m} A_{j,(m-1)}$.

For example in Fig. 2.3(b) we have the two distinct LA's: E,F,C,B and B,A,G,E. They are distinct because of the cyclic order already established in junction B by Rule 1. Now by Rule 2 the relation $EF < EG$ must hold in E. (This in turn establishes the LA M,F,E,G,N,O).

A more complicated strategy may be adopted to force the correct order at a junction when these two rules cannot be applied directly. One assumes a cyclic order in the junction and tries to reach a contradiction by a sequence of inductive steps. If a contradiction is reached, the cyclic order opposite to the one selected is forced on the junction. For example, in Fig. 2.3(c) we assume an order in junction R such that $RQ < RN$. Then we can establish the LA's P,Q,R,N,G,A,D and D,C,F,M,P. They are distinct as implied by the natural order in D, and hence by Rule 2 force in P the relation $PQ < PM$. At the same time we have the two LA's R,Q,O and O,N,R

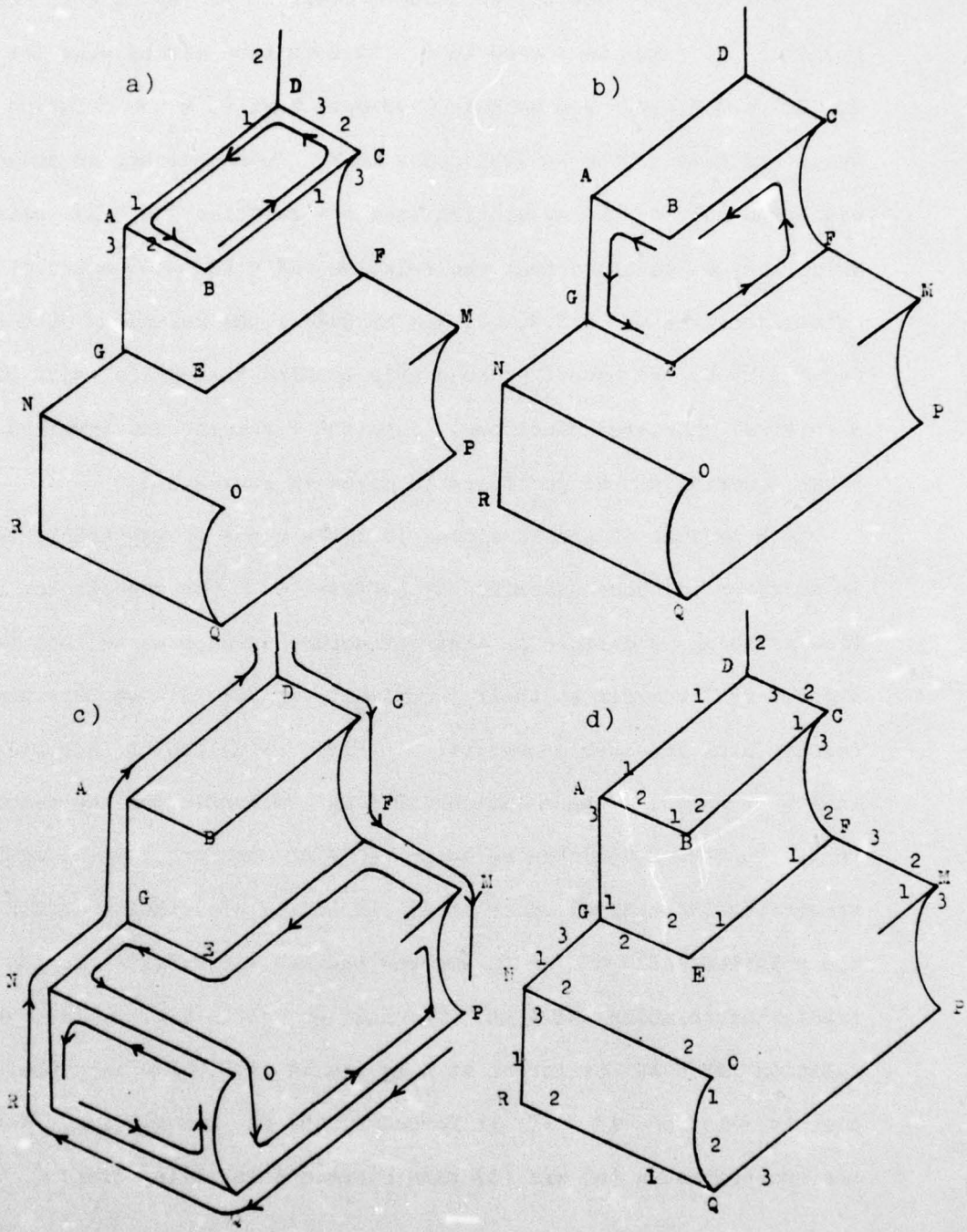


Fig. 2.3 Forced Cyclic Order

which are distinct due to the assumed order in R. By Rule 2, the relation ON < OQ is forced in O. This in turn establishes the LA M,F,E,G,N,O,Q,P,M. Now by Rule 1 we must have in M the relation MP < MF, which contradicts the natural order in M. This sequence of deductions was based only on the assumption that the relation RQ < RN exists at R, which lets us conclude that the relation RN < RQ must exist at R. This establishes the LA O,N,R,Q,O, and by Rule 1 the relation OQ < ON is induced at O. We have thus been able to find the cyclic order at 4 of the 5 original unordered junctions. Junction P remains undetermined. The final ordering of the junctions is given in Fig. 2.3(d).

A byproduct of the foregoing is a new class of impossible objects, in addition to those described by Huffman (8). The objects can be identified as being impossible if their structure is such as to lead to contradictory cyclic order at their junctions. In Fig. 2.4 we have some instances of impossible objects. In Fig. 2.4(a) the LA A,E,D,C,B,A forces in A - by Rule 1 - the relation AB < AE contradicting the natural order in A. In Fig. 2.4(b) the relation AD < AB is forced in A, again contradicting the natural order at A. In Fig. 2.4(c) the LA A,E,F,B,A forces the relation AB < AE in A, whereas the LA A,B,C,D,E,A forces the contradicting relation AE < AB. The same occurs in Fig. 2.4(d), where the relation AB < AE is forced at A by the LA A,E,F,B,A and the contradictory relation AE < AB is forced by the LA A,B,C,D,E,A. Note that the two polyhedra (a) and (d) pass Huffman's labelling test.

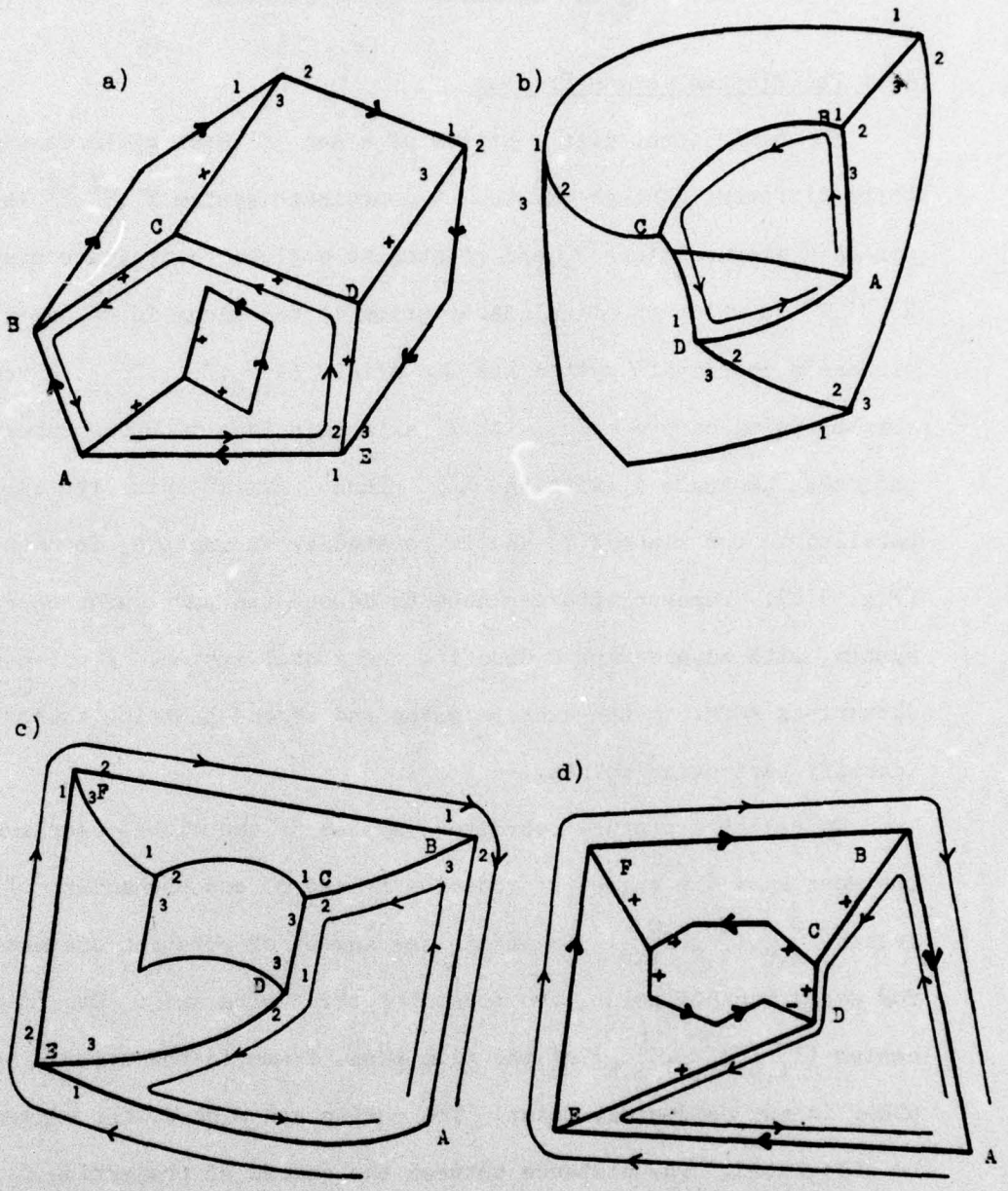


Fig. 2.4 Impossible Objects

III. GEOMETRIC CONSIDERATIONS

3.1 The Picture Taking Process

The basic input data consists of a set of three pictures taken from three different vantage points. A coordinate system X^i,Y^i,Z^i is defined for each picture plane P_i and related to a global coordinate system X^0,Y^0,Z^0 in which the final description of the scene is determined. Each picture's coordinate system has its origin $(X^0_{i,0},Y^0_{i,0},Z^0_{i,0})$ on the corresponding camera axis. Its Y^i axis coincides with the camera axis and makes an angle ϕ_i with the X^0Y^0 plane. Its X^i axis lies in a plane parallel to the plane X^0Y^0 and is rotated by an angle θ_i from the X^0 axis (Fig. 3.1). Superscripts are used to denote the particular coordinate system, with superscript 0 denoting the global system. First-position subscripts refer to the picture plane and second-position subscripts identify particular points.

To relate a picture coordinate system to the global coordinate system one must know the angles of rotation θ_i and ϕ_i and the amount of translation $X^0_{i,0},Y^0_{i,0},Z^0_{i,0}$. To obtain the angles of rotation one measures, for every vantage point, two points on the camera axis. One point is the center $(X^0_{i,N},Y^0_{i,N},Z^0_{i,N})$ of the film plane frame in the camera, and the other is any convenient point. The cosine and sine of the angles can now be calculated. The distance between the center of projection C_i and the point $(X^0_{i,N},Y^0_{i,N},Z^0_{i,N})$ is denoted by G_i. The distance between the center of projection and the point $(X^0_{i,0},Y^0_{i,0},Z^0_{i,0})$ is denoted by F_i. The ratio $R_i = G_i/F_i$ is determined by measuring corresponding diagonals of both

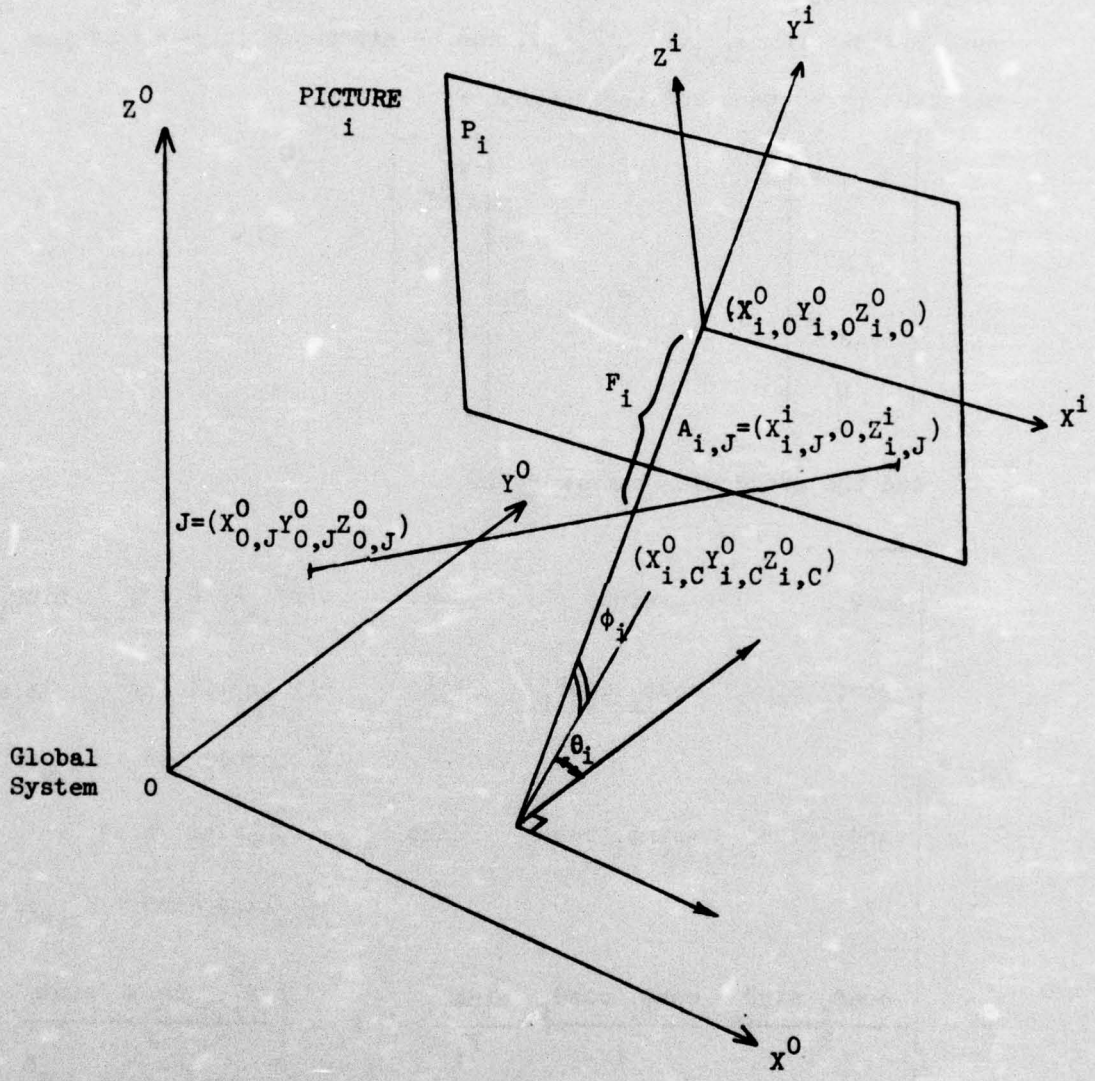


Fig. 3.1 The Picture-Taking System

the film plane frame and the uniformly enlarged picture. The relation between a point J in space, whose coordinates in the global system are $(X^0_{0,J}, Y^0_{0,J}, Z^0_{0,J})$, and whose projection coordinates in a picture coordinate system are $(X^i_{i,J}, Y^i_{i,J}, Z^i_{i,J})$, can be expressed in terms of the measured parameters and the unknown F_i (5,14):

$$\begin{bmatrix} X^i_{i,J} & W \\ Y^i_{i,J} & W \\ Z^i_{i,J} & W \\ W \end{bmatrix} = \underline{M}_i \begin{bmatrix} X^0_{0,J} \\ Y^0_{0,J} \\ Z^0_{0,J} \\ 1 \end{bmatrix} \tag{3}$$

And the projection matrix $\underline{M}_i$ is

$$\underline{M}_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 & -(X^0_{i,N}\cos\theta_i + Y^0_{i,N}\sin\theta_i) \\ -\cos\phi_i \sin\theta_i & \cos\phi_i \cos\theta_i & \sin\phi_i & -(F_i(R_i+1) + X^0_{i,N}\cos\phi_i\sin\theta_i + Y^0_{i,N}\cos\phi_i\cos\theta_i + Z^0_{i,N}\sin\phi_i) \\ \sin\phi_i \sin\theta_i & -\sin\phi_i \cos\theta_i & \cos\phi_i & -(X^0_{i,N}\sin\phi_i \sin\theta_i - Y^0_{i,N}\sin\phi_i\cos\theta_i + Z^0_{i,N}\cos\phi_i) \\ \frac{-\cos\phi_i \sin\theta_i}{F_i} & \frac{\cos\phi_i \cos\theta_i}{F_i} & \frac{\sin\phi_i}{F_i} & -R_i - \left(\frac{-X^0_{i,N}\cos\phi_i \sin\theta_i}{F_i} + \frac{Y^0_{i,N}\cos\phi_i \cos\theta_i + Z^0_{i,N}\sin\phi_i}{F_i}\right) \end{bmatrix} \tag{4}$$

In order to compute F_i for every picture we look at a set of pre-measured points $\{(X^0_{0,j}, Y^0_{0,j}, Z^0_{0,j})\}$, $j=1 \ldots n$, whose set of projections $\{(\hat{X}^i_{i,j}, \hat{Z}^i_{i,j})\}$ $j=1 \ldots n$, $i=1 \ldots 3$, are measured in every picture. To minimize the sum-of-squares differences between the calculated projections and the measured projections we require that

$$\frac{\partial}{\partial F_i} \left(\sum_{j=1}^{n} (X^i_{i,j} - \hat{X}^i_{i,j})^2 + \sum_{j=1}^{n} (Z^i_{i,j} - \hat{Z}^i_{i,j})^2 \right) = 0, \quad i=1 \ldots 3 \qquad (5)$$

for every picture. The three equations can now be solved for F_i.

3.2 Transformation Techniques

Interrelation between the three picture coordinate systems can be achieved by transforming the coordinates via the global system. The translation $(X^0_{i,0}, Y^0_{i,0}, Z^0_{i,0})$ can be expressed as follows:

$$\begin{bmatrix} X^0_{i,0} \\ Y^0_{i,0} \\ Z^0_{i,0} \end{bmatrix} = \begin{bmatrix} X^0_{i,N} \\ Y^0_{i,N} \\ Z^0_{i,N} \end{bmatrix} + F_i(R_i+1) \begin{bmatrix} -\cos\phi_i \sin\theta_i \\ \cos\phi_i \cos\theta_i \\ \sin\phi_i \end{bmatrix} \qquad (6)$$

The matrix of transformation from the global system to the system of pictures i is:

$$\underline{T}_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 & -(X^0_{i,0}\cos\theta_i + Y^0_{i,0}\sin\theta_i) \\ -\cos\phi_i \sin\theta_i & \cos\phi_i \cos\theta_i & \sin\phi_i & -(-X^0_{i,0}\cos\phi_i \sin\theta_i + Y^0_{i,0} \cos\phi_i \cos\theta_i + Z^0_{i,0}\sin\phi_i) \\ -\sin\phi_i \cos\theta_i & -\sin\phi_i \cos\theta_i & \cos\phi_i & -(X^0_{i,0}\sin\phi_i \sin\theta_i - Y^0_{i,0} \sin\phi_i \cos\theta_i + Z^0_{i,0}\cos\phi_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (7)$$

Thus if $X^j_{0,J}\ Y^j_{0,J}, Z^j_{0,J}$ are the coordinates of a point expressed in the coordinate system of picture j and $X^i_{0,J}\ Y^i_{0,J}, Z^i_{0,J}$ are the coordinates of the same point in the coordinate system of picture i, we have

$$\begin{bmatrix} X^i_{0,J} \\ Y^i_{0,J} \\ Z^i_{0,J} \\ 1 \end{bmatrix} = \underline{T}_i\ \underline{T}_j^{-1} \begin{bmatrix} X^j_{0,J} \\ Y^j_{0,J} \\ Z^j_{0,J} \\ 1 \end{bmatrix}$$

Other parameters of importance are the coordinates $(X^0_{i,C}, Y^0_{i,C}, Z^0_{i,C})$ of the center of projection (center of lens) C_i for every picture:

$$\begin{bmatrix} X^0_{i,C} \\ Y^0_{i,C} \\ Z^0_{i,C} \end{bmatrix} = \begin{bmatrix} X^0_{i,N} \\ Y^0_{i,N} \\ Z^0_{i,N} \end{bmatrix} + F_i\ R_i \begin{bmatrix} -\cos\phi_i\ \sin\theta_i \\ \cos\phi_i\ \cos\theta_i \\ \sin\phi_i \end{bmatrix} \qquad (8)$$

3.3 Matching Incomplete Data in Three Pictures

The line structure data extracted from the pictures may be incomplete because of occlusion or because of preprocessor failure. Let us consider the data imperfections due to preprocessor failure. They can be divided into two groups:

(1) Data imperfections involving junctions.

a. Missing junctions.

b. False junctions - for example, when half of the bar of a T junction is missing, causing it to look like a V junction.

c. A V junction that is actually a Y or W junction with one line missing.

Note that the V in subgroup c, although of wrong type, is the projection of a vertex; this is not true for V in b.

(2) Data imperfections involving lines.

a. Part of a line is missing and the broken end is open (dangling).

b. Part of a line is missing and a junction of wrong type is created at the broken end.

c. The whole line is missing.

We call a junction valid when it is a projection of a vertex. Thus, junctions of types A, S, and T are not valid, junctions of type Y and W are valid and for V junctions we do not know. The precise nature of a V junction must be left undecided until more evidence is collected, as will be explained later.

A junction of type S which is missing a line and is reported as an A junction, does not cause any confusion and, therefore, is not counted as an imperfection.

3.4 Junction Matching

In order to arrive at a true description of the 3D scene in spite of some data imperfections, we compare the line structures from the three pictures against each other and use the information found in one picture to check the information found in another.

Let us start with junctions. We have two goals. One is to determine which of the V junctions are valid, or, to put it differently for which V junction is there enough evidence to support such an assumption. The

second is to group, from the different pictures, junctions that are the projection of the same vertex. From here on in this section whenever we use the term "junction" we shall mean a V, Y or W junction.

If two junctions, one each in two different pictures, are projections of the same vertex, they must obey certain geometric rules. Let P_i, C_i and $A_{i,J}$ be the picture plane, the center of projection, and the projection of vertex J, respectively, for picture i. Then for two pictures, i and j, C_i, C_j, J, $A_{i,J}$, and $A_{j,J}$ are coplanar, and the lines $C_iA_{i,J}$, $C_jA_{j,J}$ intersect any plane P_q in three colinear points: C_{ijq}, J_{iq} and J_{jq}, respectively (Fig. 3.2). We shall call the line formed by these points a <u>match line</u>. Thus if we re-project into the plane P_q junction $A_{i,J}$ from C_i, junction $A_{j,J}$ from C_j, and C_i from C_j, we can check whether this condition is satisfied. Two junctions from different pictures that obey this condition are said to be <u>matchable</u>. Obviously, if a vertex I is in the plane C_i, C_j, J then $A_{i,I}$ and $A_{j,J}$ are matchable and so are $A_{i,J}$ and $A_{j,I}$. To resolve such ambiguities, we refer to the third picture, picture k. Now we require that C_i, C_j, C_k be not colinear. Then if $A_{i,J}$, $A_{j,J}$, and $A_{k,J}$ are the three projections of vertex J, the three different match lines on plane P_q form a triangle whose vertices are J_{iq}, J_{jq}, and J_{kq}. (Fig. 3.3). We call the set of three junctions, from different pictures, that are matchable in pairs, a <u>triple</u>. By forming triples we can eliminate many pairs that are matchable but are not projections of the same vertex. However, even in the pure geometric case we may have three projections of three different vertices A,B, and D that form a triple. (Fig. 3.4).

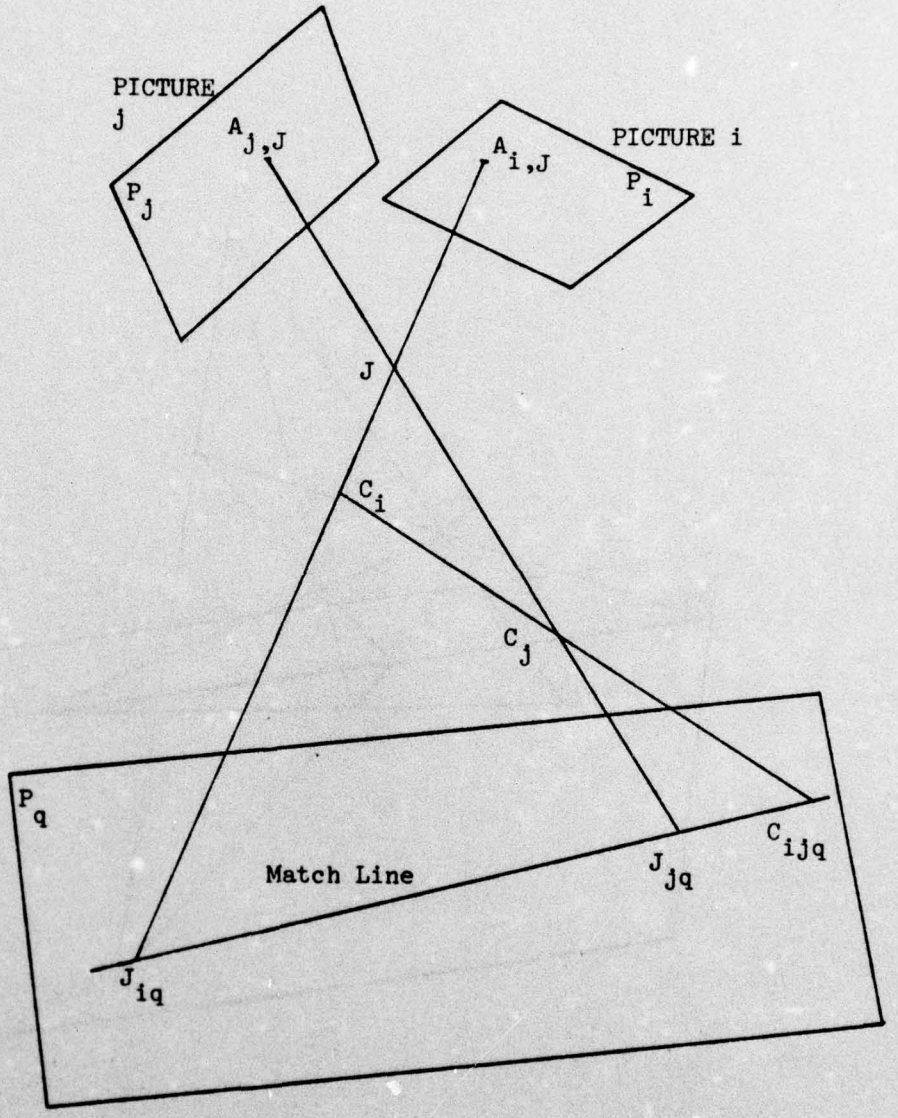


Fig. 3.2 The Match Line

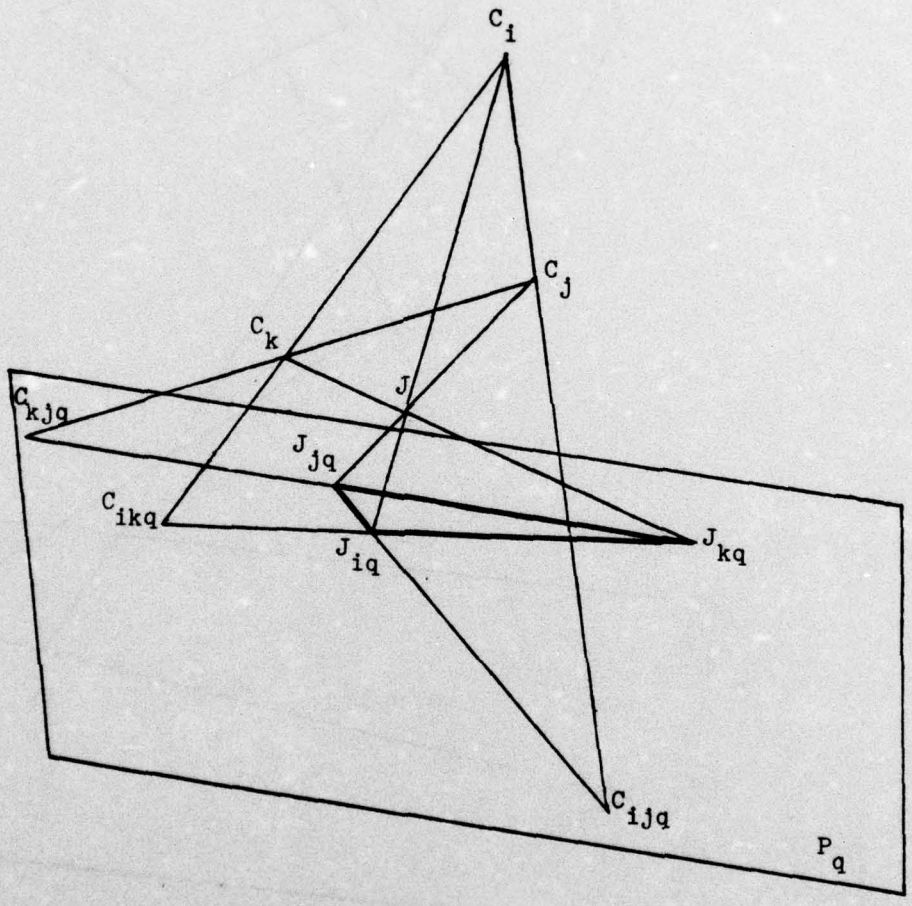


Fig. 3.3 The Match-Line Triangle for Vertex J
(Match Lines Shown Bold)

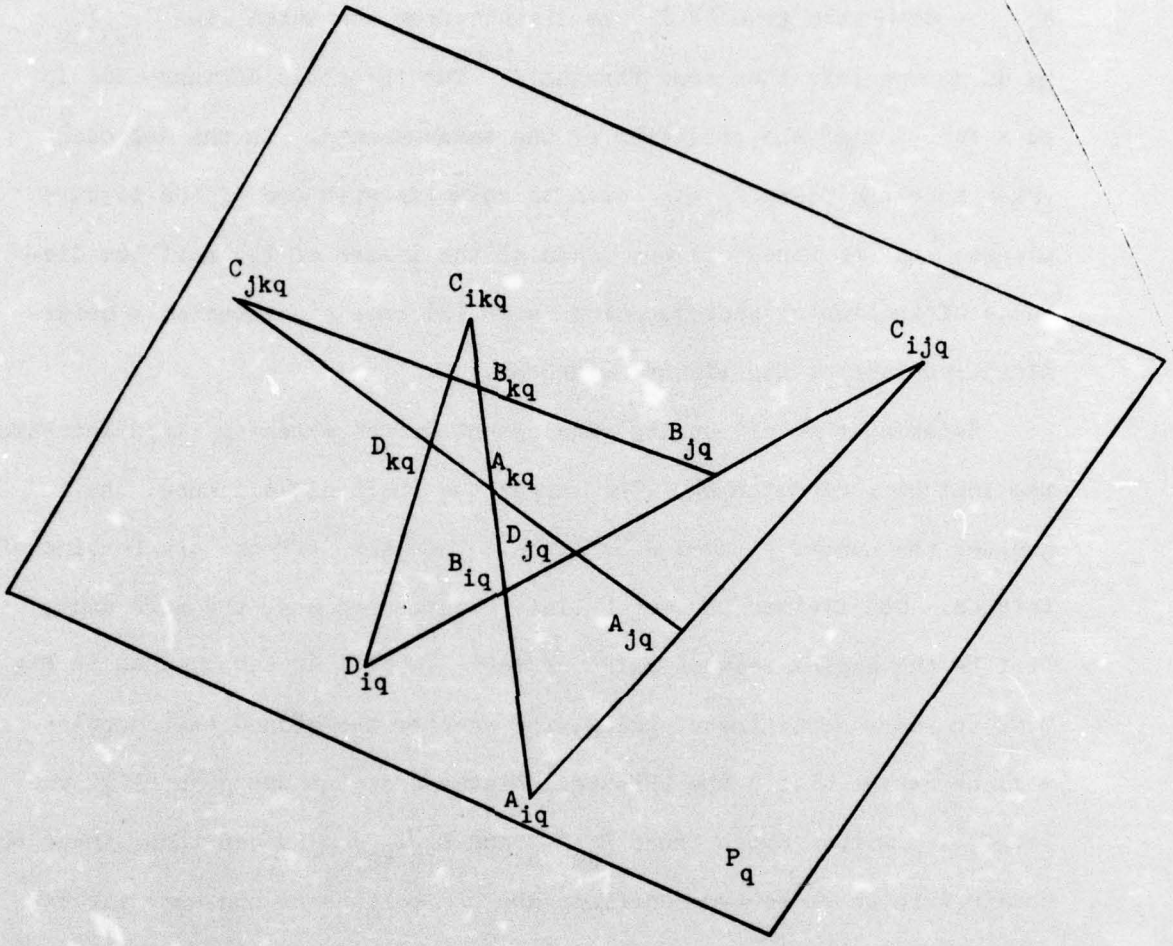


Fig. 3.4 Triple Ambiguity (A_{kq},B_{iq},D_{jq})

We are dealing here with real data; thus we must accept $A_{i,J}$ and $A_{j,J}$ as matchable even if J_{iq} is distant from the match line $C_{ijq}J_{jq}$ by an amount less than some threshold. The threshold distance should be a function of the precision of the measurements. In the approach taken here the plane P_q was taken to coincide with one of the picture planes, and the threshold was taken as the lesser of (1) half the distance of the two closest junctions and (2) some experimentally determined fraction of the picture diagonal.

Relaxing the collinearity requirement on the matching lines increases the ambiguity of matching. The larger the threshold distance, the greater the number of ambiguous cases. This also affects the forming of triples. The flatter the match-line triangle becomes, the more acute will be the angles between pairs of match lines. As can be seen in Fig. 3.5, the more acute these angles, the greater the chance that a point will be caught within the threshold distance of two wrong match lines (e.g., I_{iq} within match lines $B_{jq}B_{iq}$ and $B_{iq}B_{kq}$). We can thus expect to obtain triples where two junctions are projections of the same vertex but the third is not.

Two matchable junctions that are projections of the same vertex are said to <u>match</u> each other. A triple in which the junctions match each other is called a <u>match triple</u>.

To establish matches, we find for every junction all matchable junctions in the two other pictures. Then we form all the triples. To find the match triples in the set of all triples, we have to use some picture context. The more severe the conditions causing ambiguity, the more the need for context information.

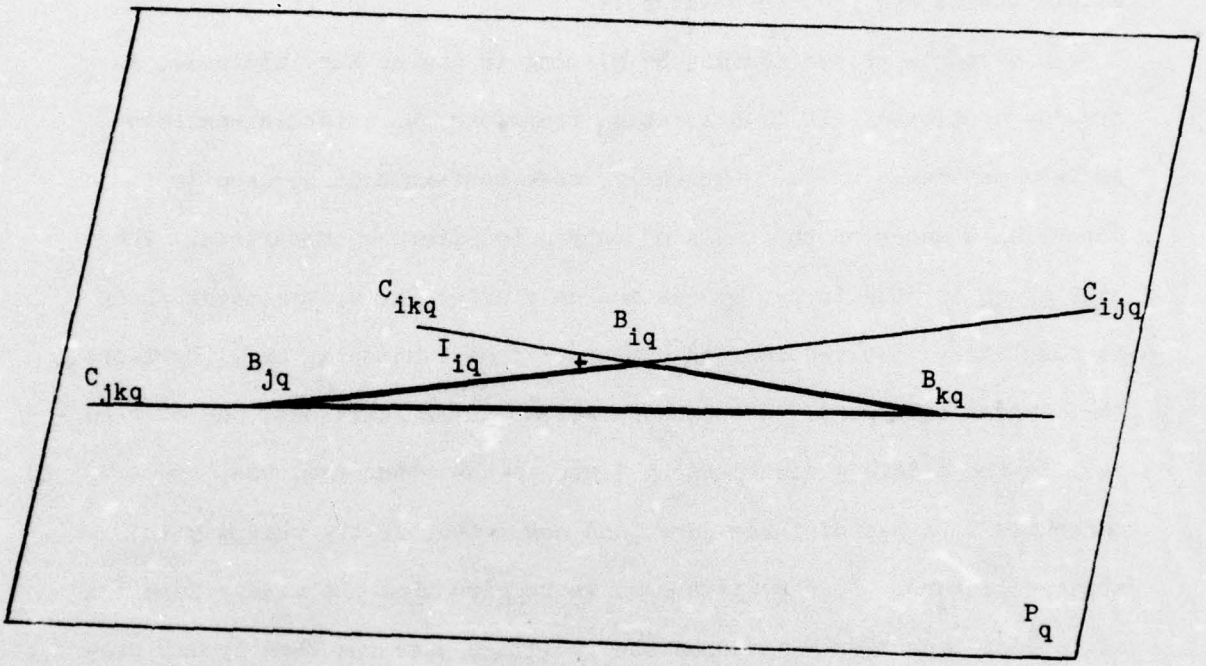


Fig. 3.5 Ambiguity Problem Arising from Acuteness of Angles Between Match Lines. (The wrong triple is created.)

The kind of context we shall use is line connections between junctions. If we have three lines - each in a different picture - whose ends form two triples, the two triples are each considered to be a match triple. (The chance of naming a wrong triple as a match triple still exists but is now reduced greatly).

A vertex's projection may be missing in one or more pictures, as already mentioned. It is desirable, therefore, to establish matches between matchable pairs. Naturally, more context must be used to establish a match on the basis of only a two-picture comparison. The pair match is done in two passes and only after the triple match phase is completed. (It reduces the number of free candidates and, therefore, the chances of error). We select a set of three junctions, one of them - call it the middle - connected by lines to the other two, that are matchable to a set of three junctions connected, in the same way in another picture. In the first pass we require that the middle junction will be of type Y or W and that the two lines have the same cyclic precedence in the two junctions. The three junctions are then assumed to match the three junctions in the other picture. In the second pass we allow one or both of the middle junctions to be of type V and do not check the cyclic precedence (it is not known yet for the two-line junction). The second pass is a little more hazardous, not having the ability to use cyclic precedence, but since it is carried out after most junctions are already matched, there is only a small chance of making an error. We also accept two junctions forming a loop, which are matchable to two other junctions forming a loop, as matching to the other two.

If one junction in a triple is matched to the other two by two applications of a pair-wise match, we consider the triple to be a match triple; that is, we consider the remaining two as matching also.

Every V junction that is matched to a junction in another picture is marked as valid (although it may miss a visible line). The match supplies additional evidence that the junction is a projection of a vertex.

An illustrative example is shown in Fig's. 3.6 to 3.8. The figures consist of three pictures of the same scene, taken from different vantage points. In every picture some lines, or parts of lines are missing. Many Y and W junctions are thus reported as V junctions, but the most misleading information is in the false shape identification of junctions 15 in picture 1, 27 in picture 2 (both are T junctions missing a line and reported as type V) and 24 in picture 2 (an S junction reported as type V).

The program reports many matchable relations. To mention a few: junction 9 in picture 1 is reported matchable to junctions 1,8,10,12,27 in picture 2 and to junction 21 in picture 3. Junction 27 in picture 2 (a false junction) is reported as matchable to junctions 5,9,10 in picture 1, and junction 24 in picture 3 is reported as matchable to junctions 5,6,16 in picture 1 and to junctions 1,7,11,17 in picture 2. Thirteen triples are formed (the subscript in the junction indicates the picture number)

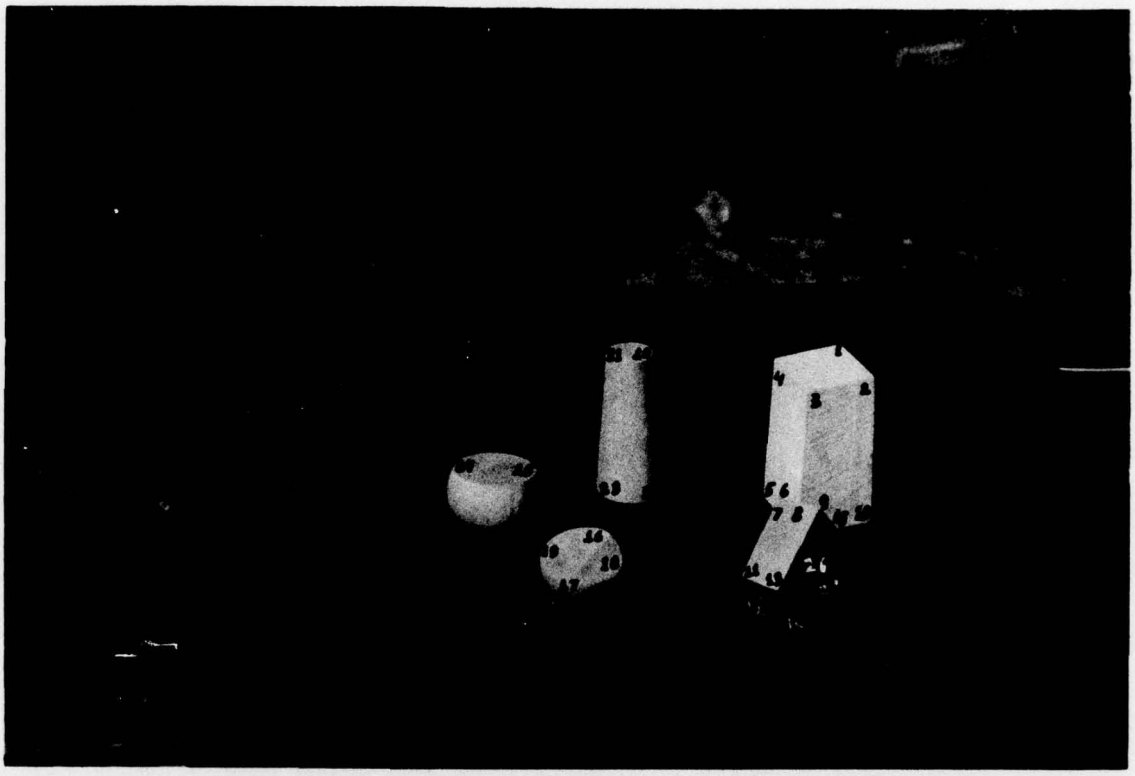


Fig. 3.6 Picture 1 of Illustrative Example

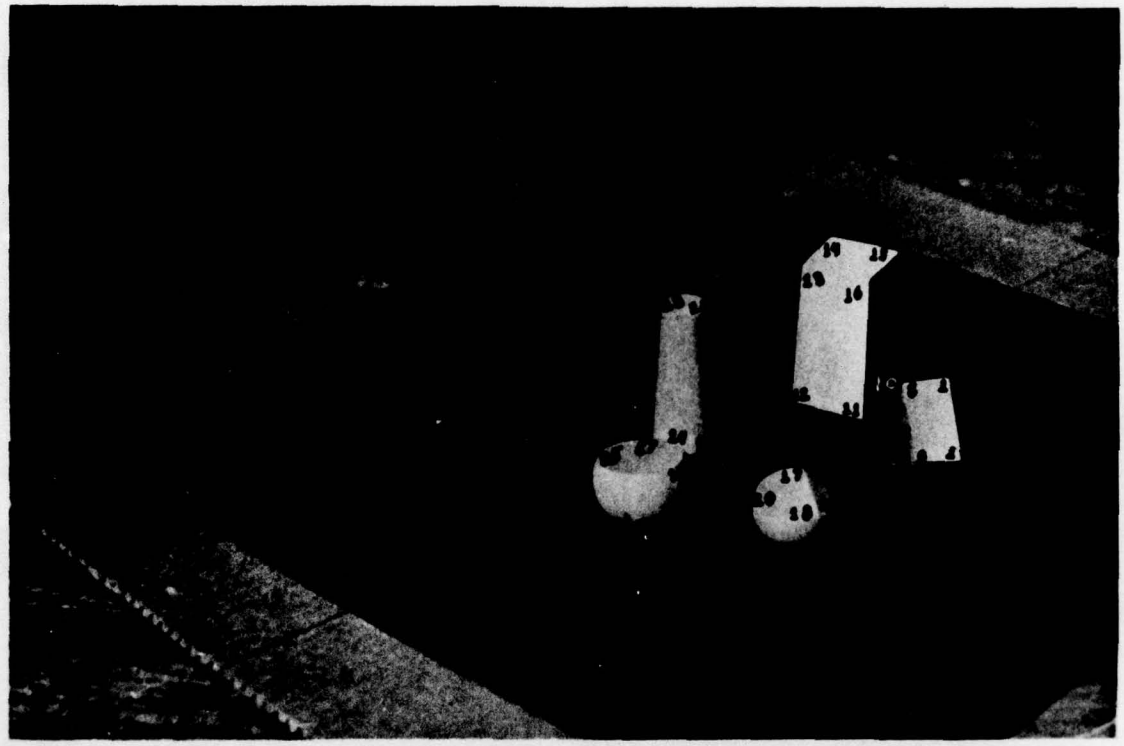


Fig. 3.7 Picture 2 of Illustrative Example

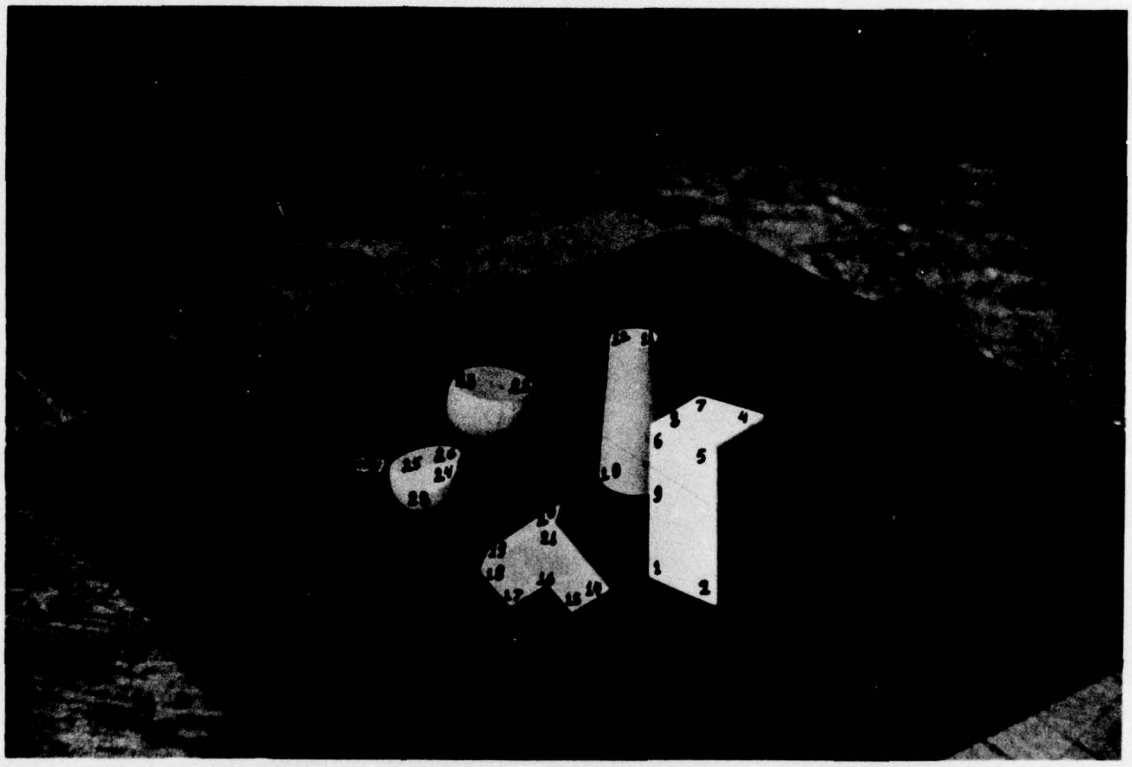


Fig. 3.8 Picture 3 of Illustrative Example

$(5_1,1_2,24_3)$	$(9_1,1_2,21_3)$	$(14_1,3_2,17_3)$
$(3_1,16_2,6_3)$	$(6_1,8_2,20_3)$	$(4_1,13_2,7_3)$
$(6_1,1_2,24_3)$	$(11_1,9_2,19_3)$	$(1_1,14_2,4_3)$
$(16_1,17_2,24_3)$	$(13_1,2_2,18_3)$	$(2_1,15_2,5_3)$
		$(10_1,10_2,2_3)$

On the basis of picture context the last nine triples can be accepted as being match triples. Of the four rejected triples, two are the three projections of two vertices, namely $(3_1,16_2,6_3)$ and $(16_1,17_2,24_3)$, but at this stage no context support is found. This is partly corrected in the next stage where we compare two pictures and establish matches between pairs of junctions. In the first pass we get the matches of the pairs $(16_1,17_2)$, $(17_1,18_2)$, $(12_1,4_2)$, $(3_1,16_2)$ and $(7_2,13_3)$. In the second pass we get $(5_1,12_2)$, $(11_2,1_3)$, $(16_3,26_1)$ and $(16_2,6_3)$. The last one gives, together with the pair $(3_1,16_2)$ reported in the first pass, the match triple $(3_1,16_2,6_3)$. Hence the following V junctions are marked as valid:

1_1, 3_1, 4_1, 5_1, 6_1, 10_1, 12_1, 17_1, 26_1

1_2, 3_2, 7_2, 10_2, 12_2, 13_2, 14_2, 16_2, 18_2

1_3, 5_3, 6_3, 7_3, 13_3, 16_3, 17_3, 18_3, 19_3, 21_3

No support was gained for the wrongly reported junctions

15_1, 24_2, and 27_2

nor for the (correctly reported) junctions

5_2, 6_2, 3_3 and 15_3.

IV. LINE MATCHING AND DATA RECOVERY

4.1 Introduction

Line matching is not a straightforward procedure. The fact that between two junctions we may have one, two, or three lines, complicates the situation, and it becomes even worse in the case of imperfect data. We have to be very careful in matching lines that all the possibilities are taken into consideration.

Line matching also serves as an important tool for recovering data. Data recovery consists basically of three different actions: (1) detection of a missing connection between two junctions, (2) extension of an existing line from one valid junction to another valid junction, and (3) detection of the location of a missing valid junction, validating the one found there (if any), or creating a synthetic junction.

Data recovery, if successful, leads to conditions for further line matching, which in turn enables more data recovery, and so on.

Cyclic order in an unordered valid junction is fixed if conditions for Rule 1 or Rule 2 exist (See Chapter II). Some previously unordered junctions can be ordered on the basis of line matching, as we shall see later. This in turn may create conditions for further application of Rule 1 or Rule 2, and so on.

Line matching, data recovery, and cyclic ordering is done in several passes. Each pass creates new conditions for the next pass until finally a condition is reached in which no new results can be obtained.

4.2 Fitting a Line to a Set of Points

It is important for the procedure to be able to determine the type of line represented by a set of points and sometimes to determine its exact equation as well.

Given a set of points (x_i, z_i), we try to fit a line to it so that the sum of the squared errors will be minimized. First we try to fit a straight line

$$a_2 x + a_1 z + a_0 = 0$$

where

$$a_0 = \Sigma x_i z_i \; \Sigma x_i z_i - \Sigma x_i^2 \; \Sigma z_i^2$$

$$a_1 = \Sigma x_i^2 \; \Sigma z_i - \Sigma x_i z_i \; \Sigma x_i$$

$$a_2 = \Sigma x_i \; \Sigma z_i^2 - \Sigma z_i \; \Sigma x_i z_i$$

The average distance of the points from the straight line is calculated and, if it is less than some threshold T_1, we accept the fit. The threshold is selected on the basis of visual judgment.

If the average distance is greater than some $T_2 > T_1$, then we declare the set as representing a curved line. For an average distance lying between these two values, we leave the type undetermined. If the set was declared to represent a curved line and we wish to fit a conic g to it,

$$g(x,z) \equiv a_5 x^2 + a_4 z^2 + a_3 xz + a_2 x + a_1 z + a_0 = 0$$

then by minimizing the sum of squared errors

$$\Sigma\, g^2(x_i, z_i) \equiv \underline{a}^T \cdot \underline{X} \cdot \underline{a}$$

and constraining a_0 to equal 1, we can solve for the rest of the coefficients using $\underline{X} \cdot \underline{a} = 0$. Here $\underline{a}$ is the vector of coefficients and

$$\underline{X} = \begin{bmatrix} \Sigma x_i^4 & \Sigma x_i^2 z_i^2 & \Sigma x_i^3 z_i & \Sigma x_i^3 & \Sigma x_i^2 z_i & \Sigma x_i^2 \\ \Sigma x_i^2 z_i^2 & \Sigma z_i^4 & \Sigma x_i z_i^3 & \Sigma x_i z_i^2 & \Sigma z_i^3 & \Sigma z_i^2 \\ \Sigma x_i^3 z_i & \Sigma x_i z_i^3 & \Sigma x_i^2 z_i^2 & \Sigma x_i^2 z_i & \Sigma x_i z_i^2 & \Sigma x_i z_i \\ \Sigma x_i^3 & \Sigma x_i z_i^2 & \Sigma x_i^2 z_i & \Sigma x_i^2 & \Sigma x_i z_i & \Sigma x_i \\ \Sigma x_i^2 z_i & \Sigma z_i^3 & \Sigma x_i z_i^2 & \Sigma x_i z_i & \Sigma z_i^2 & \Sigma z_i \\ \Sigma x_i^2 & \Sigma z_i^2 & \Sigma x_i z_i & \Sigma x_i & \Sigma z_i & \Sigma 1 \end{bmatrix} \tag{9}$$

The foregoing methods have their drawbacks (11,16,1) but they are good enough for the purpose here.

4.3 Line Matching

The matching of lines is done around matched junctions, taking two pictures at a time. Two lines in two pictures, matching the same line in a third picture, are declared to match each other. With every junction there are associated three indices 1,2, and 3 for the three possible lines that share the junction. (If the junction is cyclically ordered, these indices coincide with the cyclic order.) We say that the junction

is linked in an index t if there is a line corresponding to that index whose other end junction is a valid junction.

The two matched junctions around which we try to match lines fall into the following three cases:

(1) both contain three lines,

(2) one contains three lines and the other contains two lines, or

(3) both contain two lines.

In each of these three cases when one line in one junction is matched to a line in the other junction and both junctions are ordered cyclically, further line matching can be done by proceeding cyclically around the junctions.

At the beginning of the line matching process all the 3L (3 lines) junctions are cyclically ordered according to their natural order, and, therefore, in Case 1, matching one line forces the match of the other two lines. At a later step in the procedure a line may be added to an unordered 2L junction, yielding an unordered 3L junction; then matching one line in a Case-1 situation will not automatically lead to the match of the rest.

Fig. 4.1 shows under what conditions we can match a line in Case 1. The basic idea is to find out whether between a pair of junctions in the same picture there is a single line, or, if more than one, to determine their relations to each other. For example in configuration (a) of Fig. 4.1 it is clear that line c_i must be the only line between junctions I_i and J_i because J_i is linked in its other two indices. It follows that the line d_j is the only line between junctions I_j and J_j

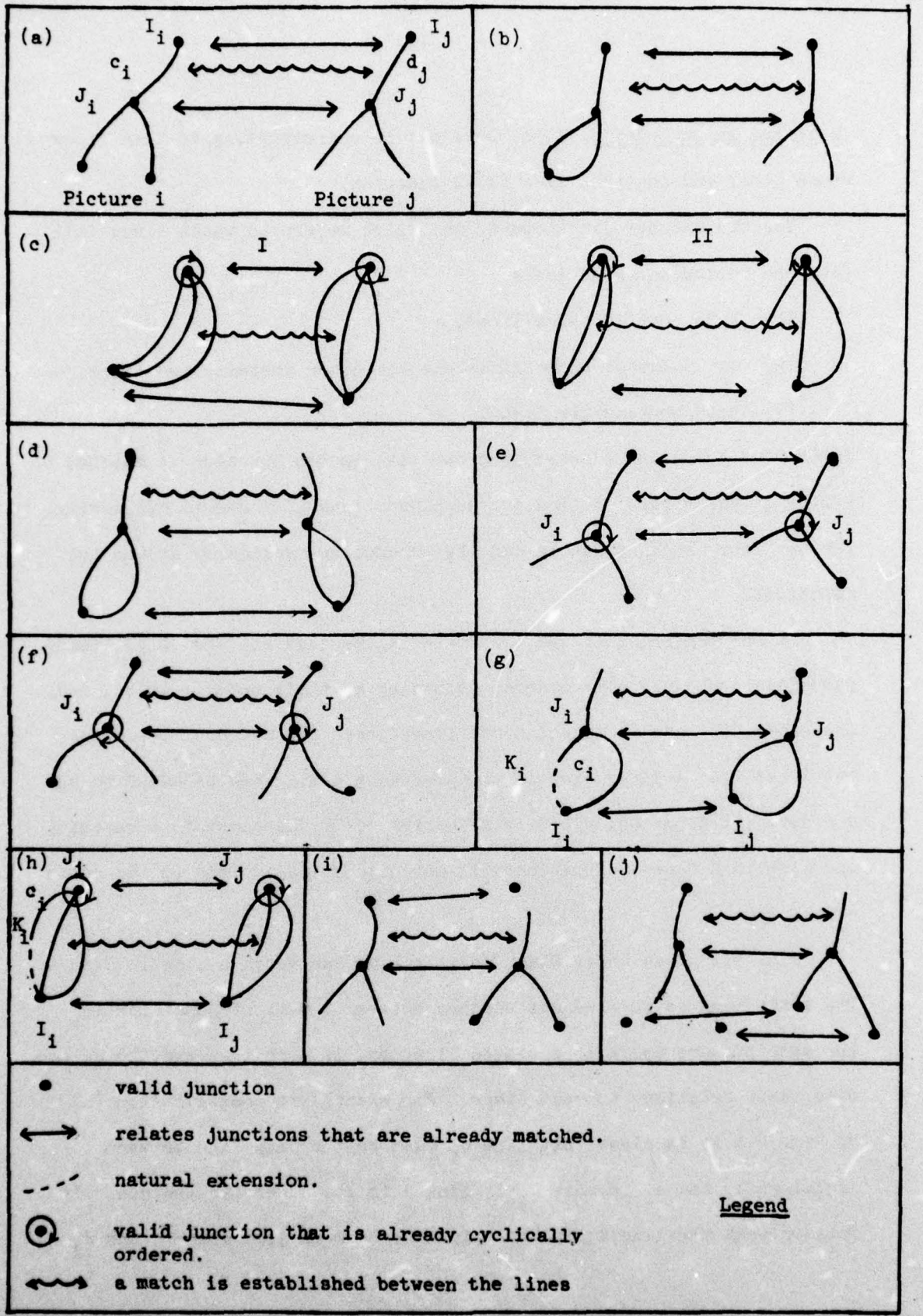


Fig. 4.1 Case-1 Match Configurations (3-Line Junctions)

considering the pairwise matching of the junctions (though it might not be clear from looking at picture j alone when J_j is not linked in its two other indices). Thus the result is a match between lines c_i and d_j. Part of the requirements in configurations (c), (e), (f) and (h) is that the junctions J_i and J_j be cyclically ordered to establish even the first match. Configurations (g), and (h) require some special condition, namely that I_i lie on the natural extension of the line c_i. This means that we must successfully fit a line to I_i and the points of c_i and that this line comply with the conditions stated in the following paragraph.

As explained in section 4.2 a straight line fit is acceptable if the average miss distance is less than some threshold. If a straight line fit is rejected and a conic is fitted instead, we get either an ellipse, a hyperbola or no locus at all. Other types of conics are not realistic since they involve a vanishing parameter, which in practice does not occur in the computation because of truncation errors. No locus at all means of course that a conic fit has been rejected. We also reject any fit accepted thus far if (1) the distance between I_i and K_i is greater than the distance between J_i and I_i (K_i is the non-valid end junction of c_i.), (2) if it is an ellipse and one of the axes is too small (to avoid a fit "jumping" from one side of the ellipse to the other), (3) if it is an ellipse and the part of the ellipse between K_i and I_i is greater than a quarter of the ellipse, and (4) if it is a hyperbola and the points of the set belong to the two branches of the hyperbola. This tightening of acceptance conditions

very much reduces the possibility of a wrong extension, but at the same time it also rules out some good extensions, especially under condition (3).

Configurations c and h present a special problem since they both deal with junctions connected by three lines. To obtain matches for them we assume that the middle line stays in the middle. Although true in most cases, it is easy to show counterexamples. But even then it is not crucial since the configuration is closed on itself and the error does not propagate. To find the middle line we note that when we walk from one junction to the other and back, changing lines in increasing cyclic order, we trace three closed paths. In two of them the path is traced in a counterclockwise manner (Fig. 4.2), and in one of them in a clockwise manner. The last one happens to be the path where the middle line is not involved. Using this fact the middle lines in both pictures can be detected and matched to each other. The rest follow cyclically.

Fig. 4.3 shows the configurations required for matching lines in Case 2. Note configurations (c) and (j) where a natural extension is a part of the conditions. Note also configurations (d), (k), (e), and (i), which imply that junctions J_i and I_i in each should be connected. We shall return to this subject in the next section.

Fig. 4.4 shows the conditions under which we match lines in case 3. Note in both configurations the implication of a missing connection, between J_i and I_i and between J_j and K_j in (a), and between J_j and I_j in (b). In Figs. 4.1 and 4.3 the shape of the three-lines junctions, (whether Y or W) is irrelevant to the configurations.

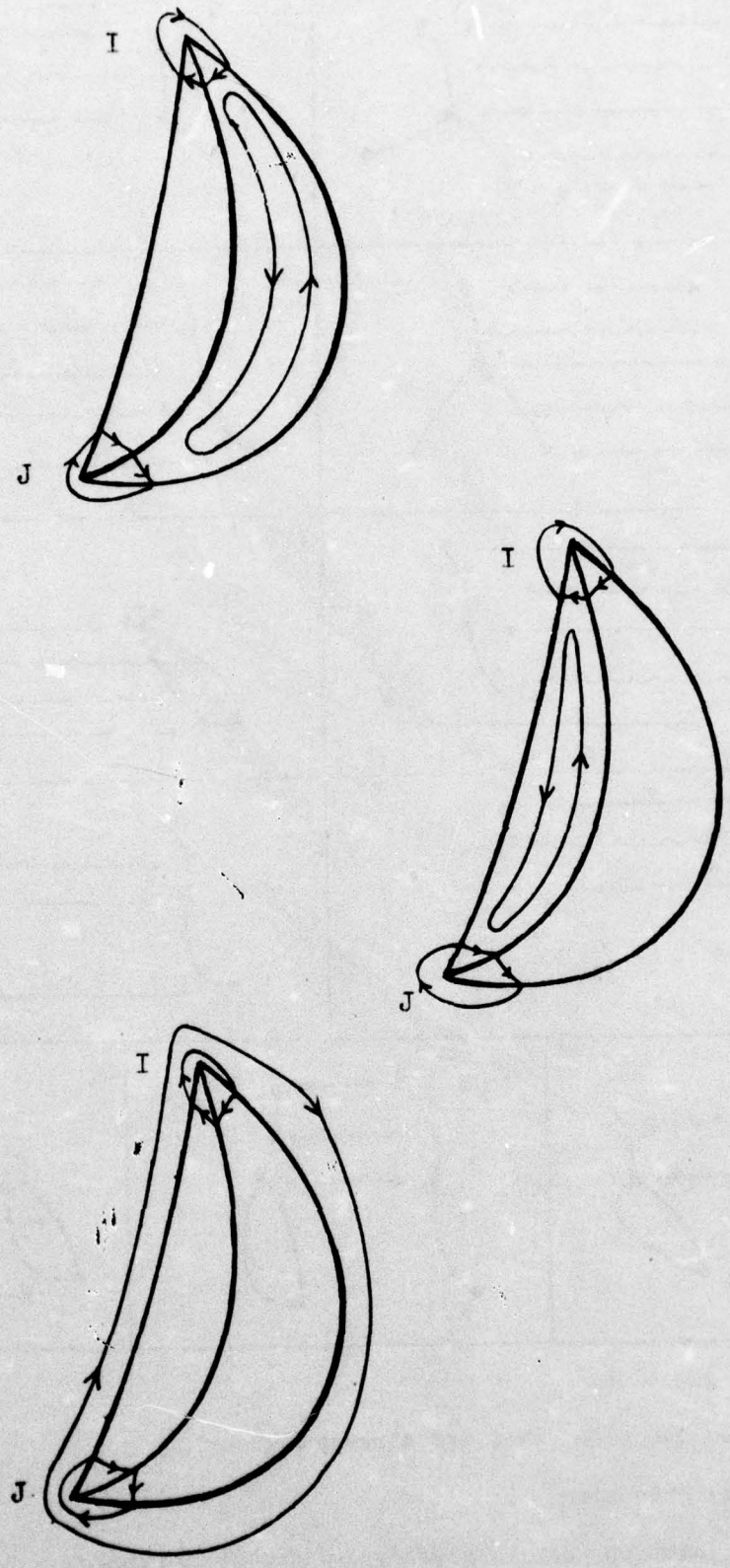


Fig. 4.2 Matching of Middle Lines in Configurations Involving Junctions Connected by Three Lines

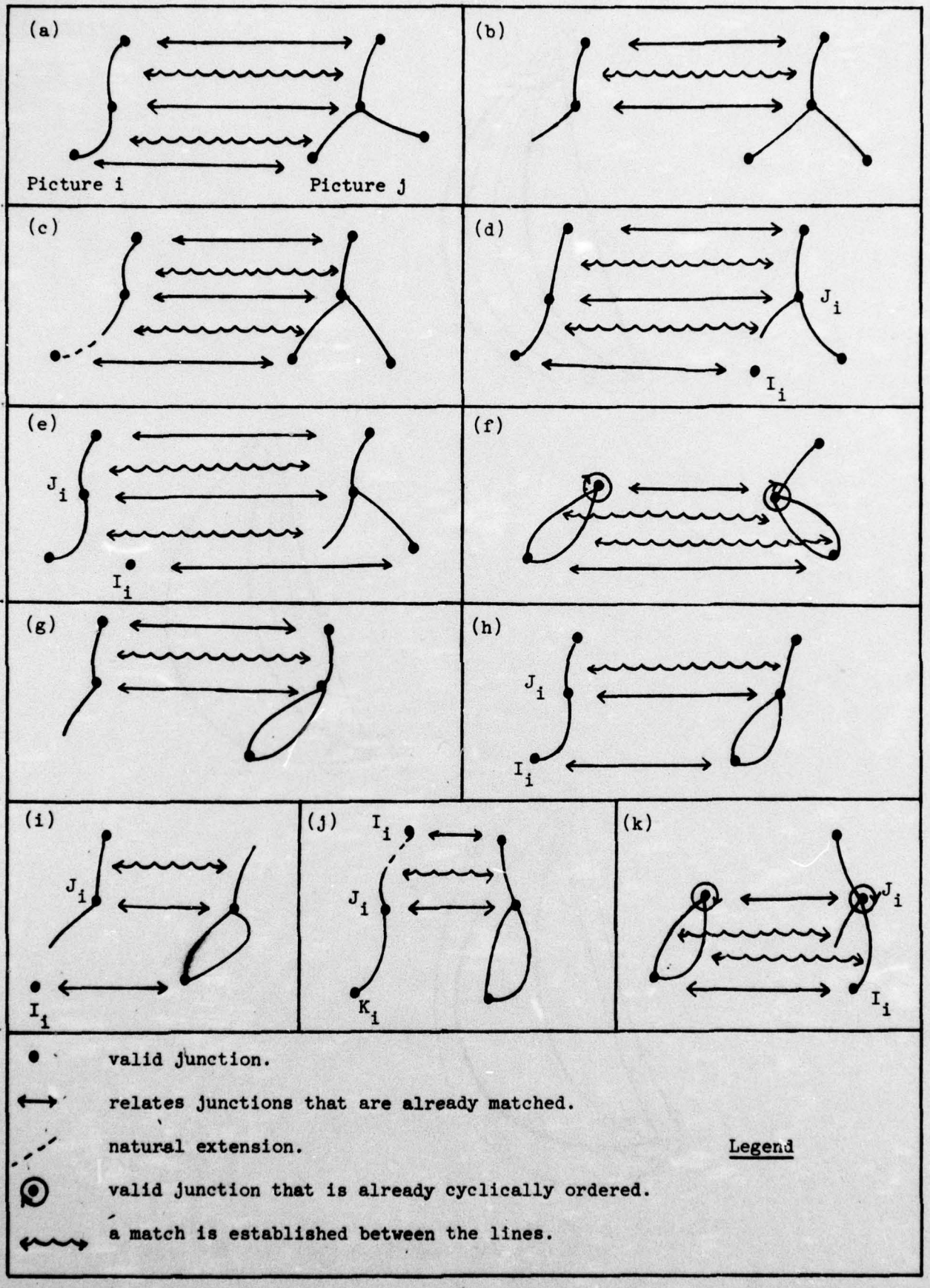


Fig. 4.3 Case-2 Match Configuration (One 3-Line and One 2-Line Junction)

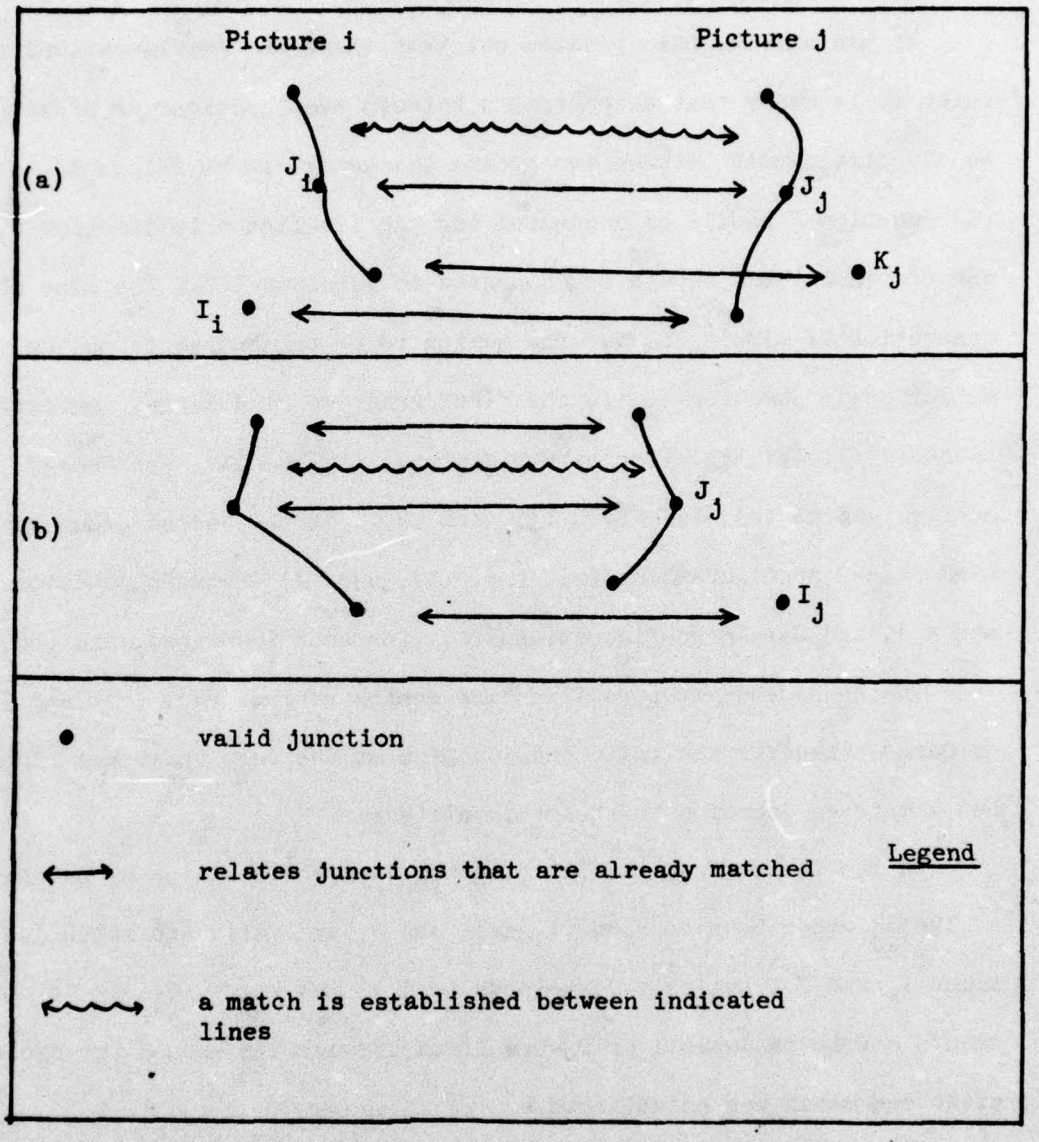


Fig. 4.4 Case-3 Match Configuration (Two 2-Line Junctions)

4.4 Data Recovery

It has already been pointed out that there are configurations for which it is clear that a connection between two junctions is missing. We can distinguish between two groups characterized as follows: (1) junction J should be connected through its line c to junction I, and (2) junction J should be connected to junction I but the line of connection is missing in J. The action to be taken depends on the situation in junction I. In the first group we find Case-1 configurations (c)II, (g), (h) (for both pictures), (i) and (j), and Case-2 configurations (c), (d), (i), (j), and (k). In the second group we find Case-2 configurations (e), (h), (i), and (j) (between junctions J_i and K_i), and Case-3 configurations (a) (for both pictures), and (b).

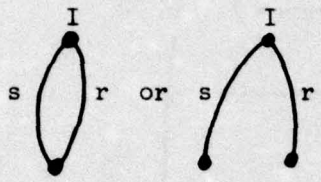
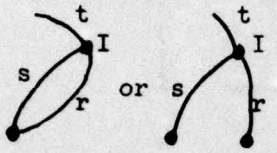
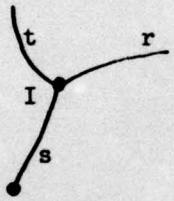
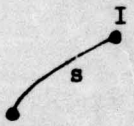
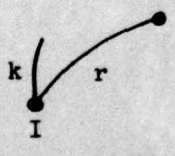
In the second group we find also configurations (a), (c), and (d), of Case-2 whenever the valid end junction of the only unmatched line has a matched junction in the other picture.

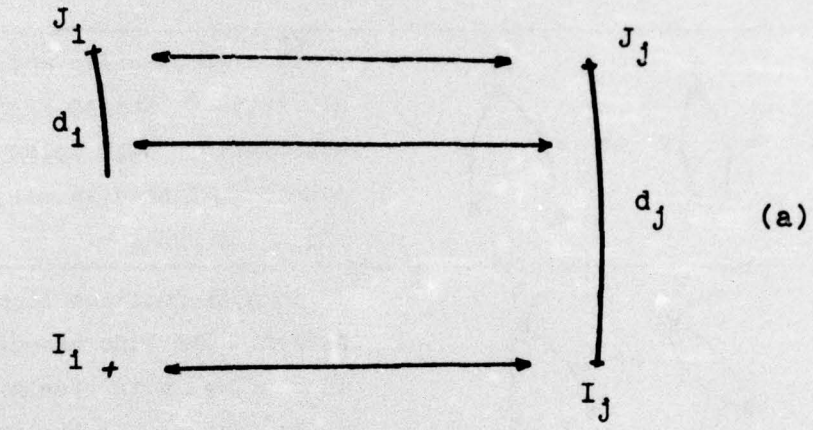
In the first group we also find the case where a line d_i matches a line d_j whose both end junctions I_j and J_j are valid and match junctions I_i and J_i, but only J_i belongs to d_i. See Fig. 4.5(a). It occurs mostly in Cases 1 or 2 when lines are matched around the cycle, after one match was established.

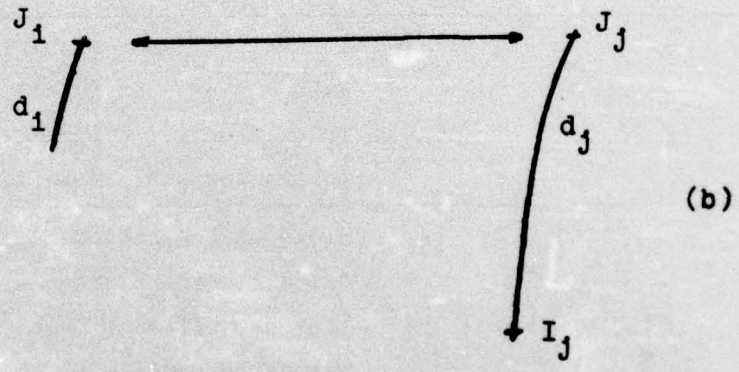
Table 4.1 shows us the different possibilities and appropriate actions taken for the first group and Table 4.2 does the same for the second group. In both cases it is assumed that the index in J with which we associate the line of connection is r. From the nature of the different configurations, r is always known. The problem is to determine

Table 4.1

Conditions for First-Group Junction Connections

	I is a 2L junction and linked in indices r and s. Action: Extend line c to junction I and associate it with index t in I. I becomes a 3L junction, linked in all three indices. J is linked in index r.
	I is a 3L junction linked in indices r and s. Action: The line associated with index t in I is combined with line c. I becomes linked in three indices. J is linked in index r.
	I is a 3L junction linked in index s. Action: The average error of points on the line associated with r and t is calculated with respect to the natural extension of c. The line with the smaller average error is combined with c. I becomes linked in a second index. J becomes linked in r.
	I is a 1L junction linked in index s. Action: Extend line c to junction I and associate it with index t or r. I becomes a 2L junction, linked in two indices. J is linked in index r.
	I is a 2L junction linked in index r. Action: Calculate the average error of the points on the line associated with t, with respect to the natural extension of c. If the error is less than some threshold combine the two lines. I becomes linked in t and I in r.
Other	No Action





Picture i Picture j

Fig. 4.5 Data Recovery Situation

with which index in I to associate this line. The possibility of having a 1L valid junction will be demonstrated later in this section. It is a result of certain data recovery actions.

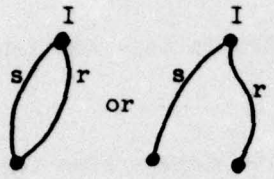
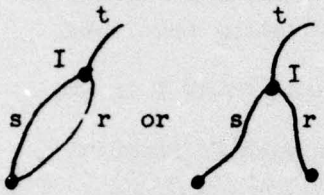
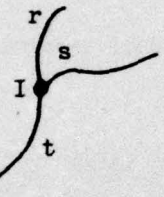
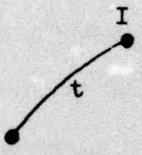
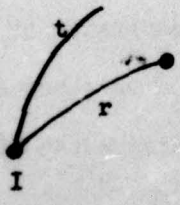
By "empty" line mentioned twice in Table 4.2, we mean a line for which only the end points are known (the two junctions) and nothing is known about its nature. Its importance lies in creating the links, which in turn create new conditions for matching lines around I in the next pass. The linking of I and J may, for example, qualify junction I for analysis as one of the Case-1 configurations and may enable the application of Rules 1 and 2. We shall return to the detection and completion of missing connections once again in what we call second-level data recovery.

Let us consider the situation shown in Fig. 4.5(b). Lines d_i and d_j either match each other or are candidates for a match. But there is no match in picture i for junction I_j. This might be crucial for the matching decision for the lines d_i and d_j as we can see by comparing Fig. 4.3(b) with Fig. 4.3(c). If there is for I_j a matching junction I_k in the third picture, it could serve as a basis for further tests and data recovery. We may then have two different scenarios:

1. There exist a junction I_i in picture i, unmatched as yet, that is matchable to I_j and to I_k. If I_i lies on the natural extension of d_i, then (I_i, I_j, I_k) is declared as a match triple, and we conclude that the junctions J_i and I_i should be connected. We also match the lines d_i and d_j if they are not matched as yet.

TABLE 4.2

Conditions for Second-Group Junction Connections

	I is a 2L junction linked in r and s. Action: Create a new "empty" line between I and J. Associate it with index t in I and index r in J. I becomes a 3L junction linked in all indices. The number of lines in J is increased by one and J is linked in index r.
	I is a 3L junction linked in 2 indices r and s. Action: Extend the line associated with t from I to J and associate it with index r in J. I is linked now in three indices. The number of lines in J is increased by 1. J is linked in index r.
	I is a 3L junction linked in index t. Action: Find which of the two lines associated with indices r and s has a natural extension to J. If there is one, extend the corresponding line to J. I is linked then in two indices. Number of lines in J is increased by 1, and it is linked in index r.
	I is a 1L junction linked in index t. Action: Create an "empty" line between I and J. Associate the line with index r in J and with one of the indices r or s in I. I becomes a 2L junction linked in two indices. The number of lines in J is increased by one, and J is linked in index r.
	I is a 2L junction linked in index r. Action: Check whether J is on the natural extension of the line associated with t. If so, extend this line to J. I is now linked in two indices. The number of lines in J is increased by 1 and it is linked in its index r.
Other	No Action

2. If there is no junction I_i that is matchable to I_j and I_k, (and has not been matched as yet) then we build a synthetic junction. This is done as follows: From I_j and I_k we can determine the approximate location in 3D of the corresponding vertex. It will be the point with the minimal sum of absolute distances from the projecting lines $C_k I_k$ and $C_j I_j$. This point is found most easily by looking at the x, y coordinates of the intersection of the lines' projections on an x-y plane in a coordinate system whose z axis is normal to both lines. The calculated projection, from C_i on picture i, of this vertex will serve as the synthetic junction I_i. If the lines d_i and d_j are not matched as yet but the synthetic junction I_i lies on the natural extension of d_i, or if d_i and d_j already match, then we accept the synthetic junction as a new valid junction, declare the triple (I_i, I_j, I_k) as a match triple, and extend the line d_i to I_i. I_i is now a valid 1L junction linked in one index.

To demonstrate how important a synthetic-junction formation can be for line matching, let us look at the three images shown in Fig. 4.6(a). Fig. 4.6(b) shows the line matching that can be established before the formation of the synthetic junction 4_i. Fig. 4.6(c) shows the line matching established by the configuration of Fig. 4.1(i), possible only after 4_i was formed.

Finally, an unordered junction for which Rules 1 and 2 cannot be applied, can be ordered if at least two of its lines are matched to two lines of an ordered junction so that the cyclic precedence of the first two will agree with that of the last two. The indices associated with the formerly unordered junction are then adjusted to comply with the cyclic order.

4.5 Example

We shall describe some of the results obtained when a program for the foregoing procedure was tested on the scene shown in Fig's. 3.6, 3.7 and 3.8. In Fig. 4.7 we give a schematic description of the three pictures. First the unordered junctions 6_1, 12_1, 17_1, 1_2, 3_2, and 18_2 were ordered each by applying Rule 1.

The configuration of Fig. 4.3(a) in 14_1 and 3_2 brought the match of lines $\overline{12_1, 14_1}$ to $\overline{4_2, 3_2}$ and $\overline{13_1, 14_1}$ to $\overline{3_2, 2_2}$.

The configuration of Fig. 4.3(a) in 14_1 and 17_3 brought the match lines $\overline{14_1, 13_1}$ to $\overline{17_3, 18_3}$ and $\overline{14_1, 26_1}$ to $\overline{17_3, 16_3}$. Combining this with the preceding match we get the match $\overline{17_3, 18_3}$ to $\overline{3_2, 2_2}$. Through this match we are able also to arrange the cyclic order in junction 17_3.

The configuration of Fig. 4.1(a) in 11_1 and 9_2 gave the match of lines $\overline{11_1, 6_1}$ to $\overline{9_2, 8_2}$, $\overline{11_1, 13_1}$ to $\overline{9_2, 2_2}$ and $\overline{11_1, 12_1}$ to $\overline{9_2, 4_2}$.

The configuration of Fig. 4.3(a) in 11_1 and 19_3 gave the match of lines $\overline{11_1, 6_1}$ to $\overline{19_3, 20_3}$ which will match $\overline{9_2, 8_2}$ - combining the above - and $\overline{11_1, 13_1}$ to $\overline{19_3, 18_3}$ which will match $\overline{9_2, 6_2}$. The result of the matching of the last group is that we can now establish the cyclic order in junction 19_3.

Also $\overline{6_1, 9_1}$ match $\overline{8_2, 1_2}$ based on the configuration of Fig. 4.3(a) in 6_1 and 8_2 ; $\overline{6_1, 9_1}$ match $\overline{20_3, 21_3}$ on basis of Fig. 4.3(a) in 6_1 and 20_3 ; and combining the two we get that $\overline{20_3, 21_3}$ matches $\overline{8_2, 1_2}$.

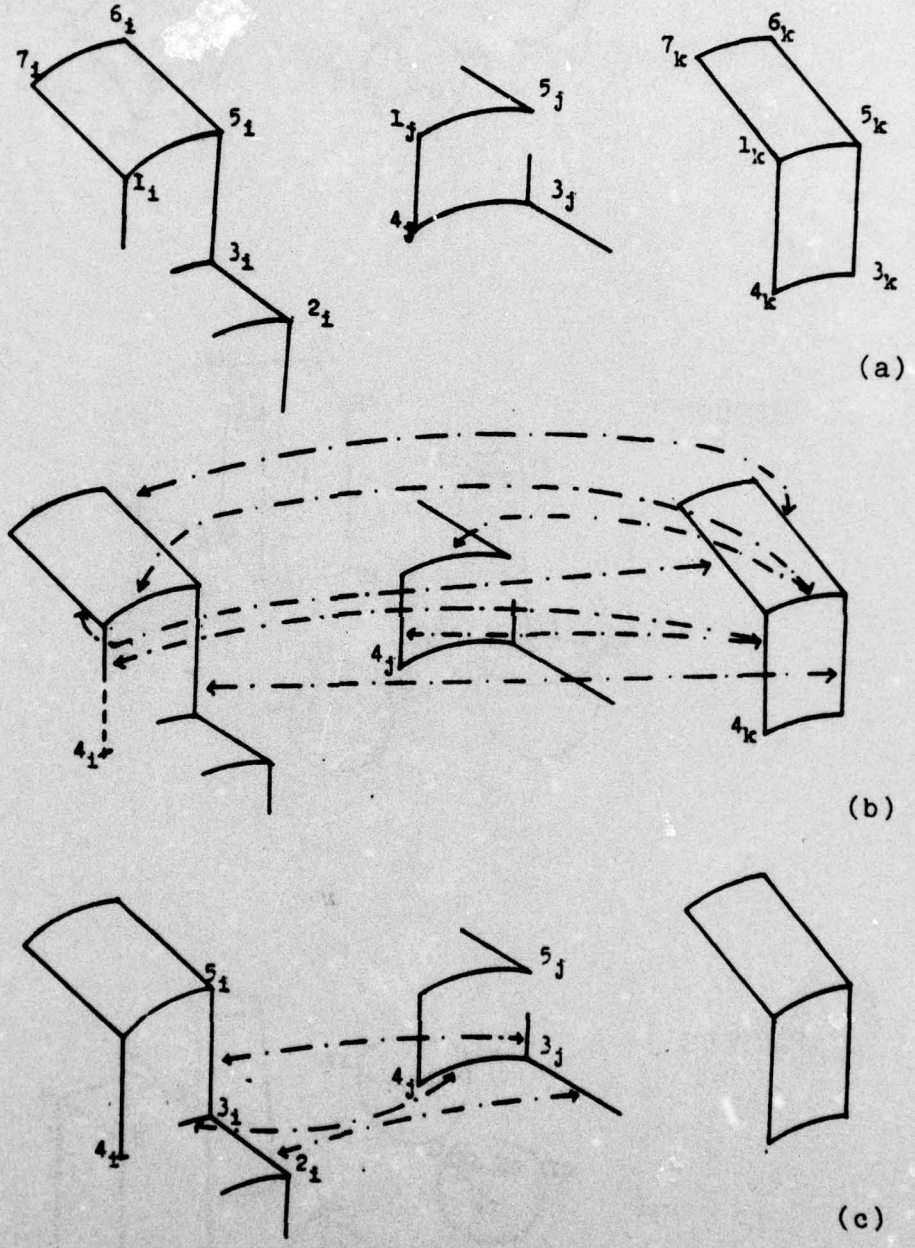
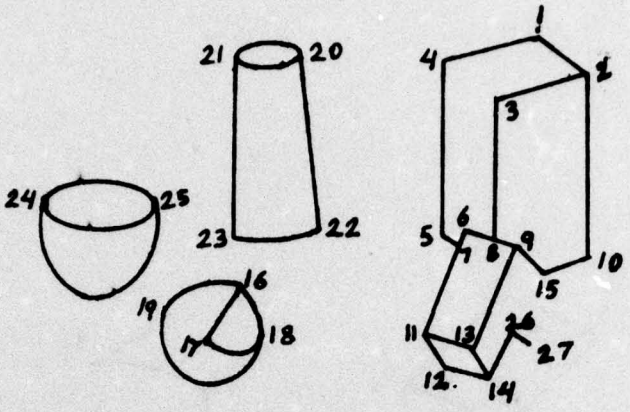
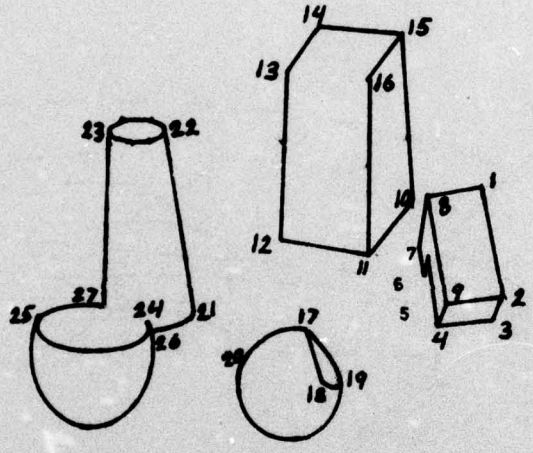


Fig. 4.6 Use of a Synthetic Junction

PICTURE 1



PICTURE 2



PICTURE 3

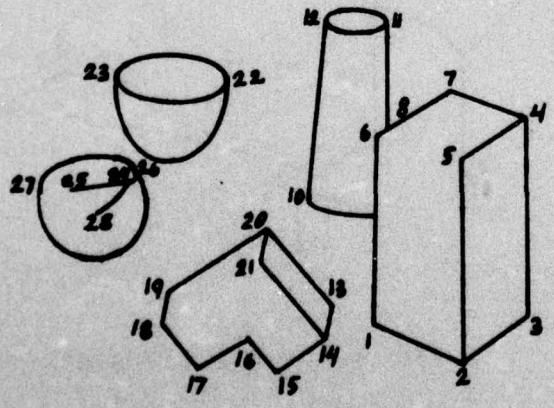


Fig. 4.7 Schematic Description of the Pictures Shown in Fig's. 3.6, 3.7 and 3.8.

From Fig. 4.1(a) in 13_1 and 2_2 we obtained the match of $\overline{9_1, 13_1}$ to $\overline{2_2, 1_2}$. From Fig. 4.3(a) in 13_1 and 18_3 and a match between junctions 9_1 and 21_3 we obtain the linkage of junctions 18_3 and 21_3 by an "empty" line, which was matched to line $\overline{13_1, 9_1}$ and thus to line $\overline{2_2, 1_2}$. Junction 18_3 was arranged cyclically by comparison with junction 13_1, and this enabled the application of Rule 1 for ordering junction 21_3; this in turn enabled the application of Rule 1 for ordering junction 13_3.

Next we obtained also the match of line $\overline{9_1, 15_1}$ to $\overline{21_3, 14_3}$ from the just formed Fig. 4.1(a) configuration in junctions 9_1 and 21_3.

From 4.3(a) in 1_1 and 4_3 we found the match of $\overline{1_1, 4_1}$ to $\overline{4_3, 7_3}$. The Fig. 4.1(a) configuration in 2_1 and 15_2 gave the match of $\overline{2_1, 1_1}$ to $\overline{15_2, 14_2}$, $\overline{2_1, 10_1}$ to $\overline{15_2, 10_2}$ and $\overline{2_1, 3_1}$ to $\overline{15_2, 16_2}$. The configuration of Fig. 4.3(a) in 2_1 and 5_3, together with the match of 3_1 and 6_3 brought the formation of an "empty" line between 5_3 and 6_3, its match to line $\overline{2_1, 3_1}$ (and thus to $\overline{15_2, 16_2}$), the match of $\overline{5_3, 2_3}$ to $\overline{2_1, 10_1}$ (and thus to $\overline{15_2, 10_2}$), the match of $\overline{5_3, 4_3}$ to $\overline{2_1, 1_1}$ (and thus to $\overline{15_2, 14_2}$), and the cyclic ordering of junction 5_3.

Line $\overline{8_2, 7_2}$ was matched to $\overline{20_3, 13_3}$ using the Fig. 4.1(a) configuration in 8_2 and 20_3. Line $\overline{11_2, 10_2}$ was matched to $\overline{1_3, 2_3}$, and $\overline{11_2, 16_2}$ to $\overline{1_3, 6_3}$ using the Fig. 4.3(a) configuration in 11_2 and 1_3. Junction 1_3 is ordered by comparing it with junction 11_2. This, together with the last paragraph results, permits the ordering of 6_3 by Rule 1. The configuration of Fig. 4.4(b) in 13_2 and 7_3 permits 13_2 and 16_2 to be connected by an "empty" line which is matched to line $\overline{7_3, 6_3}$; line $\overline{13_2, 14_2}$ is matched to $\overline{7_3, 4_3}$ and thus also to line $\overline{1_1, 4_1}$.

In junctions 4_1 and 13_2 the configuration of Fig. 4.3(a) was formed only after the connecting of junctions 13_2 and 16_2. Otherwise no qualified configuration could have been found there. This configuration brings the match of line $\overline{4_1, 5_1}$ to line $\overline{13_2, 12_2}$.

We also tested for the configuration of Fig. 4.1(h) in junctions 16_1 and 17_2. (In the beginning of the procedure we merge the two non-limb lines in every S junction into one line.) The extension of line $\overline{17_2, 20_2}$ to junction 18_2 was rejected on the basis of condition* 3. But a natural extension of line $\overline{16_1, 19_1}$ to junction 17_1 was found, qualifying the configuration. Thus line $\overline{16_1, 17_1}$ was matched to line $\overline{17_2, 18_2}$, line $\overline{16_1, 19_1}$ was extended to junction 17_1 and matched to line $\overline{17_2, 20_2}$, which is now extended on the basis of the match to junction 18_2. Finally, line $\overline{16_1, 18_1, 17_1}$ was matched to line $\overline{17_2, 19_2, 18_2}$.

These matches were reached before making use of any synthetic junctions. Let us see how the matching ability is improved by them. As we can see, line $\overline{3_1, 8_1}$ could not be matched. If the configuration found in 3_1 and 6_3 (Fig. 4.3(b)) could be modified to the configuration of Fig. 4.3(c) then the desired match could be established. (Junction 4_1 failed to be on the natural extension of $\overline{3_1, 8_1}$). Hence, junctions 11_2 and 1_3 are used to create a synthetic junction in picture 1 which we denote as 28_1. Junction 28_1 is found to be on the natural extension of $\overline{3_1, 8_1}$ and thus is accepted as a valid junction. Line $\overline{3_1, 8_1}$ is extended

---

* See discussion concerning natural extension in Section 4.3

to be line $\overline{3_1, 28_1}$ and the triple $(28_1, 11_2, 1_3)$ is declared to be a match triple.

The configuration of junctions 11_2 and 1_3 force the connections of junction 28_1. Thus junction 28_1 should be connected to junctions 10_1 and 5_1 under the rules of Table 4.2. It is found to be on the natural extension of $\overline{10_1, 15_1}$ which is thus extended to be line $\overline{10_1, 28_1}$. It is also found to be on the natural extension of line $\overline{5_1, 7_1}$ which turns to be line $\overline{5_1, 28_1}$. The new configuration created now in junctions 3_1 and and 6_3 (of form Fig. 4.3(e)), causes the match of $\overline{3_1, 28_1}$ to $\overline{6_3, 1_3}$ and thus also to $\overline{16_2, 11_2}$, as well as the connection with an "empty" line of 3_1 to 4_1 (and its match to $\overline{6_3, 7_3}$ and thus also to the "empty" line $\overline{13_2, 16_2}$). We also get the match of $\overline{10_1, 28_1}$ to $\overline{1_3, 2_3}$ (configuration of Fig. 4.3(a) in 28_1 and 1_3) and thus also to $\overline{10_2, 11_2}$. In addition we get the match of $\overline{5_1, 28_1}$ to $\overline{12_2, 11_2}$ (configuration of Fig. 4.1(a) in 28_1 and 11_2), and finally the following junctions are ordered cyclically as the result of all this additional matching and linking: 1_1, 3_1, 4_1, 5_1, 10_1, 28_1, 10_2, 12_2, 13_2, 14_2, 16_2 and 7_3.

4.6 Matching Lines by Range

The technique of matching lines around matched junctions cannot be applied when the lines are closed, when they have end junctions that are both of type A, when the ends are both open, or when one end junction is of type A and the other end is open. Additional tools are needed for establishing a match in these situations. We define for a line g_i in picture i range limits with respect to picture j as the two straight lines,

with the maximum angle of opening, going through C_{ij} and having points in common with g_i. (Lines a and b in Fig. 4.8). For a closed line the range limits are the two tangents to the line from C_{ij}.

Taking two pictures i and j, we create in each picture, for all qualified lines, pairs of range limits with respect to the other picture. If we now project both pictures on a plane P_q (as explained in Chapter III and shown in Fig. 3.2) we get all the range limits from both pictures as a set of lines going through C_{ijq}. If we take the range limits of a line g_i, then when projected on P_q, they form an angle α. The range limits of the line h_j when reprojected on P_q form the angle β. We define a ratio $R_t = \frac{\gamma}{\alpha+\beta}$ where γ is the overlap angle (See Fig. 4.8). When R_t is bigger than some threshold, we say that g_i and h_j are matchable by range (but are not matched yet). When we have three lines in three pictures, lines g_i, h_j and f_n, pairwise matchable by range, then we say that the lines are <u>matched</u> by range. In our example the lines $\overline{23_1, 22_1}$, $\overline{21_2, 26_2}$ and $\overline{10_3, 9_3}$, among others, are matched by range.

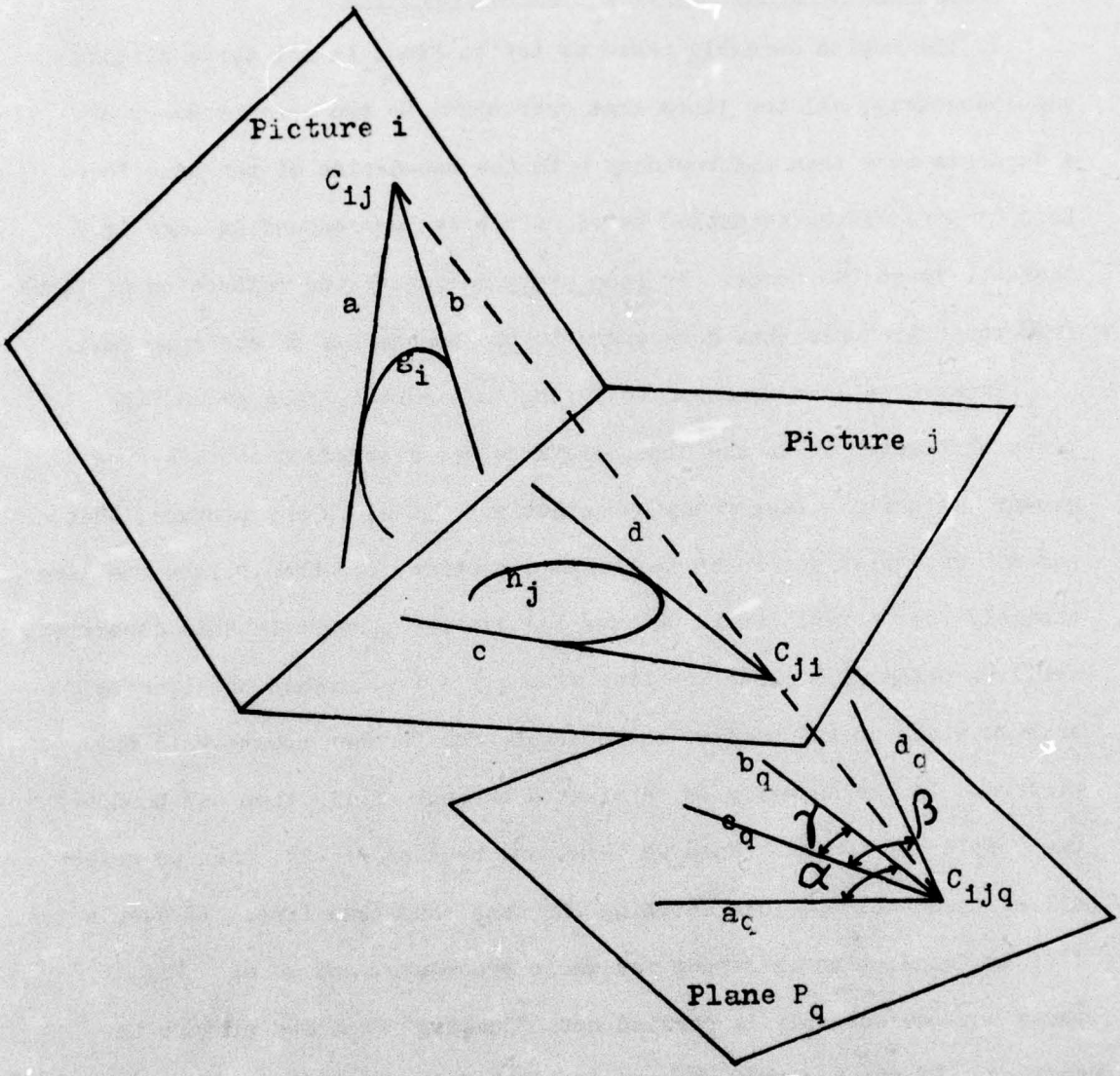


Fig. 4.8 Matching Lines by Range

V. FINAL DESCRIPTION

5.1 Region Assembly and Second-Level Data Recovery

In the region assembly phase we try to find, in all three pictures simultaneously, all the lines that correspond to the same boundary or - if a face has more than one boundary - to the boundaries of the same face. Each line should be assembled twice, since its corresponding edge is shared between two faces. By face group we denote the collection of lines from three pictures that correspond to the boundaries of the same face.

Whenever a line is found to belong to a certain face group, the lines that match it in the other pictures are also added to this face group. We build a face group by selecting a line, in any picture, that was not assembled yet in at least one direction. We then follow the line assembly (see definition in Chapter II), in the picture in this direction, until we either (1) close the line assembly - i.e., reach the line we started with, or (2) no line can be found for further assembly in this picture. If the assembly is terminated because of (1) then one boundary was completely traced. When we terminate because of (2), then we select a line in another picture matching the last assembled line. If such a line is found, then we repeat the whole procedure, and so on. Fig. 5.1 shows how the assembly is carried out, "jumping" from one picture to another. If the assembly process is blocked in all three pictures, we return to the starting point and trace the lines in the opposite direction, changing lines at the junctions in decreasing cyclic order and following the above-described procedure until we are blocked again in all three pictures. Although we are now tracing the lines in the opposite direction,

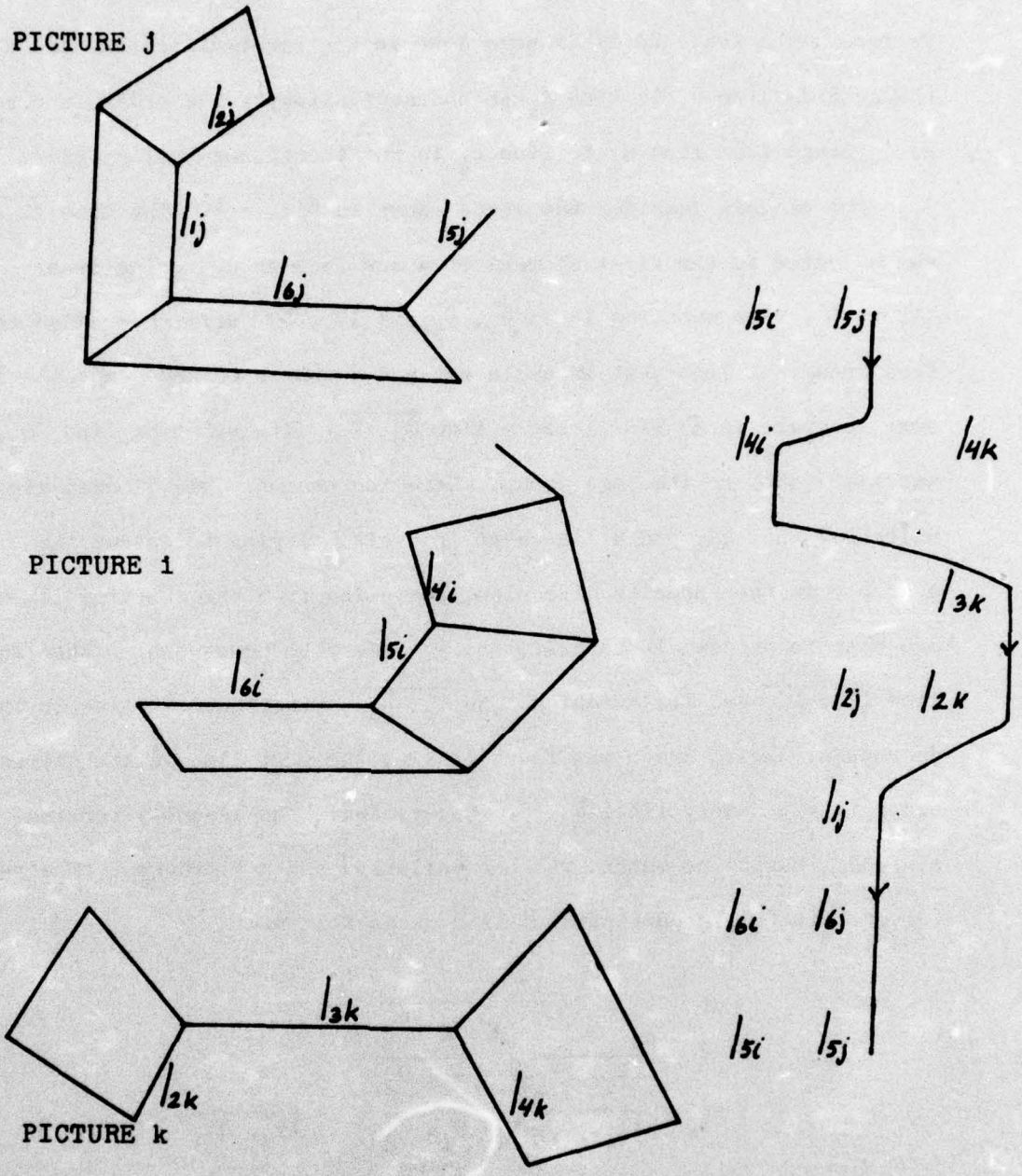


Fig. 5.1 Face Group Assembly

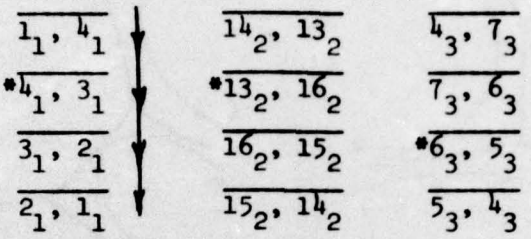
we record the trace as if it were done in the right direction, and a change from line c_i to line d_i in the decreasing cyclic order is recorded as a change from line d_i to line c_i in the increasing cyclic order.

For example consider the scene shown in Fig. 4.7. The line $\overline{11_1, 6_1}$ was selected as the first element of a new face group, going from 11_1 to 6_1. Its matching lines $\overline{9_2, 8_2}$ and $\overline{19_3, 20_3}$ were then added to the face group. A line next in cycle was not found in 6_1 and hence the line next in cycle in 8_2 was picked - line $\overline{8_2, 7_2}$. Its matching line $\overline{20_3, 13_3}$ was also added to the face group. Here the assembly was blocked since neither 7_2 nor 13_3 had a line next in cycle. Trying to extend the assembly in the opposite direction, beginning from the starting lines, we are able to proceed in both 11_1 and 9_2. Next to be assembled then was line $\overline{11_1, 12_1}$ and its matching line $\overline{9_2, 4_2}$. Since no next line in the decreasing cyclic order was found in 12_1, the next line in the decreasing order in 4_2, namely lines $\overline{4_2, 5_2}$, was picked. The assembly terminates here since 5_2, having no match, was not validated and not ordered. The recording of this face group assembly is done as follows:

$$\overline{12_1, 11_1} \quad \overline{5_2, 4_2}$$
$$\overline{11_1, 6_1} \quad \overline{4_2, 9_2}$$
$$\overline{9_2, 8_2} \qquad \overline{19_3, 20_3}$$
$$\overline{8_2, 7_2} \qquad \overline{20_3, 13_3}$$

Let us consider another example, in which a closed boundary is completely traced. The first line picked for this face group is $\overline{1_1, 4_1}$.

(The stars indicate the places where the linkage between the two junctions is the result of the data recovery phase):



The existence of a virtual junction usually indicates the existence of a multiple-boundaries face. Thus on assembling the lines that belong to a single boundary, all S and A junctions are remembered*; when the boundary trace is terminated, we return to these junctions, search for the other end of its limb, and start there another line assembly for the same face group. The limbs themselves are also added to this face group.

When all lines have been twice assembled, we may try some further data recovery actions. This second-level data recovery is carried out in the three following configurations shown in Fig. 5.2.

1. Two lines l and k in the same picture, with one non-valid end junction in each, are assembled into two face groups M and N, $k,l \in M$ and $k,l \in N$. Action: Combine l and k into one line.
2. A line l with one non-valid end junction is assembled into two face groups M and N, and there is in the same picture a 2-lines junction J with lines m and n, $m \in M$ and $n \in N$. Action: Extend the line l to junction J.

---

* In Chapter III it was stated that the two non-limb lines in every S junction are merged into one line. Thus, on assembling the lines, there is no change of lines in an S junction; the junction is bypassed and remembered.

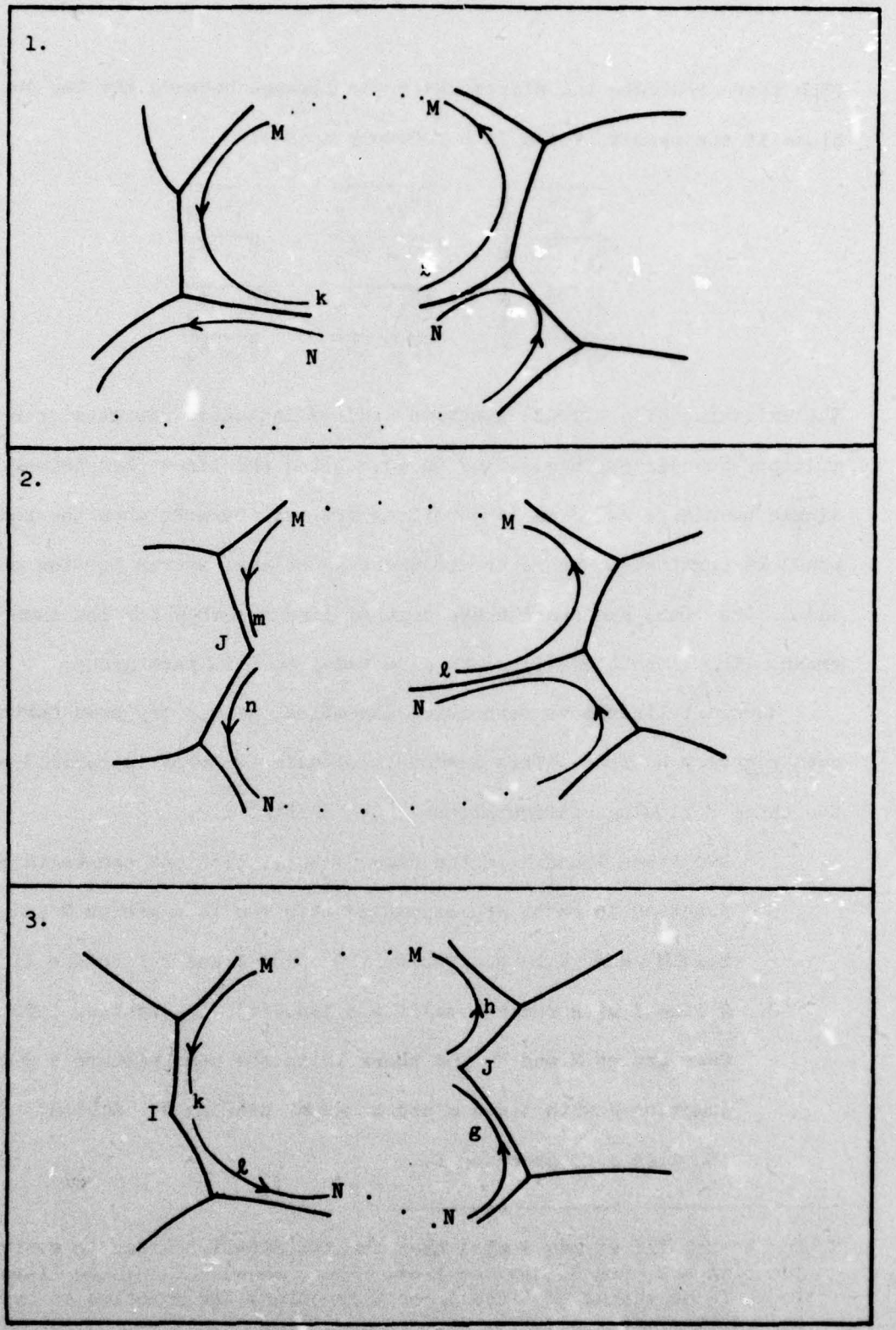


Fig. 5.2 Three Configurations for Second-Level Data Recovery

3. There are two 2-lines junctions I and J in the same picture with lines k,l and h,g, respectively, such that k and h are assembled into face group M, and l and g are assembled into face group N. Action: Create an "empty" line between the two junctions I and J.

The data recovery actions utilize the fact that two components assembled simultaneously into two face groups must correspond to the intersection of the two corresponding faces. There is of course the possibility that this intersection is sectioned into several separate edges and that the two components correspond to two different sections; but at this stage of the recognition phase, with no evidence to the contrary, we assume that they belong to the same section.

After any second-level data recovery actions have been performed, we try again to match lines, using the new conditions. Then conditions for second-level data recovery are checked again, and so on until no further line matching is possible. The face-group set is updated each time new results are obtained.

5.2 Object Formation

Each of the face groups defines (corresponds to) a face of one of the bodies in the scene. After the face-group assembly and second-level data recovery is completed, we are ready for the formation of objects. Each object is a set of face groups, with no common elements between two sets. Every face group must belong to some set. A new object is formed by picking a face group that is still free, then recursively adding to the set every face group, not as yet in the set, which has a line in common with any of the face groups already in the set.

5.3 Type of Face Determination

We would like to know for each face group whether it defines a planar or curved face. Toward that end let us define the predicate C(A) for every face group A as "the face defined by A is curved". Whenever a face group has a limb in it, we have, by definition, C(A)=1. When two face groups A and B share a straight line*, no information is conveyed about the faces' nature. However, when they share a curved line, we must have C(A) + C(B) = 1, that is, one of the faces or both must be curved. Thus, we have for every object a set of logical equations formalizing the constraints imposed by the types of lines.

The philosophy underlying our strategy for determining the nature of faces is the following: Given several pictures, every curved face will somehow manifest itself or there is no way to recognize it as curved. Thus, we are looking for a minimal cover to satisfy the set of logical equations for every object based on the information collected from all three (or more) pictures. This minimal cover will determine which of the face groups define a curved face.

We find the minimal cover by means of a minimum-cost tree search, where the cost function is the number of faces thus far determined as curved. Consider the one-picture example shown in Fig. 5.3. The set

---

* In Chapter IV we discussed how we determine whether a line is straight or curved. We say that two face groups share a curved line, if in at least one picture there is a curved line which they share.

of equations formalizing the lines constraints is:

1. C(B)=1

*2. C(B)+C(A)=1

*3. C(B)+C(F)=1

*4. C(B)+C(E)=1

5. C(C)+C(E)=1

6. C(A)+C(C)=1

*7. C(B)+C(D)=1

8. C(F)+C(C)=1

9. C(C)+C(D)=1

*10. C(G)+C(B)=1

The tree search, beginning with the first equation, is given in Fig. 5.4. The starred (*) equations are those that give no additional constraint since one of their elements participates in a single-member equation (e.g., B). The minimum-cost tree search gave the cover that determined the faces corresponding to B and C as curved. (Which agrees with our intuitive interpretation of the picture). After the nature of the faces is determined we may wish to classify those that are curved as to the type of their quadric equation. The flowchart of Fig. 5.5 shows a way of doing this.

5.4 Face Equations

We have already explained how the approximate 3D location of a point is determined from its two projections by taking the point of minimum sum of distances from the two projecting lines. If we are given three

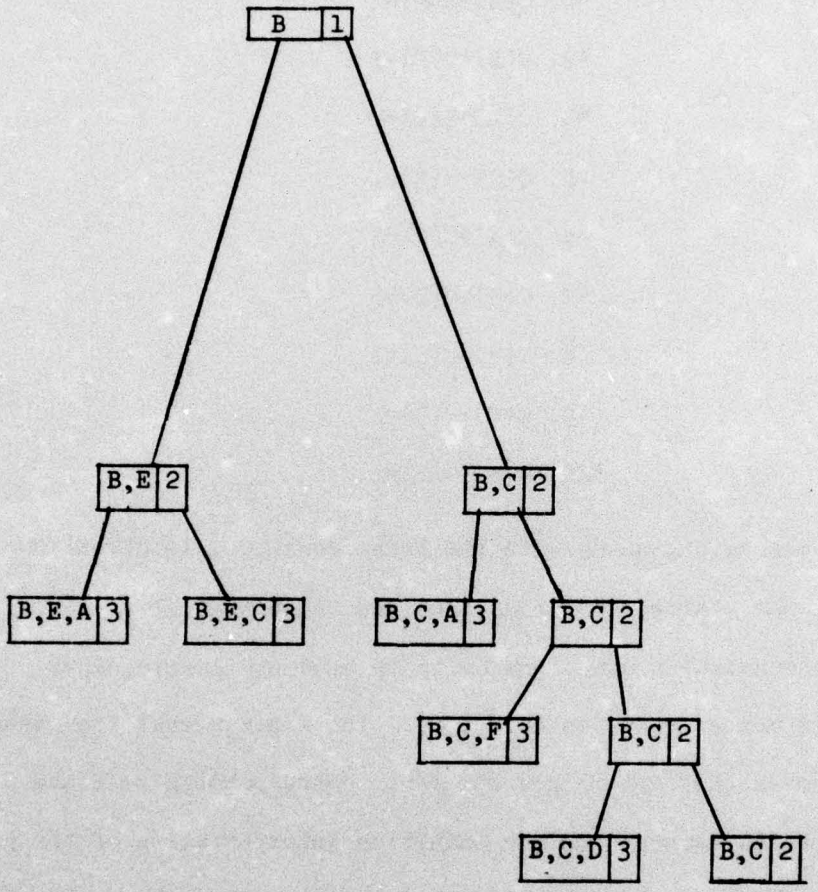


Fig. 5.3 Tree Search for Minimal Cover to Determine Curved-Face Group. (The number of the equation yielding each particular new level of nodes is indicated at the right.)

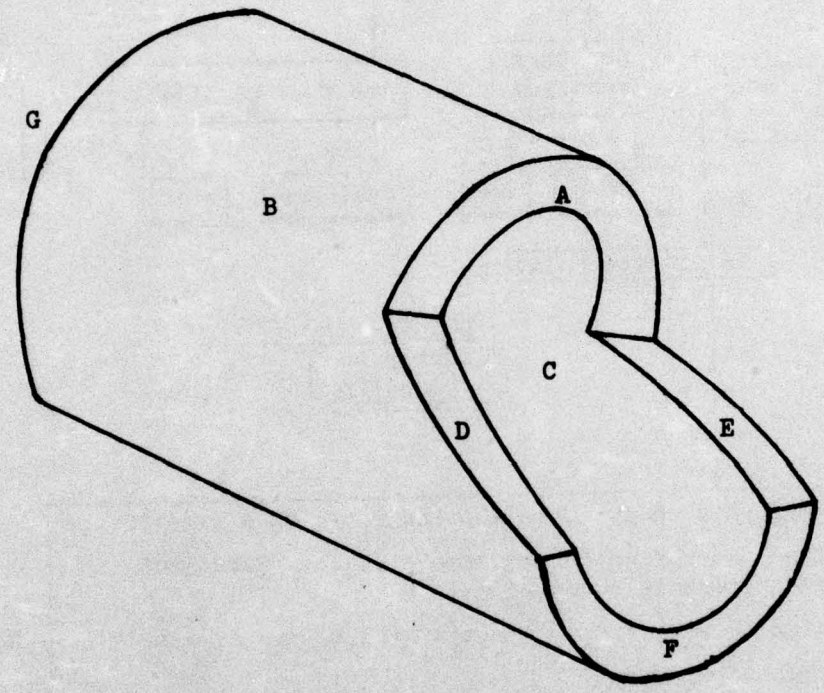
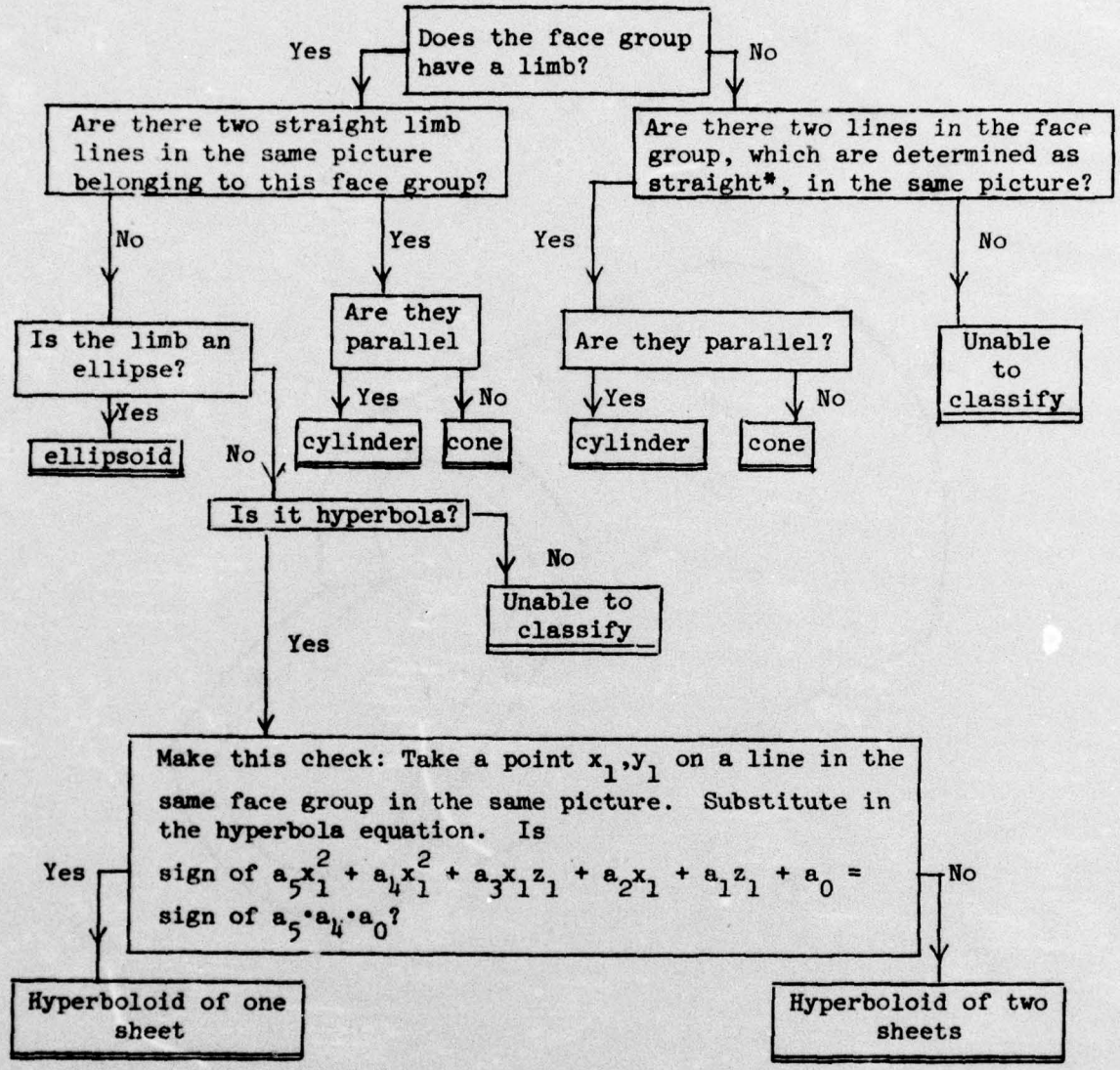


Fig. 5.4 Curved-Face Determination (Note that Face Group B will Consist of Two Line Assemblies and Two Limbs)



* Note: straight lines exist also on other types of quadric surfaces but we assume the most likely possibility. It is also assumed that parallelization is not badly distorted.

Fig. 5.5 Flow Chart for Determination of Type of Quadric Surface

projections of a point, then for a better approximation we may determine three 3D locations (by taking two projecting lines at a time) and then take their mean. This will give a very good approximation of the 3D location.

If we wish to have the equation of a face which we know to be planar, all its vertices which have at least two matched images can be collected in 3D and a plane equation fitted to them by minimizing the sum of square errors.

We shall now show how the equation of a curved face can be determined from its limb line in three different pictures.

The equation of a quadric surface is

$$Q(x,y,z) \equiv \underline{u}^T \cdot \underline{A} \cdot \underline{u} = 0$$

where A is the matrix of coefficients

$$\underline{A} = \begin{bmatrix} a_{11} & \frac{1}{2}a_{12} & \frac{1}{2}a_{13} & \frac{1}{2}a_{14} \\ \frac{1}{2}a_{12} & a_{22} & \frac{1}{2}a_{23} & \frac{1}{2}a_{24} \\ \frac{1}{2}a_{13} & \frac{1}{2}a_{23} & a_{33} & \frac{1}{2}a_{34} \\ \frac{1}{2}a_{14} & \frac{1}{2}a_{24} & \frac{1}{2}a_{34} & a_{44} \end{bmatrix}$$

and u is a point in homogeneous coordinates

$$\underline{u} = (x,y,z,1)^T$$

If we denote by u_i the center of projection for picture i, then the limb of Q with respect to u_i lies on the polar plane of u_i with respect to Q.

$$P(x,y,z) \equiv \underline{u_i}^T \cdot \underline{A} \cdot \underline{u} = 0$$

The cone of projection belongs to the linear family

$$\underline{u}^T \cdot \underline{A} \cdot \underline{u} - k(\underline{u}_i^T \cdot \underline{A} \cdot \underline{u})^2 = 0$$

Since $\underline{u}_i$ belongs to the cone of projection, we get

$$k = \frac{1}{\underline{u}_i^T \cdot \underline{A} \cdot \underline{u}_i}$$

and the cone of projection for the surface onto picture i is

$$c(x,y,z) \equiv \underline{u}_i^T \cdot \underline{A} \cdot \underline{u}_i \cdot \underline{u}^T \cdot \underline{A} \cdot \underline{u} - (\underline{u}_i^T \cdot \underline{A} \cdot \underline{u})^2 = 0$$

Expressing this cone in picture i's coordinate system and substituting 0 for y^i coordinate, we get the equation of the limb expressed in terms of the unknown quadric's coefficients. The coefficients of the calculated limb line, which are functions of the desired coefficients $a_{n,j}$, should be proportional to the coefficients of the fitted conic. This gives us, for ideal data, six non-linear equations in the ten unknowns plus the unknown proportionality coefficient:

$$F_{n,i}(a_{11}, a_{12}, \ldots\ldots a_{33}, a_{44}, c_i) = 0$$

$$n=1\ldots.6$$

A second picture will give us six more equations but only four of them will be independent of the first picture; it will add, however, an unknown proportionality coefficient.

$$F_{n,j}(a_{11}, a_{12}, \ldots\ldots a_{33}, a_{44}, c_j) = 0$$

$$n=1\ldots.6$$

The third picture will add six more equations and another unknown

$$F_{n,k}(a_{11}, a_{12}, \ldots\ldots a_{33}, a_{44}, c_k) = 0$$

$$n=1\ldots.6$$

but only two of the equations will be independent of the other two pictures. This, together with the normalization condition

$$\sum_{j=i}^{4} \sum_{i=1}^{4} a_{ij}^2 = 1$$

give us thirteen non-linear equations in thirteen unknowns. In the case of real data we can solve for the unknowns by looking for a minimum for the functions F together with the last equality equation. The algorithms concerning the faces' equations were not programmed.

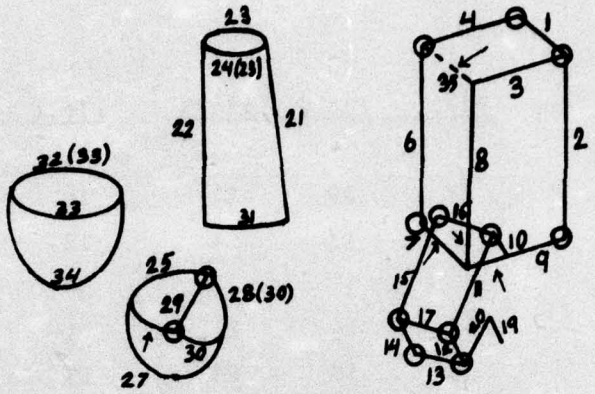
VI. EXPERIMENTAL RESULTS

6.1 First Example

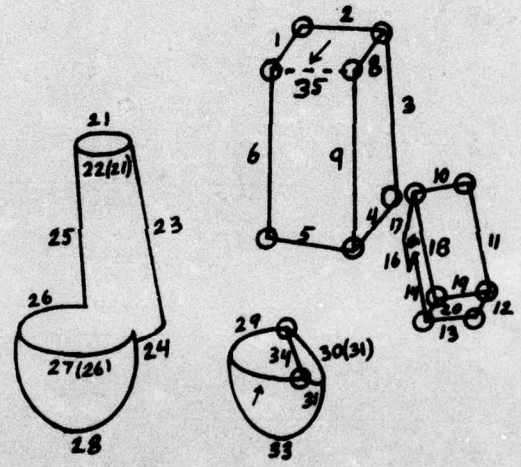
For the first example consider the scene which was discussed in Chapters 3 and 4 (see Fig. 4.7). The face groups reported by the program are as follows: (For line notation see Fig. 6.1, where also part of the data recovery results are shown.)

	Pic.1	Pic.2	Pic.3	Pic.1	Pic.2	Pic.3
Face Group 1						
Lines:	33	26	28			
Limb Lines:	34	28	26			
Face Group 2						
Lines:	33	26	28			
Face Group 3						
Lines:	31	24	10	23	21	13
Limb lines:	22,21	23,25	14,11			
Face Group 4						
Lines:	31	24	10			
Face Group 5						
Lines:	30	31	-			
	25	29	-			
Limb lines:	27	33				
Face Group 6						
Lines:	30	31	-	29	34	-
Face Group 7						
Lines:	29	34	-	25	29	-
Face Group 8						
Lines:	23	21	13			
Face Group 9						
Lines:	20	-	19	13	13	-
	-	14	-			

PICTURE 1



PICTURE 2



PICTURE 3

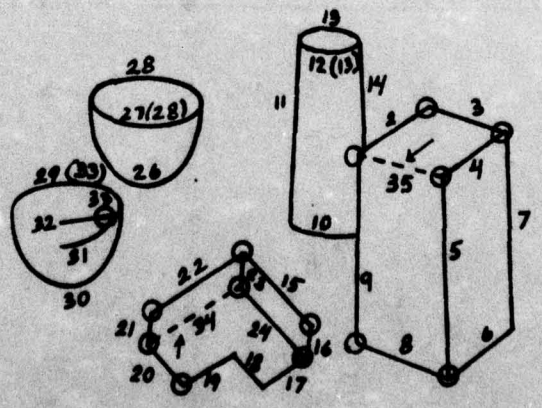


Fig. 6.1 First Example. (The Circled Junctions are Cyclically Arranged. The Arrows Point at Data Recovery Results. The Broken Lines Indicate "Empty Lines", for which only the End Points are Known.)

	Pic.1	Pic.2	Pic.3	Pic.1	Pic.2	Pic.3
Face Group 10						
Lines:	17	19	21	14	20	-
	13	13	-	12	12	20
Face Group 11						
Lines:	17	19	21	11	11	34
	16	10	23	15	18	22
Face Group 12						
Lines:	16	10	23	10	-	24
	-	-	16	-	17	15
Face Group 13						
Lines:	15	18	22	-	17	15
	14	20	-	-	14	-
Face Group 14						
Lines:	12	12	20	20	-	19
	11	11	34	10	-	24
	-	-	17			
Face Group 15						
Lines:	9	4	8	7	5	-
	-	-	6			
Face Group 16						
Lines:	8	9	9	35	35	2
	6	6	-	7	5	-
Face Group 17						
Lines:	8	9	9	9	4	8
	2	3	5	3	8	35
Face Group 18						
Lines:	6	6	-	4	1	3
	-	-	7			
Face Group 19						
Lines:	4	1	3	35	35	2
	3	8	35	1	2	4
Face Group 20						
Lines:	2	3	5	-	-	6
	1	2	4	-	-	7

The following face groups were labelled as curved: 1, 3, 5. The program determined the following bodies:

Body 1 with face groups 1 and 2.

Body 2 with face groups 3, 4 and 8.

Body 3 with face groups 5, 6 and 7.

Body 4 with face groups 9, 10, 11, 12, 13 and 14.

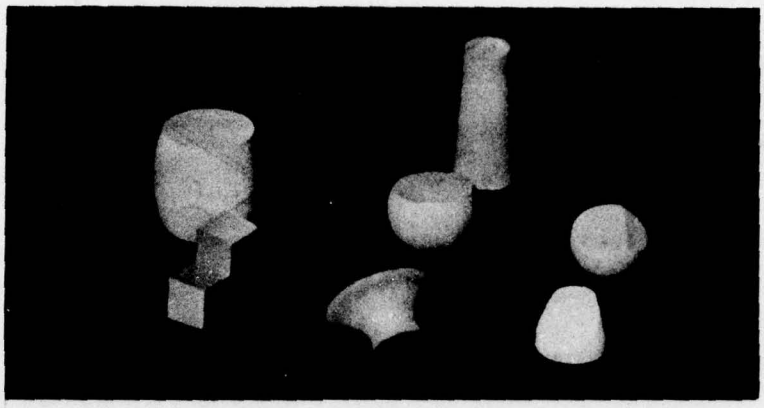
Body 5 with face groups 15, 16, 17, 18, 19 and 20.

6.2 Second Example

The scene analyzed for the second example is shown in Fig. 6.2. The input data are shown in Figures 6.3, 6.4, and 6.5. Part of the data recovery results is shown in Fig. 6.6. The computer output is given in Fig. 6.7 (a) through (f).

6.3 Third Example

The scene for the third example is shown in Fig. 6.8. Input data is shown in Fig's. 6.9, 6.10, and 6.11. Part of the data recovery results is shown in Fig. 6.12. The computer output is given in Fig. 6.13 (a) through (f). Of special interest is the determination of the curved faces.



(a)

(b)

(c)

Fig. 6.2 Second Scene Example. (a) Picture 1, (b) Picture 2, (c) Picture 3

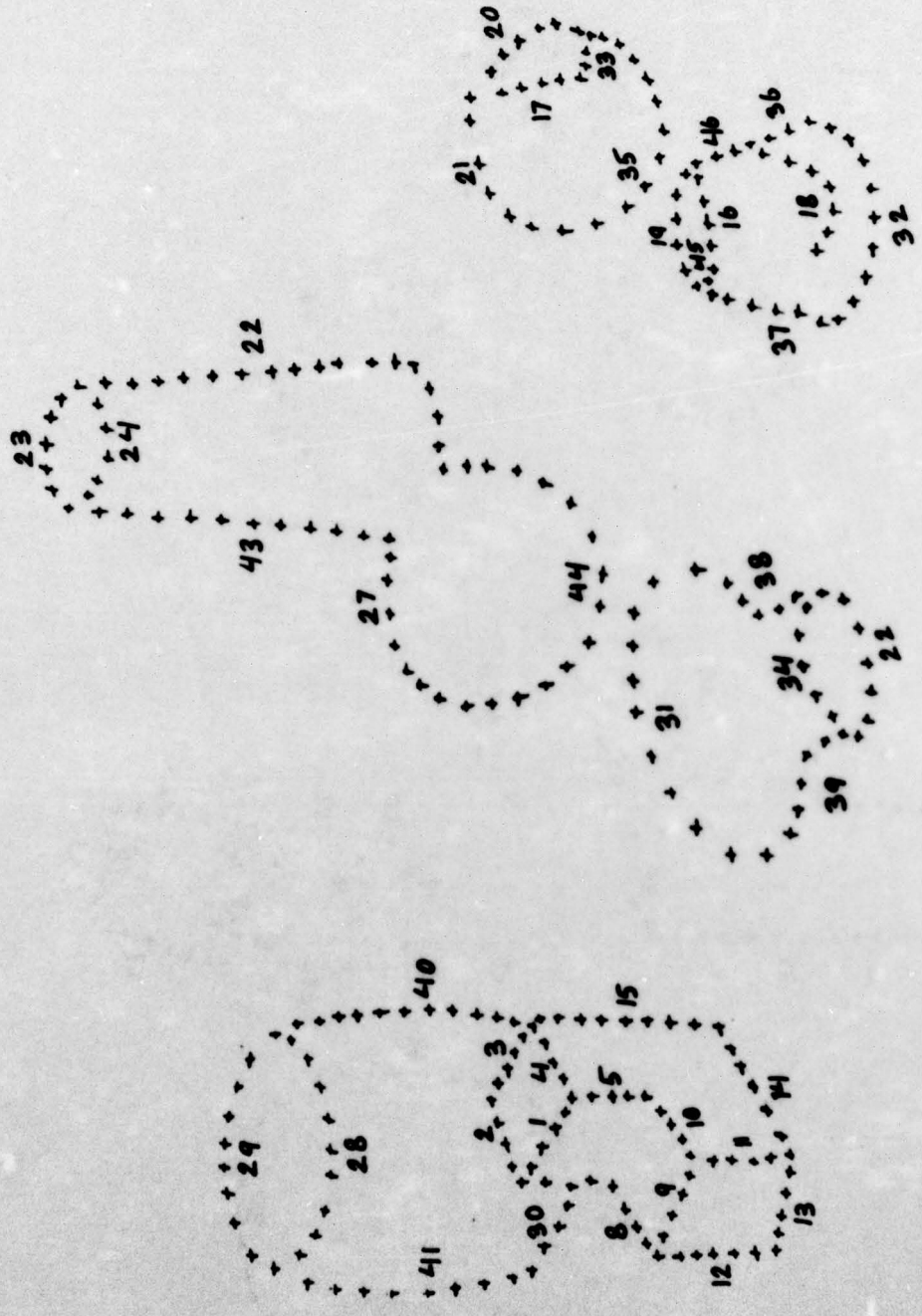


Fig. 6.3 Second Scene: Picture-1 Input Data

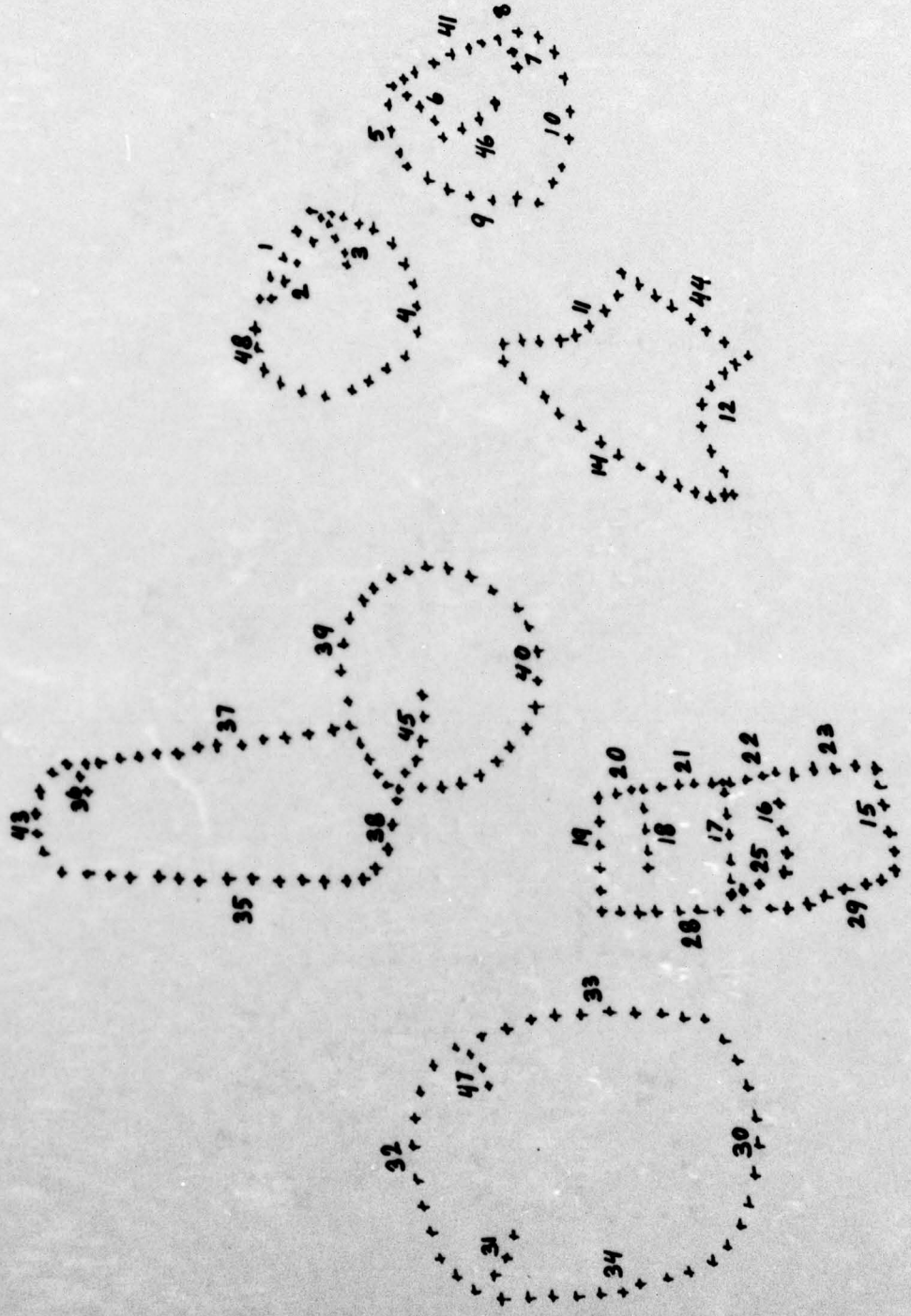


Fig. 6.4 Second Scene: Picture-2 Input Data

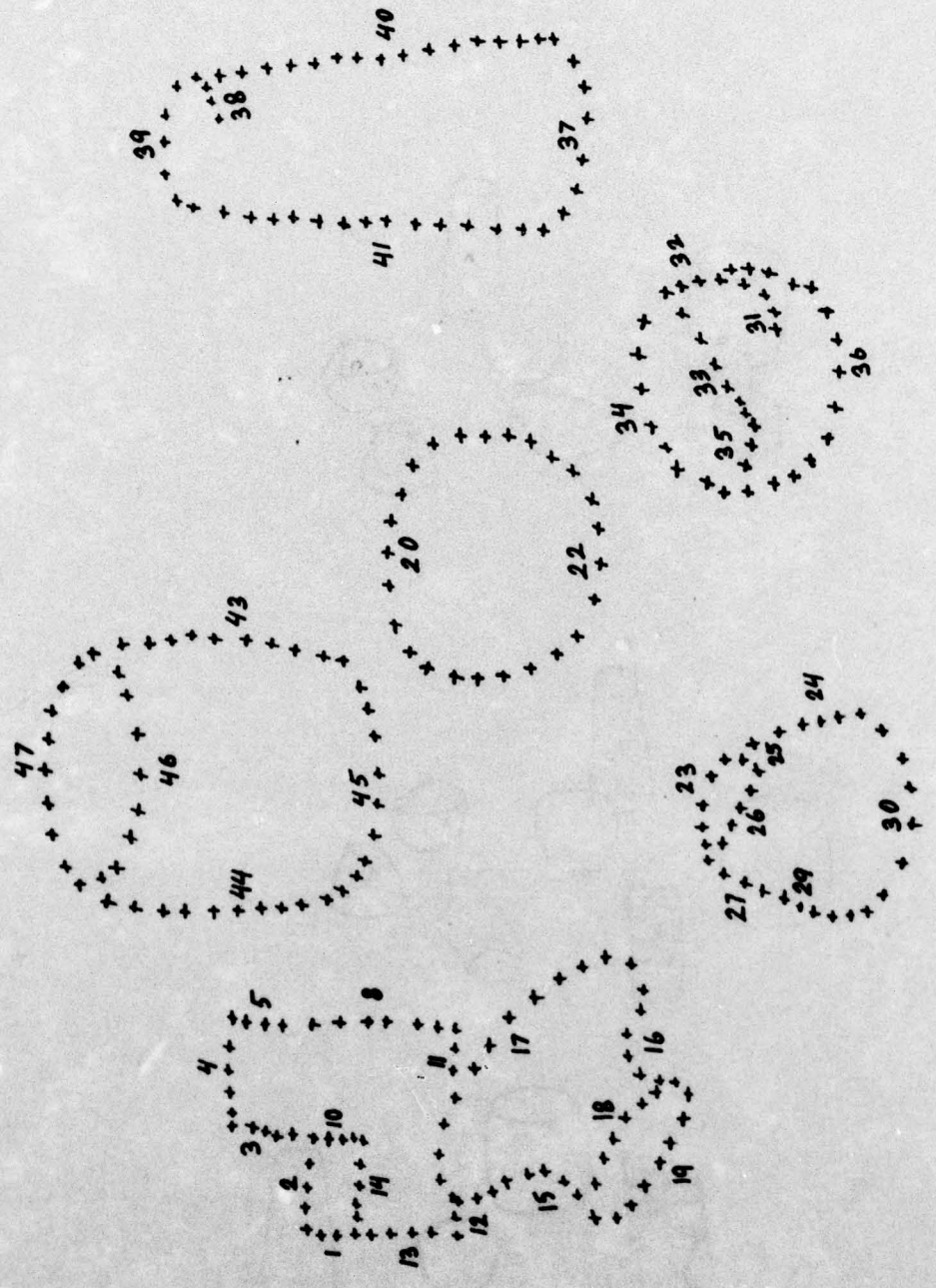


Fig. 6.5 Second Scene: Picture-3 Input Data

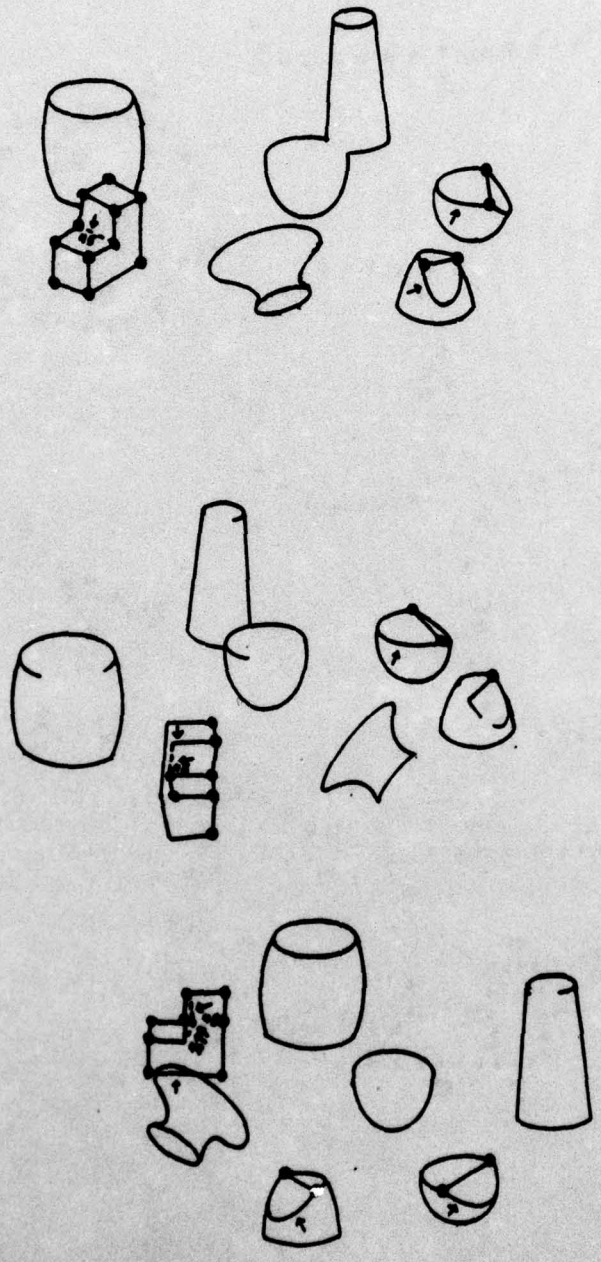


Fig. 6.6 Example 2 (For Explanation of Markings see Fig. 6.1)

```
A TRIPLE WAS FOUND      LINE      27      LINE      39      LINE
  20
A TRIPLE WAS FOUND      LINE      25      LINE      38      LINE
  37
A TRIPLE WAS FOUND      LINE      24      LINE      36      LINE
  38
A TRIPLE WAS FOUND      LINE      30      LINE      30      LINE
  45
A TRIPLE WAS FOUND      LINE      31      LINE      14      LINE
  17
A TRIPLE WAS FOUND      LINE      32      LINE      10      LINE
  30
-------------------------------------------------
                REGION ASSEMBLY
-------------------------------------------------
MEMBERS OF FACE GROUP NUMBER                 1
*************************************************
 P16 PARAMETERS          34          1          29
SINGLE MEMBER REGION     34     IN PICTURE      1
LINES                    34         64          18
LINES OF FACE GROUP       1
  ARE                    38     IN PICTURE      1          39      IN PICTURE
                          1         16      IN PICTURE     3          15
 IN PICTURE               3
 P16 PARAMETERS          31          1          31
SINGLE MEMBER REGION     31     IN PICTURE      1
LINES                    31         14          17
MEMBERS OF FACE GROUP NUMBER                 2
*************************************************
```

Fig. 6.7 (a) Computer Output Data for Example 2 (Continued)

```
 P16 PARAMETERS          34                    1                    29
SINGLE MEMBER REGION     34                IN PICTURE               1
LINES                    34                   44                    18
MEMBERS OF FACE GROUP NUMBER                   3
**************************************************
 P16 PARAMETERS          33                    1                    14
LINES                    33                   42                    31
LINES                    21                   48                    34
LINES OF FACE GROUP       3
  ARE                    35                IN PICTURE               1                     4                 IN PICTURE
                          2                    36                IN PICTURE               3
MEMBERS OF FACE GROUP NUMBER                   4
**************************************************
 P16 PARAMETERS          33                    1                    15
LINES                    33                   42                    31
LINES                    17                    2                    33
MEMBERS OF FACE GROUP NUMBER                   5
**************************************************
 P16 PARAMETERS          32                    1                    27
SINGLE MEMBER REGION     32                IN PICTURE               1
LINES                    32                   10                    30
LINES OF FACE GROUP       5
  ARE                    36                IN PICTURE               1                    37                 IN PICTURE
                          1                    9                 IN PICTURE               2                    8
 IN PICTURE              2                    29                IN PICTURE               3                   24
 IN PICTURE              3
 P16 PARAMETERS          18                    1                    12
LINES                    18                    7                    27
LINES                    19                    5                    25
MEMBERS OF FACE GROUP NUMBER                   6
**************************************************
 P16 PARAMETERS          32                    1                    28
SINGLE MEMBER REGION     32                IN PICTURE               1
LINES                    32                   10                    30
MEMBERS OF FACE GROUP NUMBER                   7
**************************************************
 P16 PARAMETERS          31                    1                    32
SINGLE MEMBER REGION     31                IN PICTURE               1
LINES                    31                   14                    17
MEMBERS OF FACE GROUP NUMBER                   8
**************************************************
 P16 PARAMETERS          30                    1                    18
SINGLE MEMBER REGION     30                IN PICTURE               1
LINES                    30                   30                    45
LINES OF FACE GROUP       8
  ARE                    41                IN PICTURE               1                    33                 IN PICTURE
                          2                    34                IN PICTURE               2                   43
 IN PICTURE              3                    44                IN PICTURE               3
 P16 PARAMETERS          28                    1                    16
SINGLE MEMBER REGION     28                IN PICTURE               1
LINES                    28                   47                    46
MEMBERS OF FACE GROUP NUMBER                   9
**************************************************
 P16 PARAMETERS          30                    1                    33
SINGLE MEMBER REGION     30                IN PICTURE               1
LINES                    30                   30                    45
MEMBERS OF FACE GROUP NUMBER                  10
**************************************************
 P16 PARAMETERS          28                    1                    16
SINGLE MEMBER REGION     28                IN PICTURE               1
```

Fig. 6.7 (b) Computer Output Data for Example 2 (Continued)

```
LINES                    28                47                46
MEMBERS OF FACE GROUP NUMBER               11
**************************************************
 P16 PARAMETERS          27                 1                21
SINGLE MEMBER REGION     27           IN PICTURE             1
LINES                    27                39                20
LINES OF FACE GROUP      11
  ARF                    44           IN PICTURE             1                40           IN PICTURE
                          2                22           IN PICTURE             3
MEMBERS OF FACE GROUP NUMBER               12
**************************************************
 P16 PARAMETERS          27                 1                35
SINGLE MEMBER REGION     27           IN PICTURE             1
LINES                    27                39                20
MEMBERS OF FACE GROUP NUMBER               13
**************************************************
 P16 PARAMETERS          25                 1                36
SINGLE MEMBER REGION     25           IN PICTURE             1
LINES                    25                38                37
LINES OF FACE GROUP      13
  ARF                    42           IN PICTURE             1                35           IN PICTURE
                          2                41           IN PICTURE             3                40
 IN PICTURE               3
 P16 PARAMETERS          24                 1                19
SINGLE MEMBER REGION     24           IN PICTURE             1
LINES                    24                36                38
MEMBERS OF FACE GROUP NUMBER               14
**************************************************
 P16 PARAMETERS          25                 1                37
SINGLE MEMBER REGION     25           IN PICTURE             1
LINES                    25                38                37
MEMBERS OF FACE GROUP NUMBER               15
**************************************************
 P16 PARAMETERS          24                 1                19
SINGLE MEMBER REGION     24           IN PICTURE             1
LINES                    24                36                38
MEMBERS OF FACE GROUP NUMBER               16
**************************************************
 P16 PARAMETERS          21                 1                14
LINES                    21                48                34
LINES                    17                 2                33
MEMBERS OF FACE GROUP NUMBER               17
**************************************************
 P16 PARAMETERS          19                 1                12
LINES                    19                 5                25
LINES                    16                 6                26
MEMBERS OF FACE GROUP NUMBER               18
**************************************************
 P16 PARAMETERS          18                 1                13
LINES                    18                 7                27
LINES                    16                 6                26
MEMBERS OF FACE GROUP NUMBER               19
**************************************************
 P16 PARAMETERS          15                 1                 2
LINES                    15              NONE                 8                NONE
LINES                     3                19                 5
MEMBERS OF FACE GROUP NUMBER               20
**************************************************
 P16 PARAMETERS          15                 1                 3
LINES                    15              NONE                 8                NONE
```

Fig. 6.7 (c) Computer Output Data for Example 2
(Continued)

AD-A035 563 RENSSELAER POLYTECHNIC INST TROY N Y COMPUTER RESEAR--ETC F/G 9/2
COMPUTER RECONSTRUCTION OF BODIES BOUNDED BY QUADRIC SURFACES F--ETC(U)
SEP 76 R SHAPIRA AF-AFOSR-2937-76
UNCLASSIFIED CRL-48 AFOSR-TR-77-0051 NL

2 OF 2
AD A035563

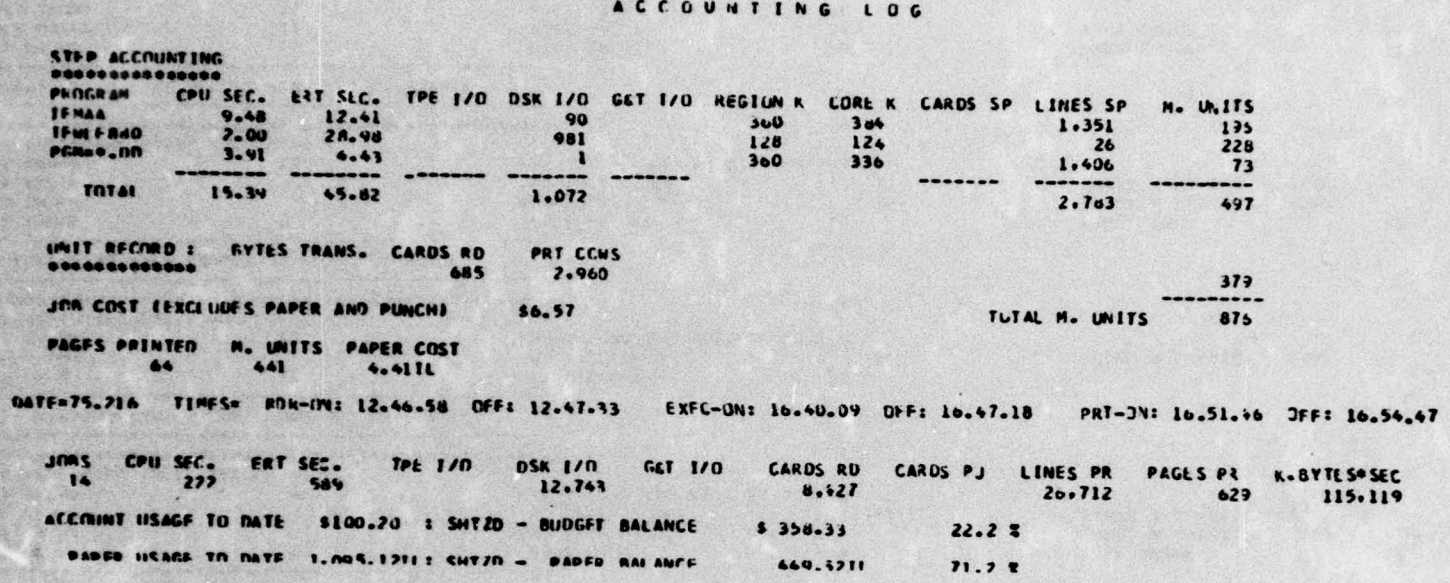
END
DATE FILMED
3-77

```
LINES                    4                  20                 47
LINES                    5                  21                 10
LINES                   10                  22                 14
LINES                   11                  23                 13
LINES                   14                NONE                 12                NONE
MEMBERS OF FACE GROUP NUMBER                 21
--------------------------------------------------
 P16 PARAMETERS         14                   1                  1
LINES                   14                NONE                 12                NONE
LINES                   13                  15                NONE               NONE
MEMBERS OF FACE GROUP NUMBER                 22
--------------------------------------------------
 P16 PARAMETERS         13                   1                  1
LINES                   13                  15                NONE               NONE
LINES                   11                  23                 13
LINES                    9                  16                  1
LINES                   12                NONE                NONE               NONE              NONE
MEMBERS OF FACE GROUP NUMBER                 23
--------------------------------------------------
 P16 PARAMETERS         10                   1                 10
LINES                   10                  22                 14
LINES                   45                  17                 49
LINES                    8                  25                  2
LINES                    9                  16                  1
MEMBERS OF FACE GROUP NUMBER                 24
--------------------------------------------------
 P16 PARAMETERS          8                   1                  6
LINES                    8                  25                  2
LINES                    7                  48                 48
LINES                    2                NONE                  4                NONE
LINES                   12                NONE                NONE               NONE              NONE
MEMBERS OF FACE GROUP NUMBER                 25
--------------------------------------------------
 P16 PARAMETERS          7                   1                  5
LINES                    7                  48                 48
LINES                   45                  17                 49
LINES                    5                  21                 10
LINES                    1                  18                  3
MEMBERS OF FACE GROUP NUMBER                 26
--------------------------------------------------
 P16 PARAMETERS          4                   1                  3
LINES                    4                  20                 47
LINES                    3                  19                  5
LINES                    2                NONE                  4                NONE
LINES                    1                  18                  3
--------------------------------------------------
 OBJECTS FORMATIONS
--------------------------------------------------
--------------------------------------------------
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER     1
--------------------------------------------------
FACE GROUP               1          FACE GROUP               2          FACE GROUP               7
FACE GROUP                   1      AND FACE GROUP                   2  SHARE A CURVED    EDGE
FACE GROUP               1          AND FACE GROUP                   7  SHARE A CURVED    EDGE
FACE GROUP               1          HAS A LIMB AND MUST BE CURVED
 THE FOLLOWING FACES ARE CURVED:
--------------------------------------------------                       1
--------------------------------------------------
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER     2
--------------------------------------------------
```

Fig. 6.7 (d) Computer Output Data for Example 2 (Continued)

```
FACE GROUP          3          FACE GROUP          4          FACE GROUP          16
FACE GROUP              3      AND FACE GROUP          4      SHARE A CURVED EDGE
FACE GROUP              3      AND FACE GROUP          16     SHARE A CURVED EDGE
FACE GROUP          3          HAS A LIMB AND MUST BE CURVED
 THE FOLLOWING FACES ARE CURVED:
**********************************************************          3
----------------------------------------------------------
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER        3
----------------------------------------------------------
FACE GROUP          5          FACE GROUP          6          FACE GROUP          17
 FACE GROUP        18
FACE GROUP              5      AND FACE GROUP          6      SHARE A CURVED EDGE
FACE GROUP              5      AND FACE GROUP          17     SHARE A CURVED EDGE
FACE GROUP              5      AND FACE GROUP          18     SHARE A CURVED EDGE
FACE GROUP          5          HAS A LIMB AND MUST BE CURVED
 THE FOLLOWING FACES ARE CURVED:
**********************************************************          5
----------------------------------------------------------
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER        4
----------------------------------------------------------
FACE GROUP          8          FACE GROUP          9          FACE GROUP          10
FACE GROUP              8      AND FACE GROUP          9      SHARE A CURVED EDGE
FACE GROUP              8      AND FACE GROUP          10     SHARE A CURVED EDGE
FACE GROUP          8          HAS A LIMB AND MUST BE CURVED
 THE FOLLOWING FACES ARE CURVED:
**********************************************************          8
----------------------------------------------------------
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER        5
----------------------------------------------------------
FACE GROUP         11          FACE GROUP         12
FACE GROUP             11      AND FACE GROUP         12     SHARE A CURVED EDGE
FACE GROUP         11          HAS A LIMB AND MUST BE CURVED
 THE FOLLOWING FACES ARE CURVED:
**********************************************************         11
----------------------------------------------------------
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER        6
----------------------------------------------------------
FACE GROUP         13          FACE GROUP         14          FACE GROUP          15
FACE GROUP             13      AND FACE GROUP         14     SHARE A CURVED EDGE
FACE GROUP             13      AND FACE GROUP         15     SHARE A CURVED EDGE
FACE GROUP         13          HAS A LIMB AND MUST BE CURVED
 THE FOLLOWING FACES ARE CURVED:
**********************************************************         13
----------------------------------------------------------
THE FOLLOWING FACE GROUPS FORM OBJECT NUMBER        7
----------------------------------------------------------
FACE GROUP         19          FACE GROUP         20          FACE GROUP          21
 FACE GROUP        22          FACE GROUP         23          FACE GROUP          24
 FACE GROUP        25          FACE GROUP         26
FACE GROUP             19      AND FACE GROUP         20     SHARE A CURVED EDGE
FACE GROUP             25      AND FACE GROUP         26     SHARE A CURVED EDGE
 THE FOLLOWING FACES ARE CURVED:
**********************************************************         19                  25
```

Fig. 6.7 (e) Computer Output Data for Example 2 (Continued)



ACCOUNTING LOG

STEP ACCOUNTING

PROGRAM	CPU SEC.	ERT SEC.	TPE I/O	DSK I/O	GET I/O	REGION K	CORE K	CARDS SP	LINES SP	M. UNITS
IFHAA	9.48	12.41		90		360	384		1,351	195
IFHIFB40	2.00	28.98		981		128	124		26	228
PGM=0.00	3.91	4.43		1		360	336		1,406	73
TOTAL	15.39	45.82		1,072					2,783	497

UNIT RECORD :	BYTES TRANS.	CARDS RD	PRT CCWS	
		685	2,960	
				379
			TOTAL M. UNITS	876

JOB COST (EXCLUDES PAPER AND PUNCH) $6.57

PAGES PRINTED	M. UNITS	PAPER COST
64	441	4.41IL

DATE=75.216 TIMES= RDR-IN: 12.46.58 OFF: 12.47.33 EXEC-ON: 16.40.09 OFF: 16.47.18 PRT-ON: 16.51.46 OFF: 16.54.47

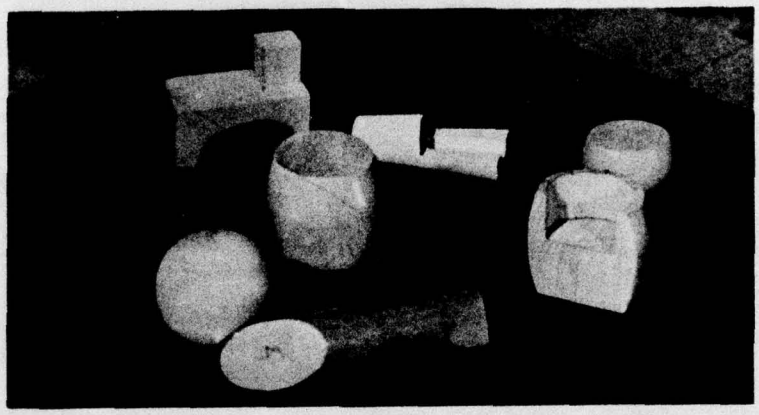
JOBS	CPU SEC.	ERT SEC.	TPE I/O	DSK I/O	GET I/O	CARDS RD	CARDS PJ	LINES PR	PAGES PR	K.BYTES*SEC
14	272	589		12,743		8,627		20,712	629	115,119

ACCOUNT USAGE TO DATE $100.20 : SHIZD - BUDGET BALANCE $ 358.33 22.2 %

PAPER USAGE TO DATE 1,005.1211 : SHIZD - PAPER BALANCE 669.6211 71.7 %

Fig. 6.7 (f) Computer Output Data for Example 2
(End)

(a)



(b)

(c)

Fig. 6.8 Third Scene Example. (a) Picture 1, (b) Picture 2, (c) Picture 3

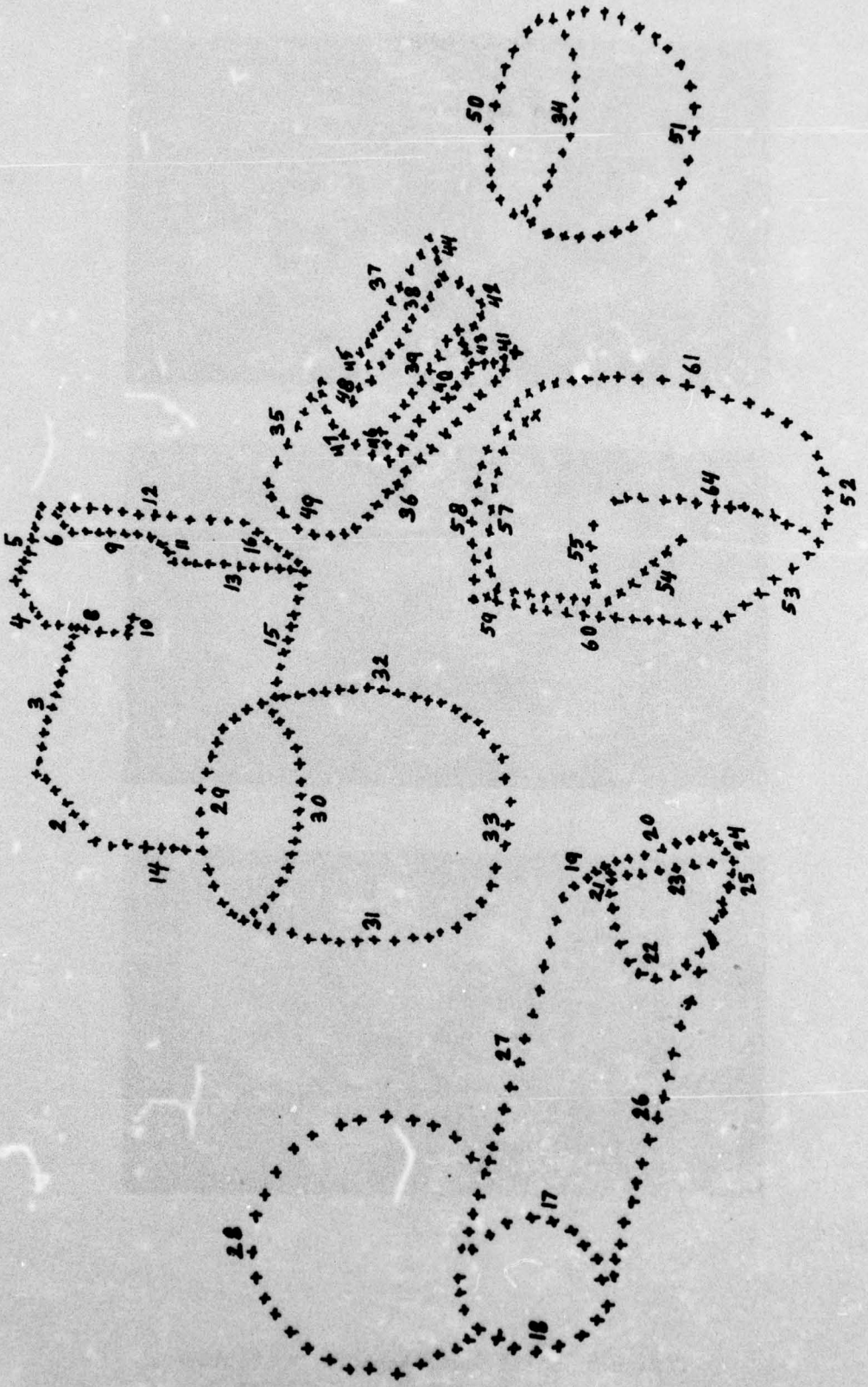


Fig. 6.9 Third Scene: Picture-1 Input Data

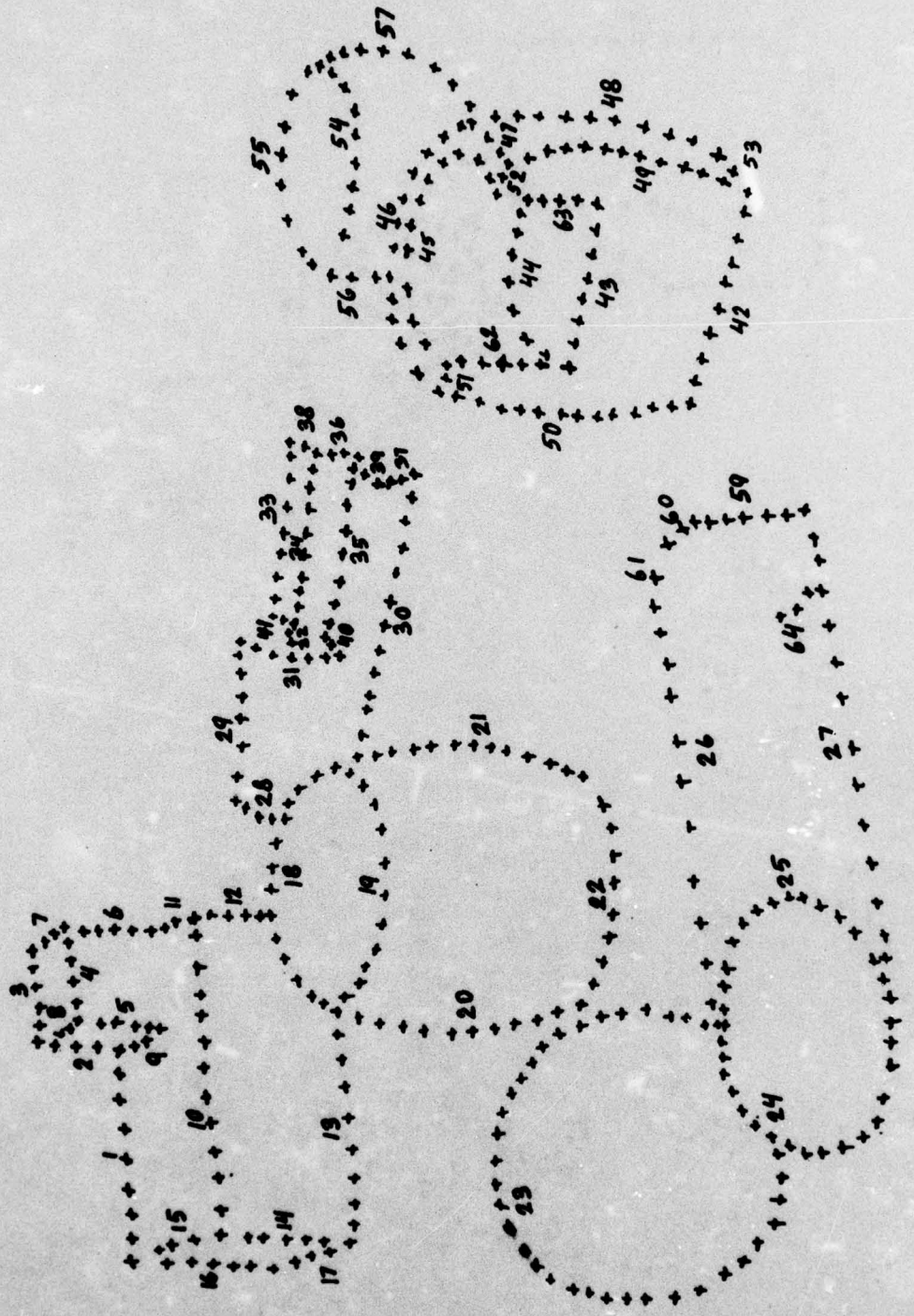


Fig. 6.10 Third Scene: Picture-2 Input Data

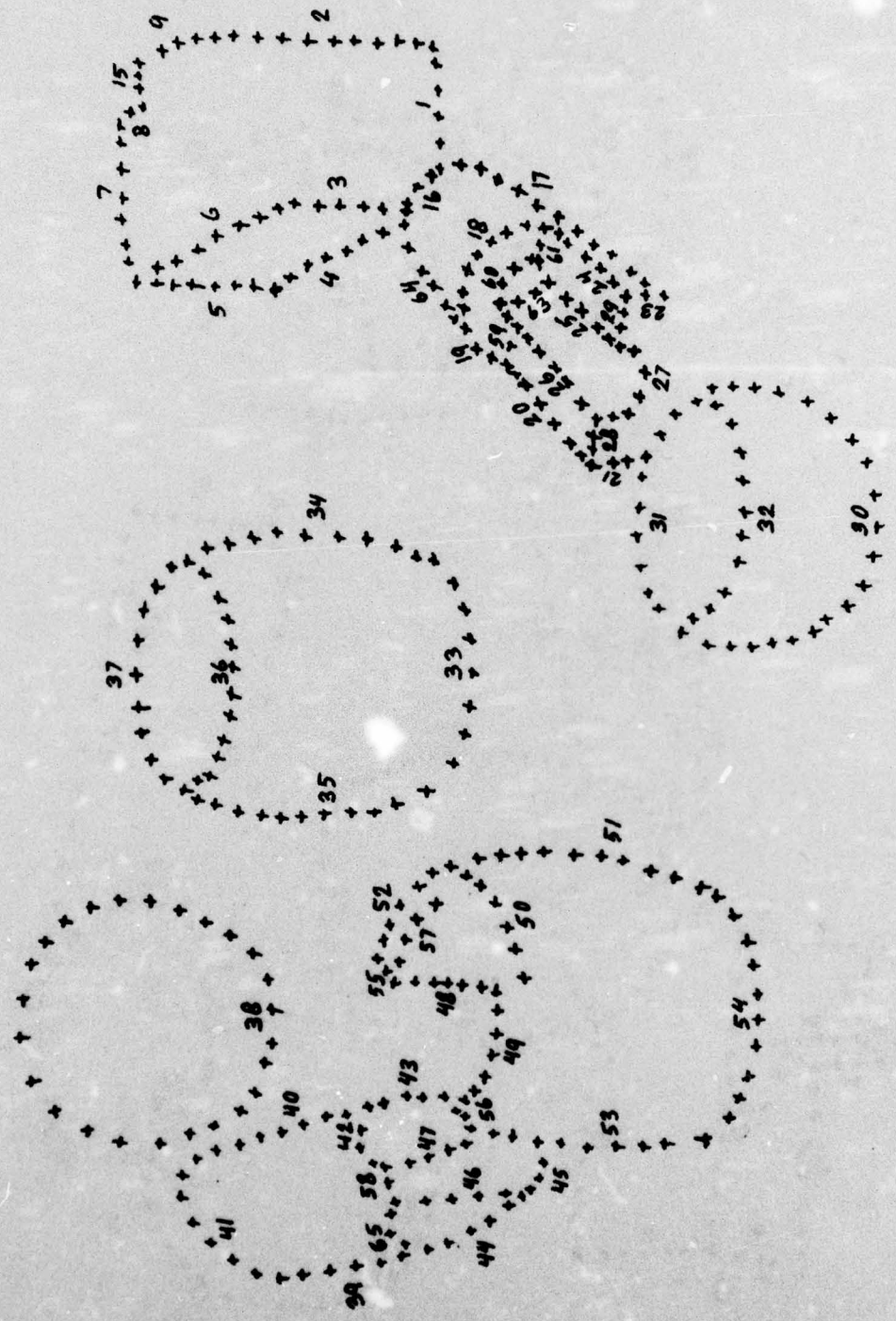
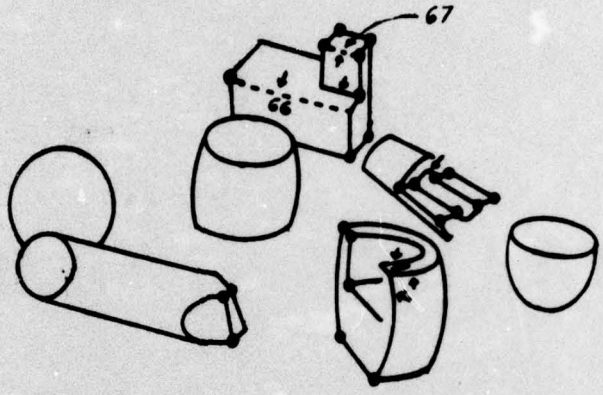
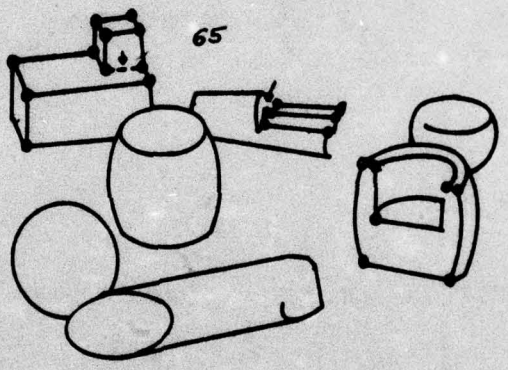


Fig. 6.11 Third Scene: Picture-3 Input Data

PICTURE 1



PICTURE 2



PICTURE 3

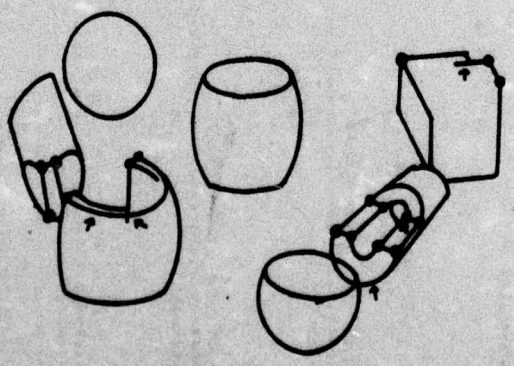


Fig. 6.12 Example 3 (For Explanation of Markings see Figure 6.1)

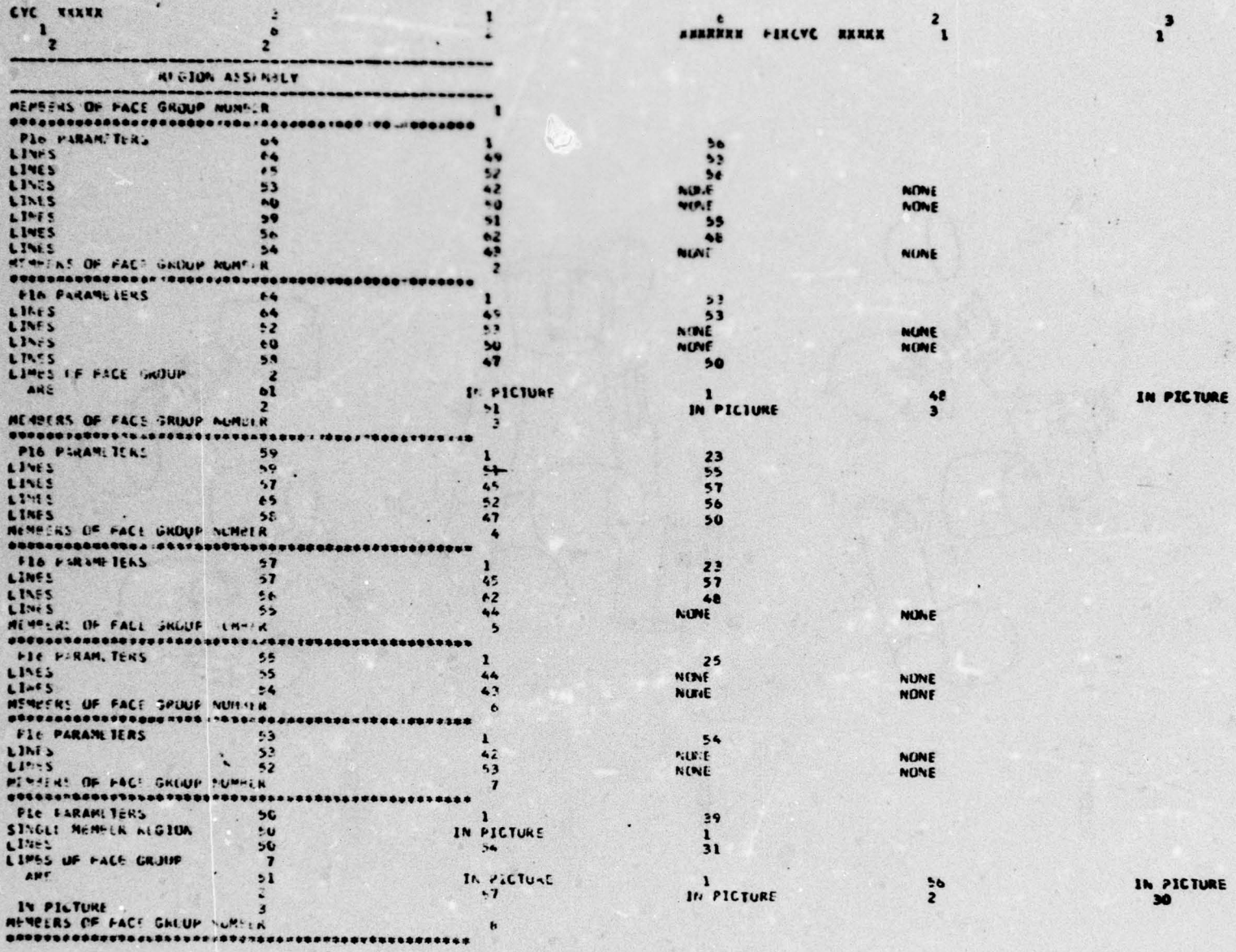
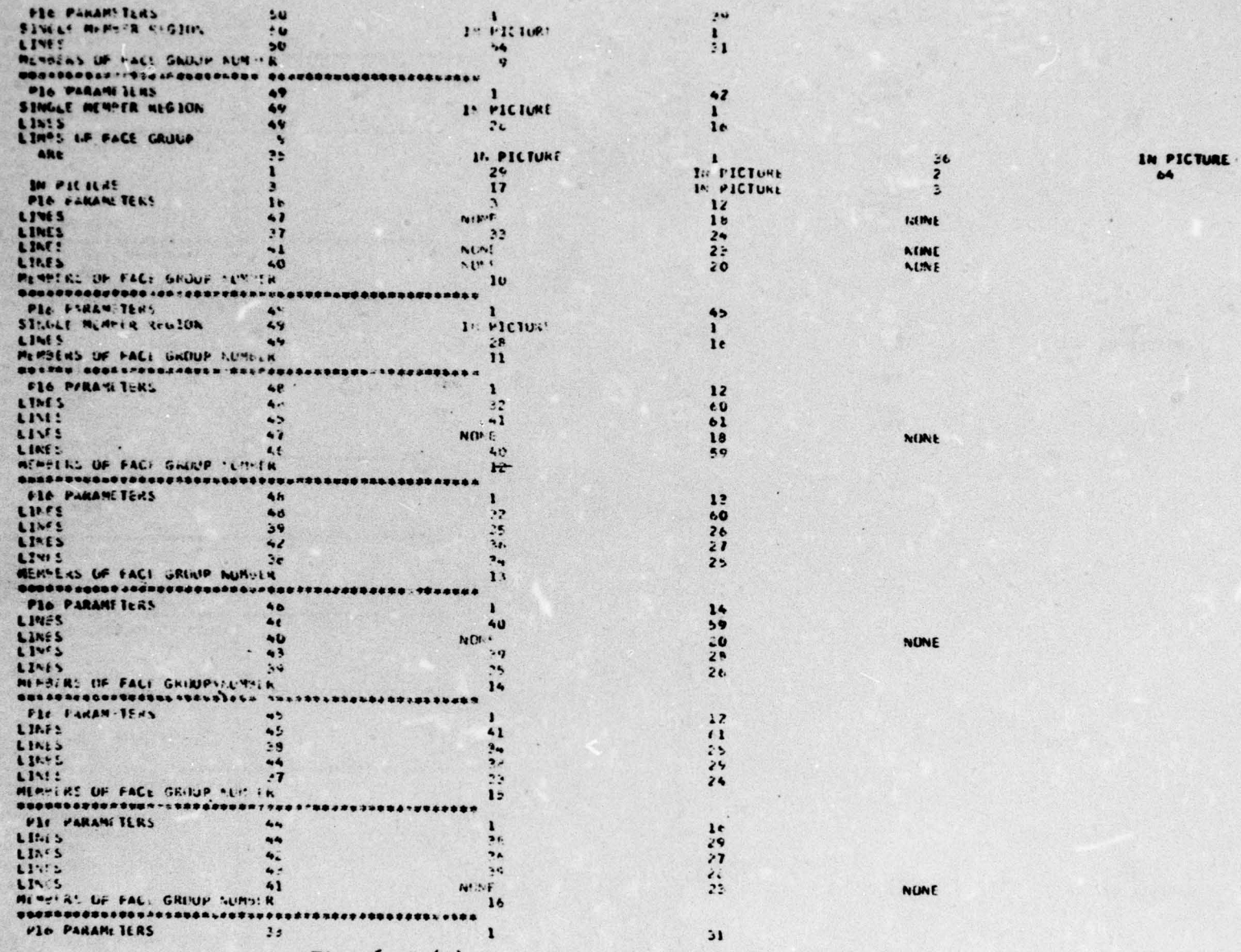


Fig. 6.13 (a) Computer Output Data for Example 3 (Continued)



```
 PIC PARAMETERS          50                    1                    29
SINGLE MEMBER REGION     50                    IN PICTURE           1
LINES                    50                    34                   31
MEMBERS OF FACE GROUP NUMBER                   9
 PIC PARAMETERS          49                    1                    42
SINGLE MEMBER REGION     49                    IN PICTURE           1
LINES                    49                    26                   16
LINES OF FACE GROUP
  ARE                    35                    IN PICTURE           1                    36                   IN PICTURE
                         1                     29                   IN PICTURE           2                    64
 IN PICTURE              3                     17                   IN PICTURE           3
 PIC PARAMETERS          16                    3                    12
LINES                    47                    NONE                 16                   NONE
LINES                    37                    32                   24
LINES                    41                    NONE                 23                   NONE
LINES                    40                    NONE                 20                   NONE
MEMBERS OF FACE GROUP NUMBER                   10
 PIC PARAMETERS          47                    1                    45
SINGLE MEMBER REGION     49                    IN PICTURE           1
LINES                    49                    28                   16
MEMBERS OF FACE GROUP NUMBER                   11
 PIC PARAMETERS          48                    1                    12
LINES                    44                    32                   60
LINES                    45                    41                   61
LINES                    47                    NONE                 18                   NONE
LINES                    45                    40                   59
MEMBERS OF FACE GROUP NUMBER                   12
 PIC PARAMETERS          46                    1                    13
LINES                    48                    32                   60
LINES                    39                    25                   26
LINES                    42                    36                   27
LINES                    36                    34                   25
MEMBERS OF FACE GROUP NUMBER                   13
 PIC PARAMETERS          46                    1                    14
LINES                    41                    40                   59
LINES                    40                    NONE                 20                   NONE
LINES                    43                    39                   28
LINES                    34                    25                   26
MEMBERS OF FACE GROUP NUMBER                   14
 PIC PARAMETERS          45                    1                    12
LINES                    45                    41                   41
LINES                    38                    34                   25
LINES                    44                    32                   29
LINES                    27                    23                   24
MEMBERS OF FACE GROUP NUMBER                   15
 PIC PARAMETERS          44                    1                    16
LINES                    44                    36                   29
LINES                    42                    34                   27
LINES                    42                    36                   26
LINES                    41                    NONE                 23                   NONE
MEMBERS OF FACE GROUP NUMBER                   16
 PIC PARAMETERS          39                    1                    31
```

Fig. 6.13 (b) Computer Output Data for Example 3 (Continued)

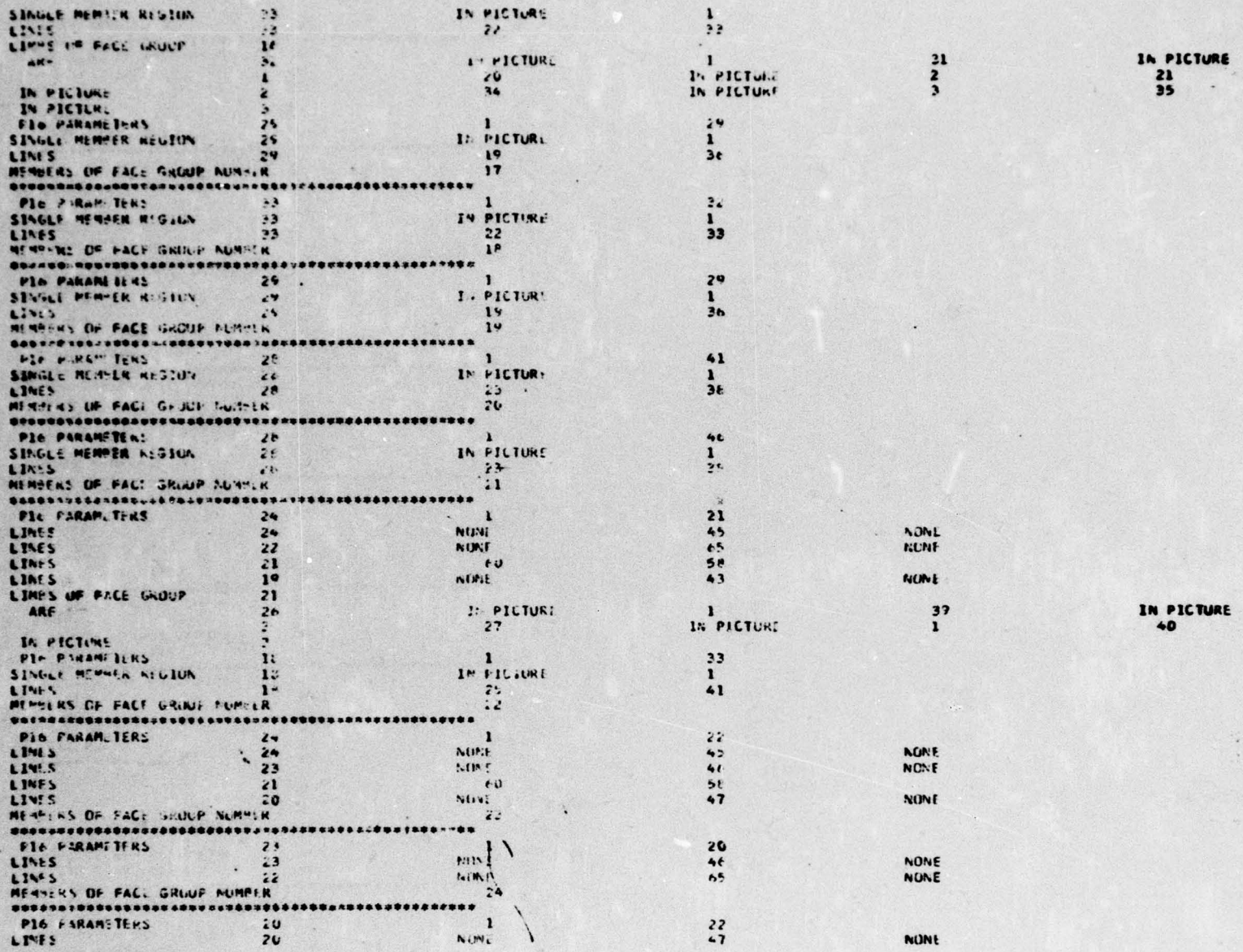


Fig. 6.13 (c) Computer Output Data for Example 3 (Continued)

LINES	14	NONE	43	NONE	
MEMBER OF FACT GROUP NUMBER			27		
FIE PARAMETERS	14		1	33	
SINGLE MEMBER SYSTEM	14	IN PICTURE	1		
LINES	16		25	41	
MEMBERS OF FACT GROUP NUMBER			28		
FIE PARAMETERS	14		1	7	
LINES	14		14	5	
LINES	2		15	7	
LINES	NONE		16	NONE	NONE NONE
LINES	NONE		17	NONE	NONE NONE
MEMBERS OF FACT GROUP NUMBER			27		
FIE PARAMETERS	14		1	58	
LINES	14		14	5	
LINES	NONE		13	NONE	NONE NONE
LINES	66		10	6	
LINES	13		12	NONE	NONE
LINES	15		NONE	NONE	NONE NONE
MEMBER OF FACT GROUP NUMBER			28		
FIE PARAMETERS	13		1	6	
LINES	13		12	NONE	NONE
LINES	16		NONE	NONE	NONE NONE
LINES	12		NONE	2	NONE
LINES	6		7	NONE	NONE
LINES	9		6	NONE	NONE
LINES	11		11	NONE	NONE
MEMBERS OF FACT GROUP NUMBER			29		
FIE PARAMETERS	12		1	5	
LINES	12		NONE	2	NONE
LINES	5		3	9	
LINES	NONE		2	NONE	NONE NONE
LINES	NONE		1	NONE	NONE NONE
LINES	NONE		18	NONE	NONE NONE
MEMBERS OF FACT GROUP NUMBER			30		
FIE PARAMETERS	11		1	2	
LINES	11		11	NONE	NONE
LINES	10		65	NONE	NONE
LINES	NONE		9	NONE	NONE NONE
LINES	NONE		1	NONE	NONE NONE
LINES	2		15	7	
LINES	10		10	6	
MEMBERS OF FACT GROUP NUMBER			31		
FIE PARAMETERS	10		1	2	
LINES	10		65	NONE	NONE
LINES	9		6	NONE	NONE
LINES	7		4	NONE	NONE
LINES	8		5	NONE	NONE
MEMBERS OF FACT GROUP NUMBER			32		
FIE PARAMETERS	8		1	9	
LINES	8		5	NONE	NONE
LINES	4		3	15	
LINES	NONE		2	NONE	NONE NONE

Fig. 6.13 (d) Computer Output Data for Example 3 (Continued)

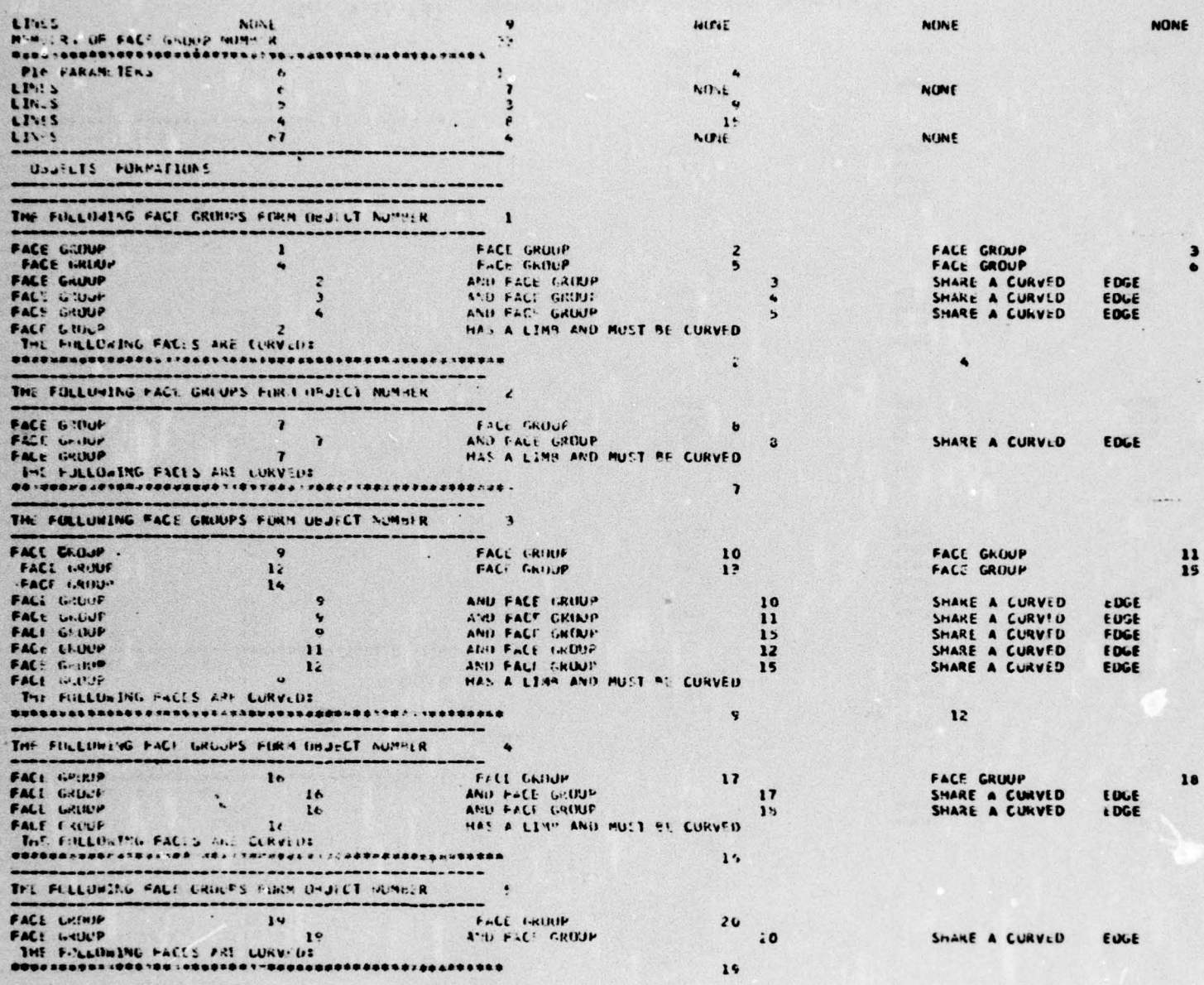


Fig. 6.13 (e) Computer Output Data for Example 3 (Continued)

```
-------------------------------------------------
THE FOLLOWING FACE GROUPS FORM OPEN LOOPS NUMBER    6
-------------------------------------------------
FACE GROUP        21          FACE GROUP        22          FACE GROUP        23
 FACE GROUP       24          FACE GROUP        25
FACE GROUP             21     AND FACE GROUP         23     SHARE A CURVED    EDGE
FACE GROUP             21     AND FACE GROUP         24     SHARE A CURVED    EDGE
FACE GROUP             21     AND FACE GROUP         25     SHARE A CURVED    EDGE
FACE GROUP        21          HAS A LINE AND MUST BE CURVED
 THE FOLLOWING FACES ARE CURVED:
oooooooooooooooooooooooooooooooooooooooooooooooo  21
-------------------------------------------------
THE FOLLOWING FACE GROUPS FORM OPEN LOOPS NUMBER    7
-------------------------------------------------
FACE GROUP        28          FACE GROUP        27          FACE GROUP        28
 FACE GROUP       29          FACE GROUP        30          FACE GROUP        31
 FACE GROUP       32          FACE GROUP        33
 THE FOLLOWING FACES ARE CURVED:
oooooooooooooooooooooooooooooooooooooooooooooooo
```

ACCOUNTING LOG

STEP ACCOUNTING

PROGRAM	CPU SEC.	ELAPSED SEC.	TAPE I/O	DISK I/O	GRAPHICS I/O	REGION K	CORE K	CARDS PU	LINES PR	NO. UNITS
IEYAA	9.25	42.15		80		360	364		1,295	192
IEWLF200	1.59	2.52		410		174	174		26	228
PGM=*.DD	6.97	7.44		1		360	268		2,187	125
TOTAL	18.19	48.37		1,070					3,508	545

UNIT RECORD :	BYTES TRANS.	CARDS RD	PRT LINES
oooooooooooo		720	3,099

	457
TOTAL NO. UNITS	1002

JOB COST (EXCLUDES PAPER AND PUNCH) 316.62

PAGES PRINTED	NO. UNITS	PAPER COST
60	553	1.38

DATE=75.216 RDR-ON: 12.44.02 OFF: 12.44.25 EXEC-ON: 16.17.37 OFF: 16.22.54 PRT-ON: 16.32.52 OFF: 16.35.26

JOBS	CPU SEC.	ELAPSED SEC.	TAPE I/O	DISK I/O	GRAPHICS I/O	CARDS RD	CARDS PU	LINES PR	PAGES PR	K.BYTES*SEC
13	207	543		11,071		7,742		23,699	565	105,349

ACCOUNT USAGE TO DATE $93.62 ; SMIZO - BUDGET BALANCE $ 364.90 20.4 %

Fig. 6.13 (f) Computer Output Data for Example 3
(End)

VII. SUMMARY AND CONCLUSIONS

7.1 Summary

The objective of this research was to investigate techniques for the computer understanding of pictures of scenes. Three important principles guided the research: (a) No complete preknowledge of the scenes' bodies should be presumed since that requires a limited repertoire of bodies. Instead, the number of bodies should be unrestricted and merely some general properties of the bodies should be prespecified. (b) These properties should be flexible enough to permit inclusion of all bodies likely to be encountered. (c) Since the input is to be gotten from photographs, the presence of data imperfections such as missing information, wrong information, and geometric inaccuracies must be taken into consideration. The difficulty of the described objective is highlighted by the fact that previous research in this area always presupposed the placing of severe restrictions on the problem environment. Where imperfect data was dealt with, only a small, fixed repertoire of polyhedra was allowed (6). Where the fixed repertoire was replaced by general properties of a family of bodies (though less general than those assumed in this thesis), perfect input data was required (3, 18).

To achieve the objective set here, set of multiple pictures of the scene was assumed to serve as the input data. New grammar rules needed for the analysis of these pictures were formalized. A procedure for establishing matches between features in the different pictures, and for verifying doubtful features, were devised, as well as procedures for eliminating wrong data and for recovering missing data. Finally, a

procedure for assembling the analyzed data into sets, each describing a single body of the scene, was constructed, including the determination of the bodies' faces and their nature.

A computer program based on this approach was written. The program was able, given real input data, to "understand" the photographed scenes, and to yield plausible descriptions. The program is cautious in the decision making process, at the expense of losing some of the given features. Thus, although the description obtained may be incomplete, no errors are expected. A common and crucial difficulty encountered in the preprocessing stage is telling a T junction from a flat Y or W junction. This can be handled by the program, if desired, by adding the flat Y and W junctions to the junctions that need verification. No restrictions need be placed on the intersection of two curved faces; however, if the intersection is of a degree higher than that of a conic, the determination of an A junction may be problematic.

The main weakness of the test of the program is that the program was not linked to an existing preprocessor. It thus was not subjected to a completed "reality" test. However, if and when such a link-up with a preprocessor is achieved, the power of the main program will grow in the following sense: In the stage of data recovery we made a final judgment as to whether or not a line should be extended to a junction on the basis of the fitted equation and a set of severe restrictions. Once the program is connected to a preprocessor, this can be replaced with a request from the program to the preprocessor to take a closer look in the particular area of interest. The data recovery decision can then be made on the basis of new more precisely determined data.

During the course of the development of the matching procedure, it became apparent that a set of three pictures is a good choice but not optimal. If we use a bigger set of pictures (covering approximately the same visible range), we are likely to need less context support and are almost certain to obtain superior results.

7.2 Specific Contributions

The following are considered to be the specific contributions presented in this dissertation.

1. New grammar rules, crucial for picture analysis, for a family of curved bodies were formulated. These rules, when also applied to polyhedra, complement the set of rules already known and strengthen our ability for machine analysis of pictures containing polyhedral-type bodies.
2. Procedures were developed for matching features in a set of pictures taken from different vantage points, taking into consideration the possible presence of serious errors in feature extraction.
3. A technique was developed for the verification of doubtful features extracted from the pictures, and for the elimination of some possible preprocessor errors.
4. Data correction and recovering techniques were developed having the ability of knowledgeably instructing the preprocessor for further data extraction efforts.

5. A procedure was formulated for assembling the analyzed data from all pictures into sets, each representing a body, and segmenting into subsets, each representing a face. In addition a procedure was given for determining the nature of each face (i.e., whether planar or curved and what type of curve if curved).

6. A program was written and tested for describing scenes of curved bodies on the basis of imperfect data extracted from multiple pictures. Except for information about some general properties, the program has no a priori knowledge of the bodies. The program consists of about 3,500 PL/I statements and was tested against data extracted from photographs of real scenes.

7.3 Directions for Future Research

In our search to provide computers with some kind of 3D perception, there is a strong tendency to imitate the way a human being's perception abilities function. There are two difficulties with that approach. Firstly we do not know how a human's perception really works. All models for human perception presented thus far at best are able to explain only a small part of this ability. Secondly it is questionable as to whether imitating human perception is the best approach for developing machine perception. It is the writer's feeling that a breakthrough in machine perception, which departs from the traditional approach, is needed and due to come.

Meanwhile, following the present approach it appears that considerable improvements in performance can be achieved if the preprocessor is incorporated more intimately into the overall process. It should not stand by itself but rather be an integral part of the whole picture-understanding process. For example, in the process of determining an intersection of lines to locate a junction (a painful task for the preprocessor) other pictures should be consulted, utilizing the geometric relations described in the thesis. The process of extracting features from the pictures should not be serial but rather parallel, and thoroughly exploiting the matching tools described. Also, as already mentioned in Section 7.1, the possibility should be considered of using pictures from more than three vantage points to provide the data for the scene analysis. Finally, thought should be given to broaden and unify the scene description terms. The two present modes of description - faces and their intersections and combinations of simple volumes - should be used together, in addition to other future terms, in a formal scene description language.

REFERENCES

1. G. J. Agin, "Representation and Description of Curved Objects," doctoral dissertation, Computer Science Department, Stanford University, Stanford, CA., October 1972.

2. A. Albano, "Representation of Digitized Contours in Terms of Conic Arcs and Straight-Line Segments," Computer Graphics and Image Processing, 3, (1), 1974, 23-33. (AD 766 315).*

3. R. T. Chien, Y. H. Chang, "Recognition of Curved Objects and Object Assembly," Proc. of the Second International Joint Conference on Pattern Recognition, IEEE publication No. 74 CH0885-4C, Copenhagen, pp. 496-510, August 1974.

4. M. B. Clowes, "On Seeing Things," Artificial Intelligence, 2, (1), 1970, 79-116.

5. R. O. Duda, P. E. Hart, Pattern Classification and Scene Analysis, John Wiley Interscience, New York, 1973.

6. G. Falk, "Computer Interpretation of Imperfect Line Data as a Three-Dimensional Scene," doctoral dissertation, Computer Science Department, Stanford University, Stanford, CA, August 1970. (AD 715 665).

7. A. Guzman, "Computer Recognition of Three-Dimensional Objects in a Visual Scene," doctoral dissertation, M.I.T., Cambridge, Mass., December 1968. (AD 692 200).

8. D. A. Huffman, "Impossible Objects as Nonsense Sentences," Machine Intelligence, 6, ed. B. Meltzer and B. Michie, American Elesevier, 1971, 295-323.

9. A. Rabinowitz, "Reconstruction of Polyhedra from Sets of this Perspective Projections," doctoral dissertation, Dept. of Electrical Engineering, New York University, April 1971. (AD 732 300).

10. U. Ramer, "Computer Edge Extraction from Photographs of Curved Objects," doctoral dissertation, School of Engineering, New York University, N.Y., November 1973. (AD 778 456)

11. L. G. Roberts, "Machine Perception of Three-Dimensional Solids," Optical and Electrooptical Information Processing, ed. J. T. Tippett et. al., M.I.T. Press, Cambridge, Mass., pp. 159-197, 1965.

---

* Items marked with "AD" numbers can be obtained for a nominal charge from NTIS, U. S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

12. R. Shapira, "A Technique for the Reconstruction of a Straight-Edge, Wire-Frame Object from Two or More Central Projections," Computer Graphics and Image Processing, 3, (4), 1974, 318-326. (AD 751 282).

13. R. Shapira and H. Freeman, "A Cyclic-Order Property of Bodies with Three-Face Vertices," Tech. Rept. CRL-42, Div. Applied Science, New York University, August 1975.

14. R. Shapira, "Computer Reconstruction of Quadric-Surfaced Objects from Two or More Visible-Line Projections," Tech. Rpt. CRL-44, Div. Applied Science, New York University, August 1975. (AD-A020 628).

15. Y. Shirai, S. Tsuji, "Extraction of the Line Drawings of Three-Dimensional Objects by Sequential Illumination from Several Directions," Proc. Second International Joint Conf. on Artif. Intelligence, British Computer Society, London, 71-79, 1971.

16. L. B. Smith, "The Use of Man-Machine Interaction in Data-Fitting Problems," doctoral dissertation, Stanford University, Stanford, CA, March 1969.

17. I. Sobel, "Camera Models and Machine Perception," Doctoral Thesis, Artificial Intelligence Memo No. 121, Stanford University, Stanford CA, May 1970. (AD 708 084).

18. D. L. Waltz, "Generating Semantic Descriptions from Drawings of Scenes with Shadows," doctoral dissertation, Artif. Intelligence Laboratory, M.I.T., Cambridge, Mass., November 1972. (AD 754 080).

19. S. A. Underwood, C. L. Coates Jr., "Visual Learning from Multiple Views," IEEE Transaction on Computers, C-24, No. 6, June 1975.

20. P. Woon, "A Computer Procedure for Generating Visible Line Drawings for Solids Bounded by Quadric Surfaces," doctoral dissertation, Dept. of Electrical Engineering, New York University, December 1970. (AD 724 744).

21. P. Woon, H. Freeman, "A Procedure for Generating Visible Line Projections of Solids Bounded by Quadric Surfaces," Information Processing 71, North-Holland Pub. Co., Amsterdam, 1120-1125, August 1972.

INDEX OF DEFINITIONS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR-77-0051	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): COMPUTER RECONSTRUCTION OF BODIES BOUNDED BY QUADRIC SURFACES FROM A SET OF IMPERFECT PROJECTIONS.	5. TYPE OF REPORT & PERIOD COVERED: Interim Sept 9
	6. PERFORMING ORG. REPORT NUMBER: CRL-48
7. AUTHOR(s): Ruth Shapira	8. CONTRACT OR GRANT NUMBER(s): AF-AFOSR-2937-76, AF-AFOSR-2755-75
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Electrical and Systems Engineering Dept, Rensselaer Polytechnic Institute, Troy, NY 12181	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F, 2304 A2
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Office of Scientific Research/NM, Bolling AFB, DC 20332	12. REPORT DATE: Sep 1976
	13. NUMBER OF PAGES: 116
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
scene analysis
artificial intelligence
image processing
3D object reconstruction
impossible objects
picture grammars

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
This thesis describes a computer program for constructing a description of solid bodies from a set of n pictures of the bodies. The bodies are assumed to be bounded by faces which are quadric or planar, and they are restricted to have all their vertices formed by exactly three faces. The pictures are taken from different vantage points, with the restriction that a slight shift in vantage point will not alter the topology of the picture. It is assumed that the program receives outline information from a preprocessor which has extracted this

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 409 952
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

6pg

information from the pictures. The outline information (set of line structures) may be imperfect in that some junctions may be erroneously reported and some lines may be missing. However, all lines due to shadows are assumed to have been eliminated by the preprocessor.

The thesis includes a technique for establishing the validity of the junctions presented by the preprocessor as well as for matching corresponding features in the line structures derived from the different pictures. New grammar rules for line-drawing projections of curved and planar solid bodies are developed. These are useful in parsing the line drawings. They have also led to the definition of a new family of impossible objects. The program works simultaneously with all the available line structures. The parsing of every line structure is supported dynamically by the results gotten thus far from the parsing of the other line structures. Through the parsing of the line structures the use of picture comparison and the application of the grammar rules, many of the preprocessor errors are detected and partly corrected. The program also can provide feedback to the preprocessor in the form of suggestion as to where to look again for lines in the pictures.

The program utilizes the extracted line structures corresponding to the different bodies in all the pictures to determine the set of faces (insofar as possible) for every body. Every face is defined by an ordered set of n-tuples. The n-tuples are the matched lines and junctions in the n different pictures. The three-dimensional coordinates of the vertices and the equations of the faces can then be determined from these n-tuples. The program was written in PL/I and has been tested on several scenes.