SURFACE CONSTRUCTION FROM PLANAR CONTOURS(U) NAVAL
POSTGRADUATE SCHOOL MONTEREY CA P G HOGAN DEC 85

UNCLASSIFIED
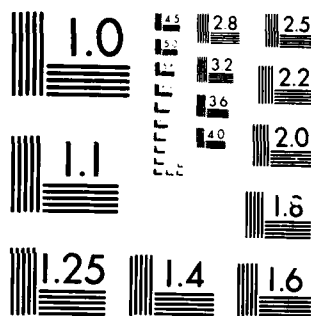
F/G 9/2

NL

END

FILMED

DTIC

MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS 1963 A

NAVAL POSTGRADUATE SCHOOL
Monterey, California

DTIC
ELECTE
MAR 0 5 1986
S D

THESIS

SURFACE CONSTRUCTION
FROM
PLANAR CONTOURS

by

Patrick G. Hogan

December 1985

Thesis Advisor: Michael J. Zyda

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION			1b RESTRICTIVE MARKINGS		
UNCLASSIFIED					
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION/AVAILABILITY OF REPORT		
			Approved for public release; distribution is unlimited.		
2b DECLASSIFICATION/DOWNGRADING SCHEDULE					
4 PERFORMING ORGANIZATION REPORT NUMBER(S)			5 MONITORING ORGANIZATION REPORT NUMBER(S)		
6a NAME OF PERFORMING ORGANIZATION		6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION		
Naval Postgraduate School		52	Naval Postgraduate School		
6c ADDRESS (City, State, and ZIP Code)			7b ADDRESS (City, State, and ZIP Code)		
Monterey, California 93943-5100			Monterey, California 93943-5100		
8a NAME OF FUNDING/SPONSORING ORGANIZATION		8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c ADDRESS (City, State, and ZIP Code)			10 SOURCE OF FUNDING NUMBERS		

			PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

SURFACE CONSTRUCTION FROM PLANAR CONTOURS

12 PERSONAL AUTHOR(S)

Hogan, Patrick G.

13a TYPE OF REPORT	13b TIME COVERED		14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____		1985 December	62

16 SUPPLEMENTARY NOTATION

	COSATI CODES		18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Computer Graphics, Triangulation, Heuristic Methods, Contour Data

19 ABSTRACT (Continue on reverse if necessary and identify by block number) Many scientific and technical endeavors require the reconstruction of a three-dimensional solid from a collection of two-dimensional contours. One method for this reconstruction involves a procedure whereby individual pairs of contours are mapped together to form triangular surface patches. In this paper, we present an algorithm which not only handles mapping situations of simple, closed contours but also mappings of multiple contours per plane and partial contour mappings. Also included is a discussion of algorithm limitations and heuristics.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT			21 ABSTRACT SECURITY CLASSIFICATION	
☑ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS			UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL			22b TELEPHONE (Include Area Code)	22c OFFICE SYMBOL
Michael J. Zyda			408-646-2305	52Zk

DD FORM 1473, 84 MAR · 83 APR edition may be used until exhausted · SECURITY CLASSIFICATION OF THIS PAGE
All other editions are obsolete

Surface Construction from Planar Contours

by

Patrick Gerard Hogan
Lieutenant Commander, United States Navy
B.S., Jacksonville University, 1977

Submitted in partial fulfillment of the
requirements for the degree of

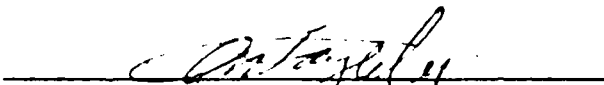MASTER OF SCIENCE IN COMPUTER SCIENCE

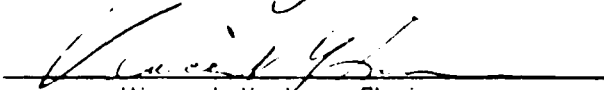from the

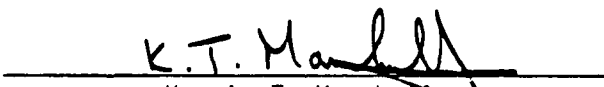NAVAL POSTGRADUATE SCHOOL
December 1985

Author: _____
Patrick Gerard Hogan

Approved by: _____
Michael J. Zyda, Thesis Advisor

_____
C. Thomas Wu, Second Reader

_____
Vincent Y. Lum, Chairman,
Department of Computer Science

_____
Kneale T. Marshall,
Dean of Information and Policy Sciences

2

ABSTRACT

Many scientific and technical endeavors require the reconstruction of a three-dimensional solid from a collection of two-dimensional contours. One method for this reconstruction involves a procedure whereby individual pairs of contours are mapped together to form triangular surface patches. In this paper, we present an algorithm which not only handles mapping situations of simple, closed contours but also mappings of multiple contours per plane and partial contour mappings. Also included is a discussion of algorithm limitations and heuristics.

Accesion For		
NTIS CRA&I	☑	
DTIC TAB	☐	
Unannounced	☐	
Justification		
By		
Distribution /		
Availability Codes		
Dist	Avail and / or Special	
A-1		

3

TABLE OF CONTENTS

LIST OF FIGURES

6

7

ACKNOWLEDGMENT

The first person that I need to acknowledge is Dr. Michael Zyda. It was under his constant supervision that this paper, explaining his algorithm, came to existence. His approach to solving this complex problem of surface construction definitely deserves the attention of the computer graphics community.

Two other individuals were instrumental in the completion of this thesis. The first of these, Lt. Steven Firth, RAN, designed the interactive drawing software package, OZDRAW, which was used exclusively in the creation of all figures contained herein. In the implementation of this algorithm, I used a menu package, developed by Capt. Mike Gaddis, USMC, for ease of program direction. Besides the direct contributions just described, these two colleagues were a constant source of programming assistance and friendship.

Finally, I would like to acknowledge my wife, Elaine, for the patience and support she has given throughout this endeavor.

I. INTRODUCTION

Many scientific and technical endeavors require the reconstruction of a three-dimensional solid from a collection of two-dimensional planar contours. These contours are obtained by some sensor method that samples the original three-dimensional solid along a finite number of parallel planes. The data extracted from that set of parallel planes are contours that lie along the solid's exterior and interior surfaces. The contours on the parallel planes appear as line segments. The line segments are either closed loops, open segments, or single points. The goal of surface construction is the formation of surface patches between contours on adjacent planes such that an approximation of the original three-dimensional solid is formed.

Surface construction by the triangulation of two-dimensional contours is the procedure by which a pair of parallel, planar contours are "mapped together" and then "triangulated" into surface patches that form a surface display. The mapping operation of the surface construction algorithm identifies which contours on consecutive, parallel planes should be mapped together, and exactly which portions of those contours should be connected. The triangulation operation forms the connections between contours on adjacent planes by building triangular tiles between those mapped contours. Each triangular tile is built from an individual line segment from one contour and a single point from the end of a line segment on the other mapped contour. This tiling operation is performed for all line segments in the connect

9

region of each mapped contour. The connect region is that section

of coordinates designated as mappable for a pair of contours on

consecutive planes.

Notationally, this problem has been specified as follows:

"An unknown three dimensional solid is intersected by a finite
number of specified parallel planes.
    . . .

The only information about the solid consists of the
intersections of its surface with the planes. Each of these
intersections is assumed to be a simple closed curve. These curves
are not completely specified; instead, a finite sequence of points
encountered during a positive (counterclockwise) traversal of each of
the original curves is given. The curve segment between two
consecutive points is approximated by a linear segment, called a
contour segment.
    . . .

We reduce the problem of constructing such an approximating
surface to one of constructing a sequence of partial approximations,
each of them connecting two contours lying on consecutive planes.
[Figure 1.1]

Let one contour be defined by the sequence of m distinct contour
points P0, P1, ..., P(m-1), and let the other contour be defined by
the sequence of n distinct contour points Q0, Q1, ..., Q(n-1). We
note that P0 follows P(m-1) and that Q0 follows Q(n-1), and so
indicies of P are modulo m and indicies of Q are modulo n. We wish
to create a surface between the contours P and Q. The surface is
constructed of triangular tiles between these two contours. The
verticies of these tiles are contour points, with the verticies of
each tile taken two from one sequence and one from the other. Thus,
each tile is defined by a set of three distinct elements either of
the form {Pi,Pk,Qj} or {Qi,Qk,Pj}. [Figure 1.2]
    . . .

Each tile's boundary will consist of a single contour segment and
two spans, each connecting an end of the contour segment with a
common point on the other contour." [Ref. 1]

This notational specification of the problem is consistent in all

papers accessible in the literature on surface construction [Ref. 1]

[Ref. 2] [Ref. 3] [Ref. 4].
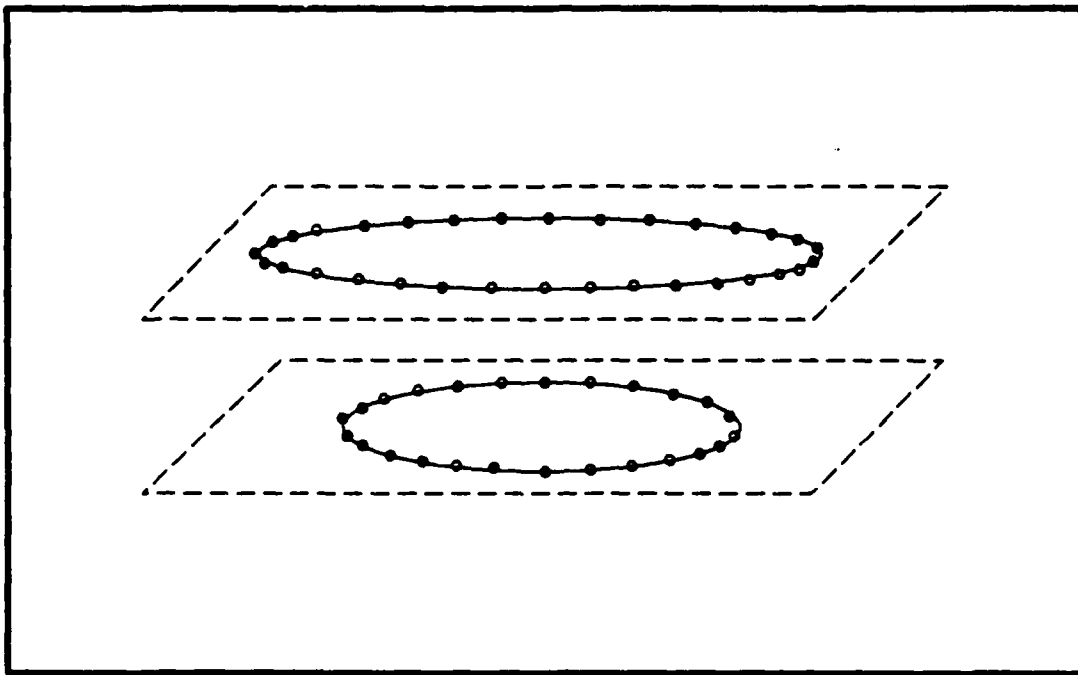
Fig 1.1 - Two contours on adjacent, parallel planes.

* - surface patch defined
by {Qj,Qk,Pk}
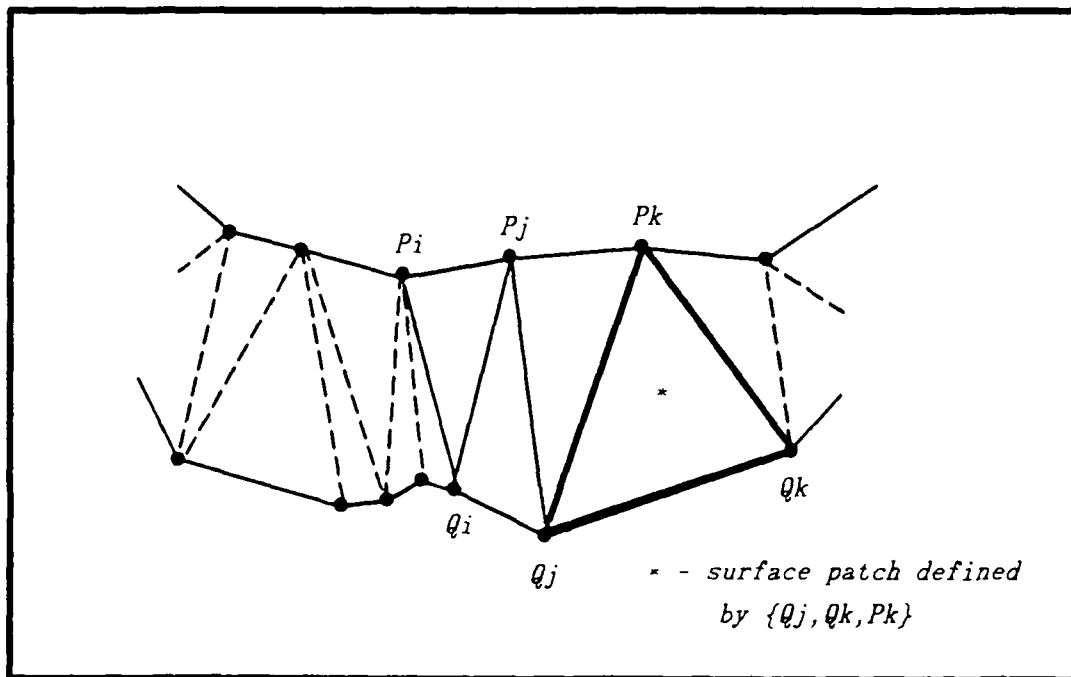
Fig 1.2 - Mapped connections into triangulated
surface patches.

11

The initial emphasis of this paper is a review of the previous algorithms for surface construction. Included in this review is a discussion of each algorithm's capabilities and limitations. After this review, we present a new algorithm for surface construction that is more comprehensive than any that has previously appeared in the literature. Following that discussion, we examine the limitations of our new algorithm.

12

II. LITERATURE REVIEW

In order to understand the nature of our new algorithm for surface construction, it is important that we understand the previous algorithms for surface construction. Four such algorithms have provided the background necessary for the development of our algorithm. These are presented in chronological order.

A. FUCHS ALGORITHM

The first algorithm we examine for the reconstruction of a three-dimensional object from its planar contours was presented by Fuchs in 1977 [Ref. 1]. The problem statement from that article (reproduced in our introduction) has been used in all subsequent papers which build upon the Fuchs algorithm. The major contribution of that article, in addition to the concise statement of the problem, is the presentation of an algorithm capable of connecting simple, closed contours (Figure 2.1).

The problem with the Fuchs algorithm stems from its inability to handle multiple contours on adjacent planes (Figure 2.2). Additionally, no mechanism is provided to handle partial contour mappings or open (non-closed) contours. With respect to the case of multiple contours on adjacent planes, no mechanism is provided to identify which of the contours should be mapped together. The general case for surface construction is to have multiple contours on each plane. The problem with partial contour mappings is that the Fuchs algorithm can only construct a complete triangulation between adjacent

13
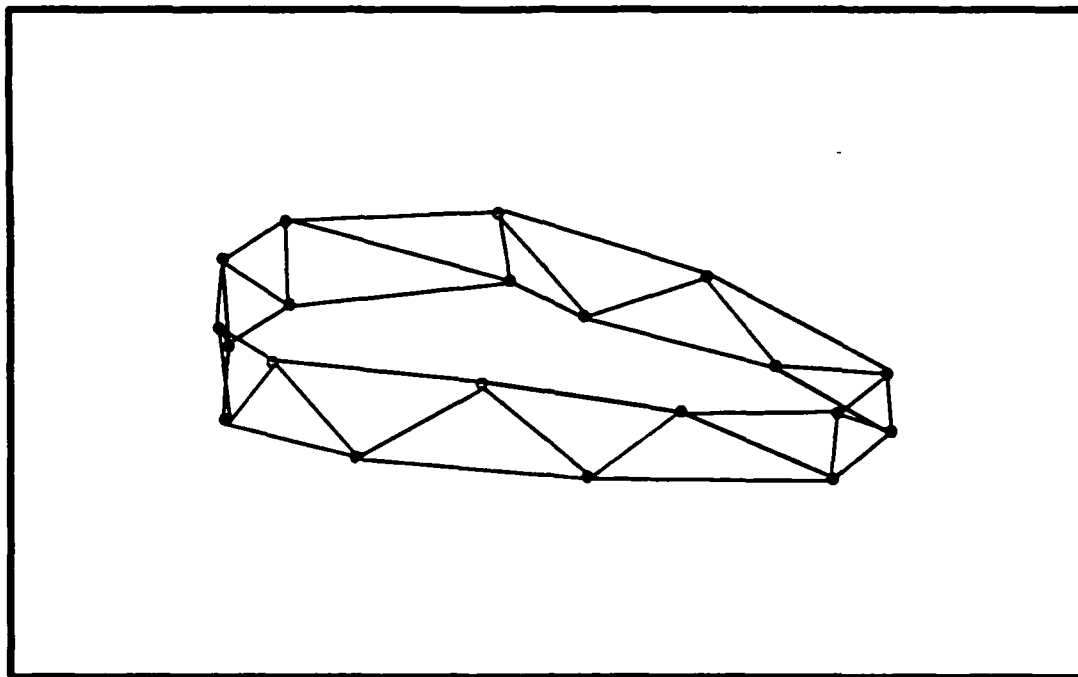
Fig. 2.1 - Triangulated pair of simple, closed contours.
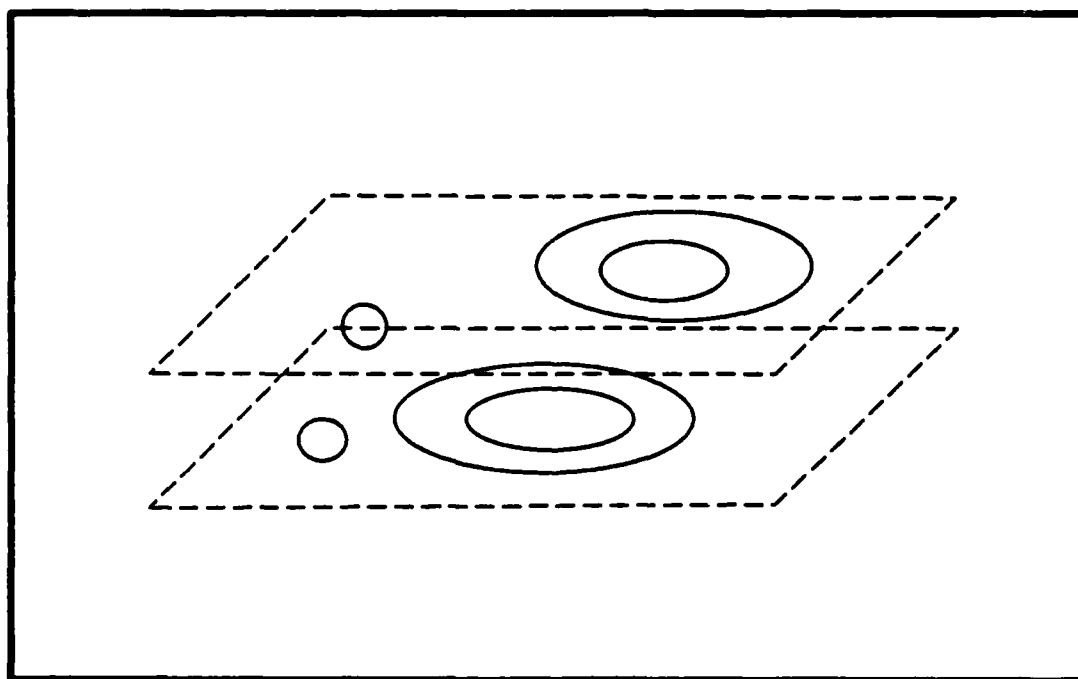
Fig. 2.2 - Example of multiple contours per plane.

14

contours. This limitation disallows partial triangulations of contours. Such partial mappings often are indicated for cases of dissimilarly sized contours. Finally, the problem of open contours can be attributed to algorithm generality. A mechanism that solves the partial contour mapping problem can also solve this problem.

B. CHRISTIANSEN ALGORITHM

In the Christiansen paper, an algorithm is presented which is similar to the Fuchs algorithm. The major dissimilarity is the inclusion of a mechanism to facilitate human interaction for the resolution of highly ambiguous contour mappings. Human interaction is used to determine the relative connection points in the contour mapping process for highly convoluted contours.

Similarly to the Fuchs algorithm, this algorithm can handle mappings of simple, closed contours. It also has capabilities for mapping together simple branches. An example of such branching, seen in Figure 2.3, is a pair of contours on one plane being mapped to a single contour on an adjacent plane. This capability allows the algorithm to handle simple cases of multiple contours on adjacent planes. The method by which this problem is solved is as follows:

1. Introduce a new node midway between the closest nodes on the branches. The Z coordinate of this node is the average of the Z coordinates of the two contour levels (planes) involved.

2. Renumber the nodes of the branches and the new nodes such that they can be considered as being one loop. [Figure 2.4]

3. Triangulate as usual. [Ref. 2: pp. 189-190]

15
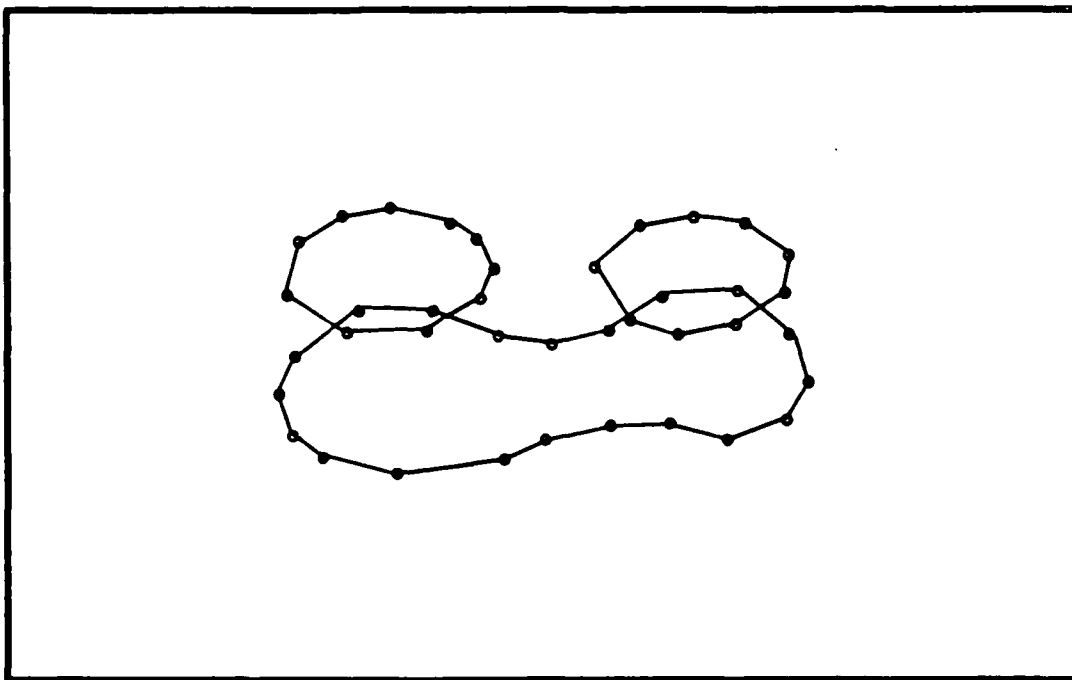
Fig. 2.3 - Simple case of branching.

introduced node
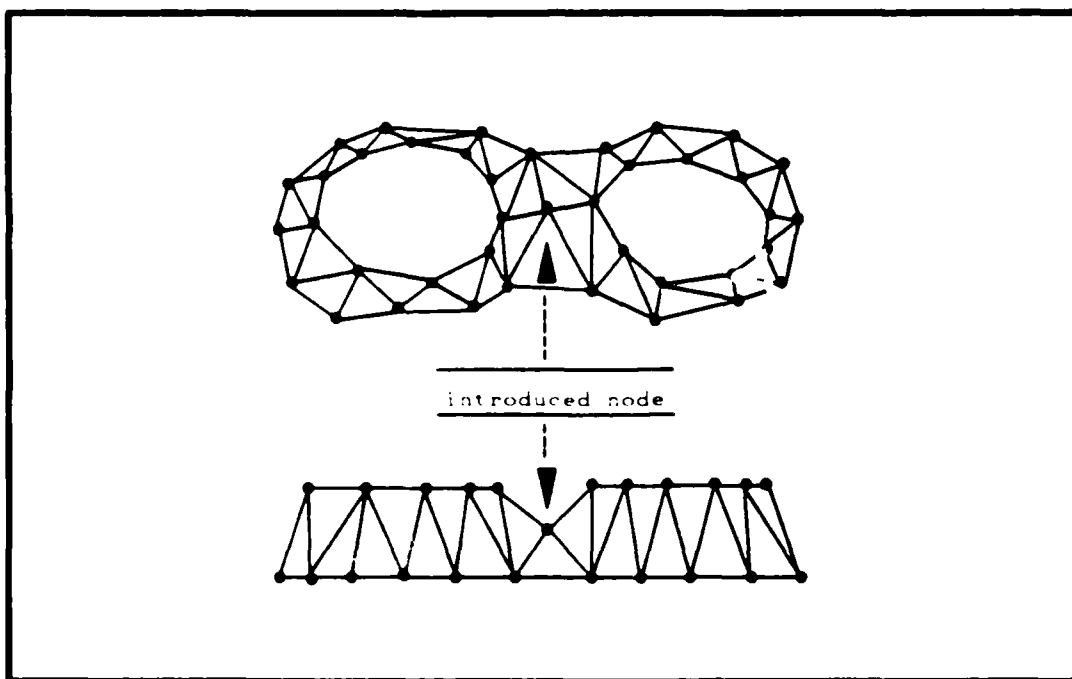
Fig. 2 4 - Triangulation scheme for branching.

16

The Christiansen algorithm is not capable of handling open contours, nor is it capable of handling complex cases of multiple contours on adjacent planes, except by way of expensive human interaction. A final note of interest with respect to this algorithm, is the use of a heuristic for selection of the nodal connections. In cases where contours on adjacent planes are mutually centered and are reasonably similar in size and shape, selection for nodal connection is based on "shortest diagonal" rather than minimum triangular area [Ref. 2: p. 188]. During this operation, one of two nodes is selected to create the next triangular surface patch. The nodes under consideration are the two "next" nodes of each contour. By determining the length of each of the possible diagonals for the surface patch, the connection node is selected based on minimum length.

C. SHANTZ ALGORITHM

The algorithm presented in the Shantz article [Ref. 3] extends the algorithms of Fuchs and Christiansen to handle contour defined objects which are highly branched and have holes. Multiple contours on adjacent planes are handled by

> ". . . first concatenating the contours on each plane into a single large contour using minimum distance links, then performing the mapping between the resulting composite contours." [Ref. 3: p. 242]

Shantz uses the simple, closed contour mechanism of Fuchs to form the connections between the composite contours. Once the connections have been formed, the extraneous ones (due to concatenation) are removed. Some difficult multiple contour cases for this algorithm require human

17

interaction to solve ambiguities. Similar to the Christiansen algorithm, Shantz states that this should be avoided since human interaction is "extremely labor intensive." He cites a case which required 50 to 80 hours of contour splitting, using an interactive cursor, to produce a surface display for the highly convoluted cortex and basal ganglia contours (extracted from the Livingston brain database).

This algorithm, similar to the Christiansen algorithm, is limited in its ability to handle cases of open contours and partial contour mappings. Also, cases of multiple contours on adjacent planes can be handled only when a composite contour can be formed, or when ambiguities are resolved via human interaction.

7. GANAPATHY ALGORITHM

The most recent algorithm for surface construction from planar contours was presented in a paper by Ganapathy [Ref. 4]. That algorithm is essentially an improvement on the Fuchs and Christiansen algorithms for simple, closed contours, without the capabilities described by Shantz. Like Fuchs, Ganapathy assumes a complete mapping of contours, which is not always possible. The improvement over the Fuchs and Christiansen algorithms is attributed to the use of a more computationally expedient heuristic for triangulations.

The problem with the Ganapathy algorithm is that it presents a general solution for handling only the simple case of mapping single, closed contours on adjacent planes. The issues of multiple contour mappings and partial contour mappings are ignored. Additionally, no

18

mechanism for user interaction is provided for resolving mapping ambiguities, further limiting the algorithm to simple cases.

E. SUMMARY

None of the above papers provides a complete solution to the problem of surface construction via the triangulation of contours. What is required is an algorithm with capabilities for multiple contours per plane and partial contour mappings. Additionally, the algorithm should support simple cases of branching and provide a mechanism for human interaction for the resolution of highly ambiguous mappings.

The surface construction algorithm we present handles not only the simple contour mapping problem, but also provides a more comprehensive procedure for solving the multiple contours per plane and partial mapping problems. The only capability lacking from our algorithm is that for handling branching as per the Christiansen paper. A detailed discussion of our algorithm follows, with a presentation of algorithm heuristics and limitations.

19

III. THE ALGORITHM

In the last section, we presented a discussion of previous algorithms for surface construction via triangulation. Here, we present an in-depth discussion of our algorithm by first discussing known input/output data structures. Following this presentation, an overview of the major parts of the algorithm precedes a detailed discussion of the parts.

A. INPUT/OUTPUT SPECIFICATIONS

The problem of surface construction of an object from a set of planar contours, as seen in Figure 3.1, can be reduced to one of constructing the surface triangulations between two adjacent planes. The specification of the problem can be best viewed by detailing the known input data structures:

* total(i)   : number of contours on plane i.

* start(j,i)   : start of contour j on plane i.

* length(j,i)   : number of coordinates in contour j on plane i.

* type(j,i)   : type of contour j on plane i.
                     (CLOSED_LOOP, OPEN_SEGMENT, or SINGLE_POINT)

* interior(j,i)   : value of contour j's interior with respect to
                     the contour line.
                     (HIGH, LOW, or INDETERMINATE)

* coords(XYZ,pointer,i): input coordinates for all contours on plane
                     i. To isolate contour j on plane i:
                          for (pointer = start(j,i) + k - 1),
                     where k = 1, length(j,i).

From the above data, we desire to produce the following output data structures:
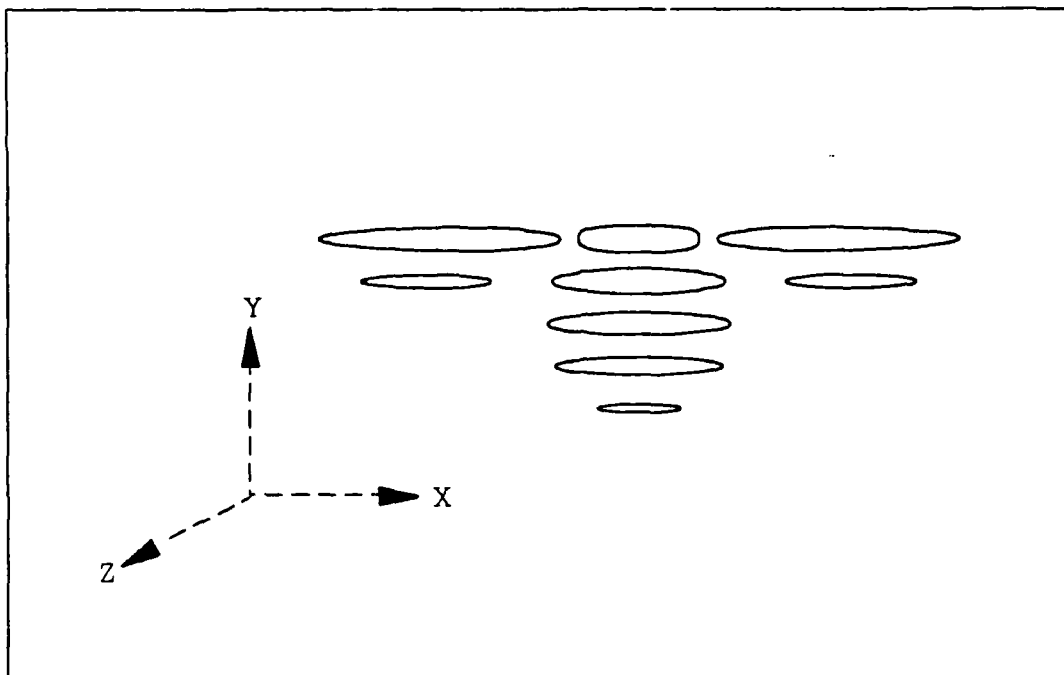
Fig. 3.1 - A partial set of planar contours from a 3D Z^2-orbital of a hydrogen molecule.
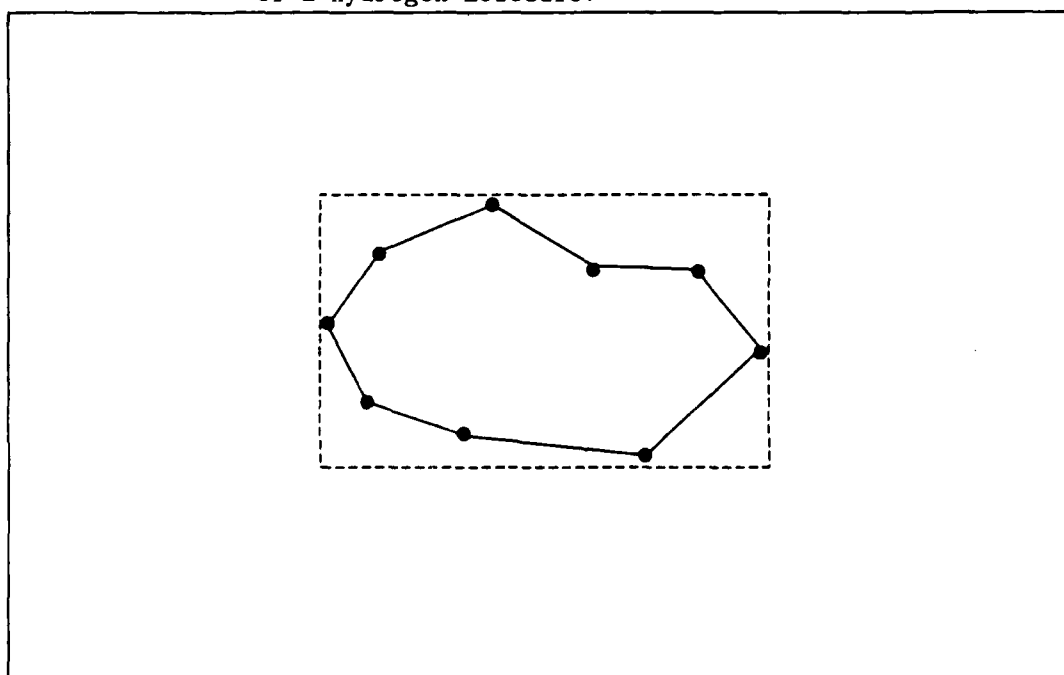
Fig. 3.2 - Two dimensional bounding box used for determining overlap percentage value.

* num_coords   : number of coordinates generated for the two
                        input planes.

* new_coords(XYZ,num_coords): coordinates generated by the surface
                        construction process for the two planes.

* new_conns(num_coords): drawing instructions for each coordinate
                        generated (SETPOINT, DRAWTO, DRAWPOINT).

If the output data is in the form of triangular surface patches, an

alternative data structure is required:

* num_patches   : number of surface patches generated for the
                        input two planes.

* new_coords(XYZ)   : new coordinates generated by the connection
                        process.

* patches(3,num_patches): a 3 by num_patches array of triangles.

B. THE ALGORITHM

    Our surface construction algorithm is composed of the following

outlined steps:

    (1) Input and Inventory Compilation:

         The data structures defining the contours are processed to

         extract the pertinent data. This data includes the number of

         contours per plane, the coordinates defining those contours and

         the types of the contours. Additionally, two-dimensional

         bounding boxes are described about each contour for processing

         consideration in step 2. This compilation of data creates the

         data structures required for surface construction.

    (2) Overlap Determination and Contour Item Mapping:

         In this step of the algorithm, we determine which contours on

         adjacent planes have significant overlap, and which contours'

         exteriors are near. This information is used to designate which

contours should be connected via triangulations. The assignment of overlap is accomplished through the use of a value for the overlap percentage. This value is computed from the areas of the two-dimensional bounding boxes, as seen in Figure 3.2, of each contour. The overlap percentage is used to give priority to contour mappings that have the highest percentage of total overlap area.

In this step of the algorithm, we also perform consistency checks for each contour pair. One such consistency check is executed using the contour interior specification and the overlap percentage value. Contour interior specifications are assigned as the value of a contour with respect to its immediate interior. As such, a contour is LOW valued if it is taken from the exterior of a solid object, such as the skin of an apple. Conversely, a contour is HIGH valued if its immediate interior is non-solid. Using these pieces of information, we are able to eliminate contour mappings of high overlap percentage which result in erroneous approximations of the original three-dimensional solid.

To illustrate the application of this consistency check, let us consider the mapping example for Figure 3.3. Here we are presented with a set of contours taken from a solid cone standing within a hollow cone. In this case, contour 1 on plane 1 has a high overlap percentage with contour 2 on plane 2. However, since contour 2 on plane 2 is low valued with respect
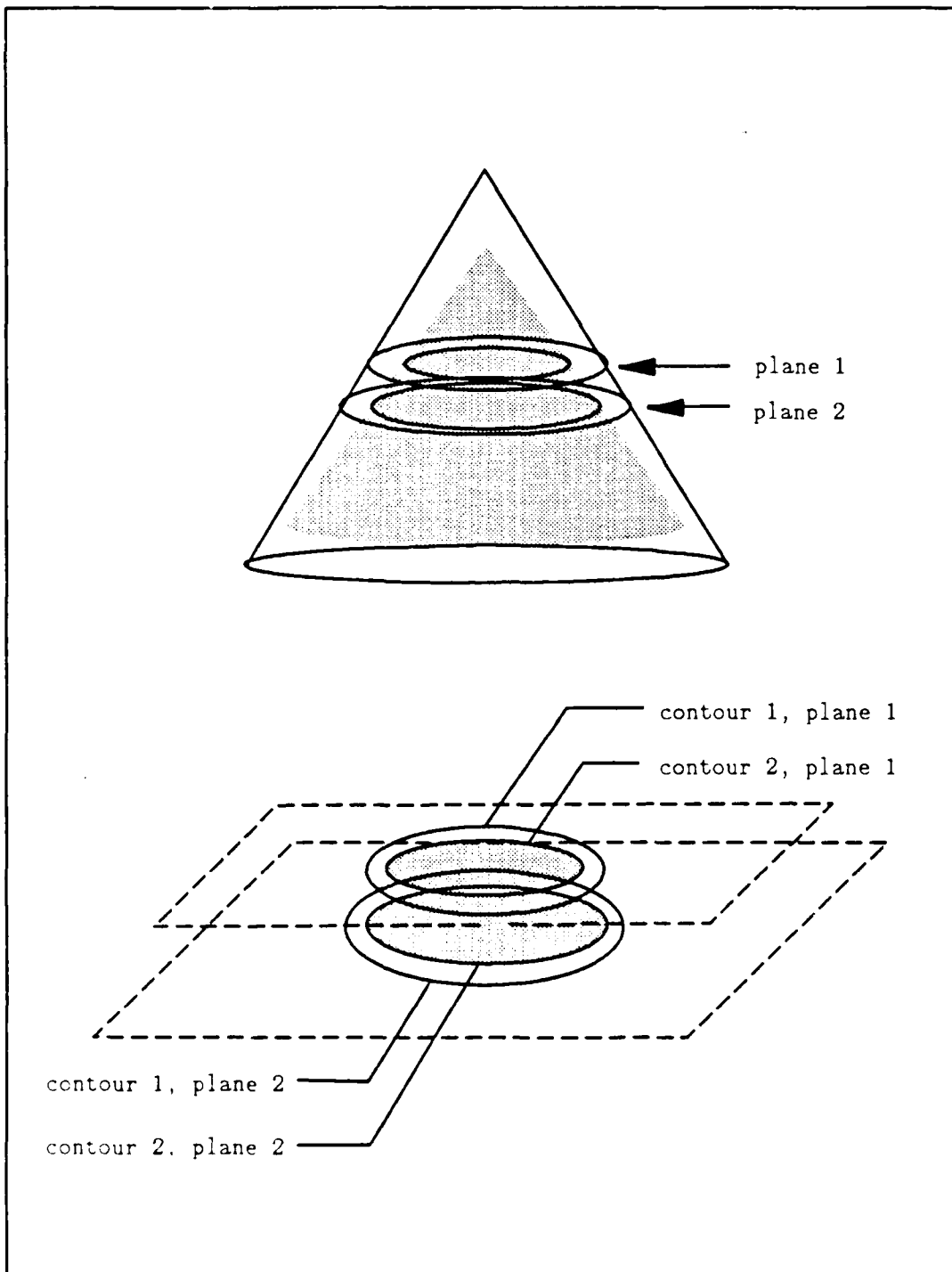
Fig. 3.3 - Example of consistency check using item interior
specifications with overlap percentage values.

to its solid interior and contour 1 on plane 1 is high valued, this mapping can be eliminated.

The interior specifications are also used to determine whether the mapping is interior to interior or exterior to exterior. An interior to interior mapping is one which maps the interior of one contour to the interior of another contour. This form of mapping is indicative of contours taken from a surface with a shallow gradient, i. e., a surface where the mapped contours are of similar size and shape, and where the contours have significant overlap. An exterior to exterior mapping is one which maps the exterior of one contour to the exterior of another contour. This form of mapping is indicative of contours taken from a surface with a steep gradient, i. e., a surface where mapped contours are of dissimilar size and shape, and where the contours overlap percentage is slight. Interior to interior mappings are more common. The exterior to exterior mapping is indicated for cases of two contours with a low percentage of overlap and differing interior specifications (HIGH:LOW, or vice versa).

(3) Form the Coordinate Mapping for each Mapped Contour Pair:

For each coordinate pair from step two, we form a complete coordinate to coordinate mapping. A coordinate mapping is a tentative set of triangulation connections between the contour pairs. There are two procedures for determining this initial coordinate mapping. The procedure used is dependent on the type of mapping found for the paired contours in the

previous step (interior to interior, or exterior to exterior).
Additionally, both procedures try to form triangulation segments
of shortest length, similar to the Christiansen algorithm. A
general statement of this selection process is that we are
trying to map coordinate i of contour n, plane 1 to coordinate j
of contour m, plane 2 such that the distance between the two
coordinates is minimized. An additional qualification to this
distance minimizing criterion is that coordinate connections do
not cross, i. e., coordinates 3 and 4 of plane 1 are not mapped
to coordinates 6 and 5 of plane 2 respectively.

(4) Continuity Recognition:

The coordinate to coordinate mapping formed in step three is
examined for continuity. Continuity, in this case, is defined as
follows. First, we form sets of coordinates from the coordinate
mapping such that each coordinate of each set is constrained
within a coordinate tolerance and within a distance range. The
coordinate tolerance factor is a ratio of the number
of coordinates in the larger contour divided by the number of
coordinates in the smaller contour times a window value. The
tolerance factor is used to group coordinates into a single set
based upon their mapped coordinate number being within plus or
minus tolerance of the last mapped coordinate added to the set.
The tolerance sets formed are then compared for overlapping
distance ranges. Any sets that have overlapping distance ranges
are then merged. The merged set with the smallest distance in

26

it is the set of coordinates for which connections should be generated. All other coordinates are left unconnected.

(5) Mapping Cancellation:

Once we have decided to generate the connections for a part of a contour, we cancel any further mappings to that piece of the contour. This operation is required for partial mappings in which two or more contours on one plane are to be mapped to a single contour on another plane. This cancellation precludes connecting contour points which have already been selected for connection.

(6) Connection Formation:

We generate the coordinates for the triangulation connections specified in step four. "In between" coordinates, coordinates not directly mapped but within the tolerance factor for the connection mapping, are also added to the picture. The goal of the process is to form minimum area triangular surface patches for each segment of the mapped connection region.

1. Input and Inventory Compilation

The input data to the algorithm consists of the contour descriptions for two adjacent planes of a three-dimensional solid. The purpose of this step of the algorithm is to segment this data into separate contour descriptions and to determine the individual characteristics of each contour. Figure 3.4 consists of two adjacent planes, each having three concentric rings of similar shape and continuity. Figure 3.6 consists of two closed loops on each of its
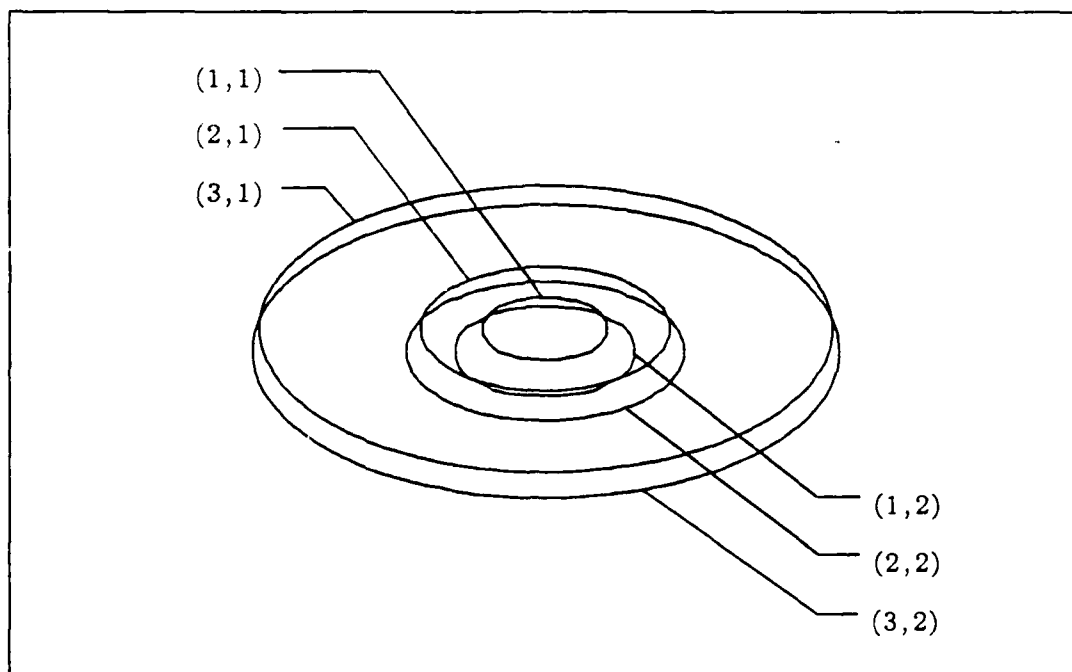
Fig. 3.4 - Example of multiple contours per plane on adjacent planes.
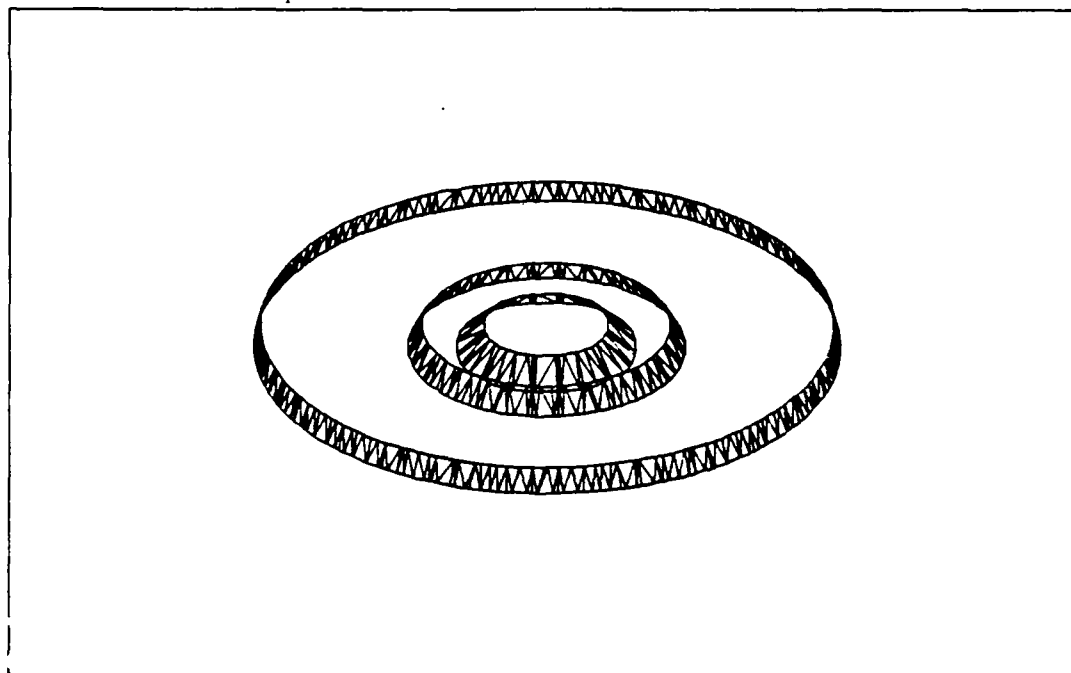
Fig. 3.5 - Connection of Figure 3.4.

28
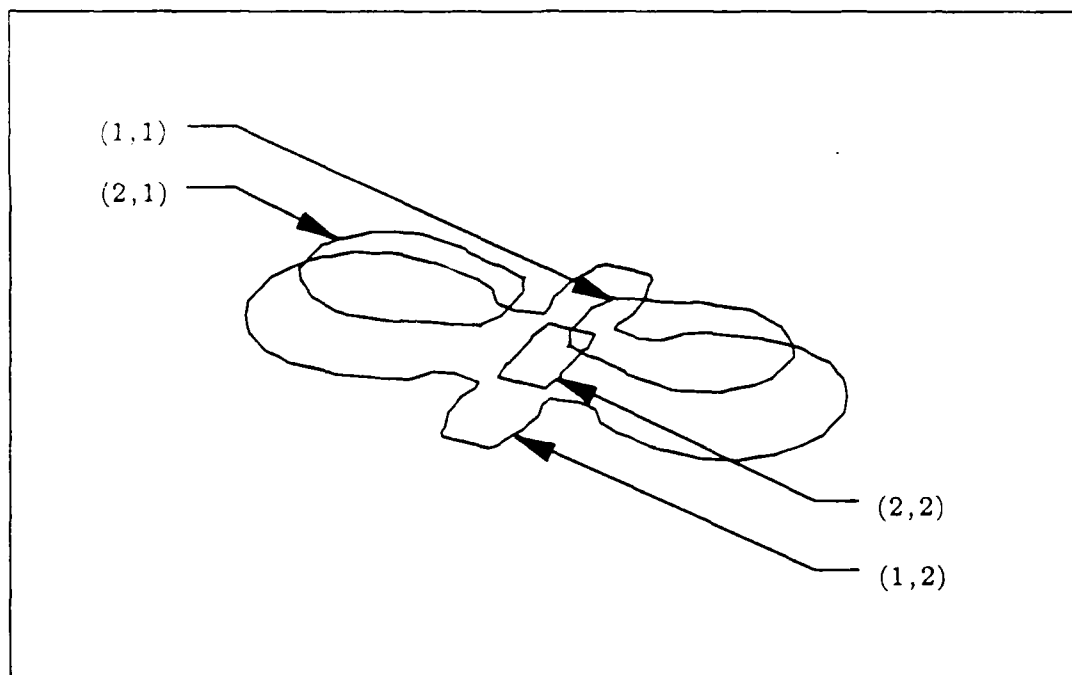
Fig. 3.6 - Example of a set of contours requiring partial mappings and an exterior to exterior mapping; (1,1) and (2,1) to (2,2).
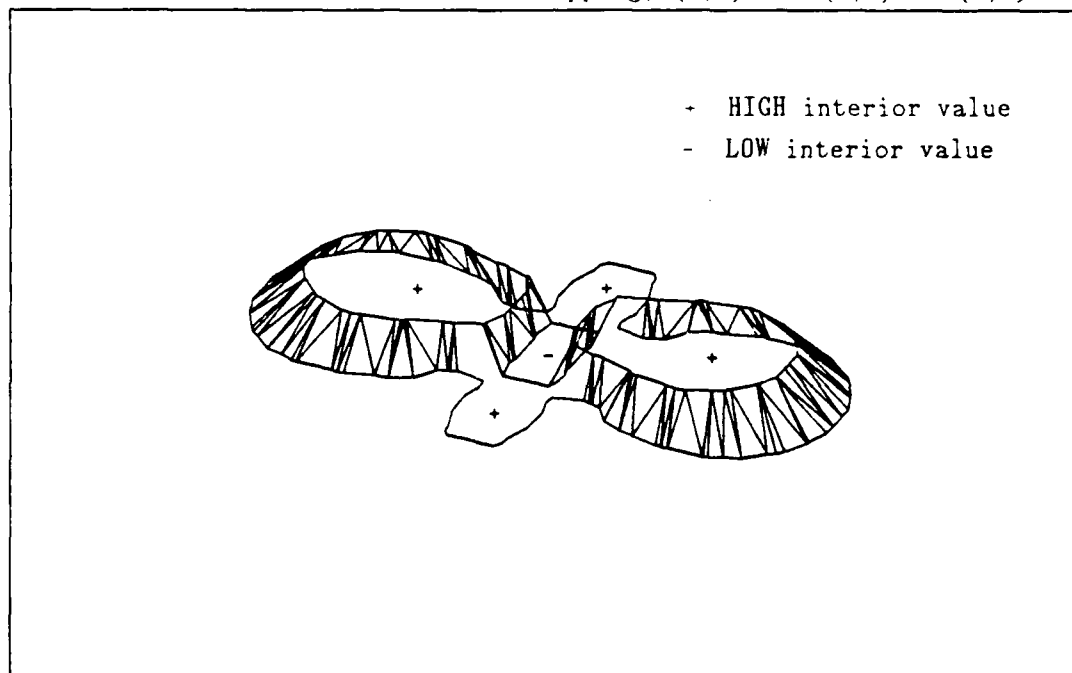
Fig. 3.7 - Connection of Figure 3.6, with contour interior values for each contour.

planes. Plane 1 has two small interior lobes, while plane 2 has one
large surrounding contour with a small interior contour. The contour
descriptions for these figures are composed of:

- the starting coordinate location,

- the total number of coordinates,

- the contour types,

- the interior values, and

- the contours' two-dimensional bounding boxes.

With the exception of the interior values, all of these characteristics
are easily obtainable from the input data.

The procedure necessary to obtain the contour interior
specifications requires an evaluation of the data values lying along
and interior to the contour (see Figure 3.3). If these values are not
contained in the input data, a mechanism needs to be provided to allow
for user specification of contour interior values. The range of
interior values is HIGH, LOW or INDETERMINATE. Without this value the
contour pairing operation encountered in the multiple contours per
plane situation is difficult. In that case, some form of human
interaction is necessary to designate which pairs of contours should be
mapped together. If an interior value is not available, and the
mapping situation is not complex, it can be set to INDETERMINATE
without surface construction degradation.

2. Overlap Determination and Contour Mapping

The overlap determination and contour mapping procedure of the
surface construction algorithm is the process by which tentative

30

contour to contour mapping assignments are made. The contour
characteristics which are necessary for this procedure are the
two-dimensional bounding boxes and the contour interior specifications.
This mapping process is the key component in the disambiguation of
multiply paired contours.

The overlap determination and contour mapping procedure is
accomplished in the following manner. First, the two-dimensional
bounding box of each contour on plane 1 is compared for overlap with
the two-dimensional bounding box of each contour on plane 2. The
coordinates which define these bounding boxes are the minimum and
maximum X and Y coordinates from each of the contour descriptions.
(Additionally, these coordinates are adjusted by a constant value to
promote overlap for exterior to exterior mapping situations.) From
this operation, a table called the overlap table is produced. It is a
two-dimensional table that contains a value for each possible pairing
of contours between the two planes. The value recorded in each table
entry indicates the extent to which each contour overlaps. If there is
no bounding box overlap for a pair of contours, a value of 0.0 is
recorded in the table. If there is overlap, the value recorded in the
table represents the percentage of overlap with the larger of the two
contours. This value is computed by dividing the area of the bounding
box overlap by the area of the bounding box of the larger contour.

After the overlap percentage has been computed for a contour
pairing, it is used in conjunction with the interior specifications to
determine the mapping type for the contour pair. An interior to
interior mapping is indicated when a high percentage of overlap

31

(greater than 10%) exists for a pair of contours. A consistency check for matching interior specifications is performed for every pair of contours that exhibits this high an overlap. The consistency check requires that each contour pair have either HIGH:HIGH, LOW:LOW, or INDETERMINATE:anything (HIGH or LOW) interiors. Contour pairings with high overlap but inconsistent interior specifications result in an adjustment to the overlap table of 0.0 percentage of overlap. An exterior to exterior mapping is indicated when the overlap percentage is low (less than 10%) and item interiors are non-matching. Finally, all contours with low overlap percentages and matching interiors are zeroed in the overlap table.

Figures 3.8 and 3.9 graphically represent the overlap determination and contour mapping for Figures 3.4 and 3.6. Included in these figures are the overlap tables produced by this procedure. The table in Figure 3.8 shows three valid overlap percentages for three different contour pairs: (1,1) - (1,2), (2,1) - (2,2), and (3,1) - (3,2). Four of the entries have been zeroed by the consistency check mechanism. Without this capability, high valued overlap percentages would appear in the overlap table with human interaction required for their disambiguation. The table in Figure 3.9 shows two high overlap percentages and two low overlap percentages. This data indicates that contours (1,1) and (2,1) both map interior to interior with contour (1,2). The low overlap percentages indicate that contours (1,1) and (2,1) map exterior to exterior with contour (2,2).

32
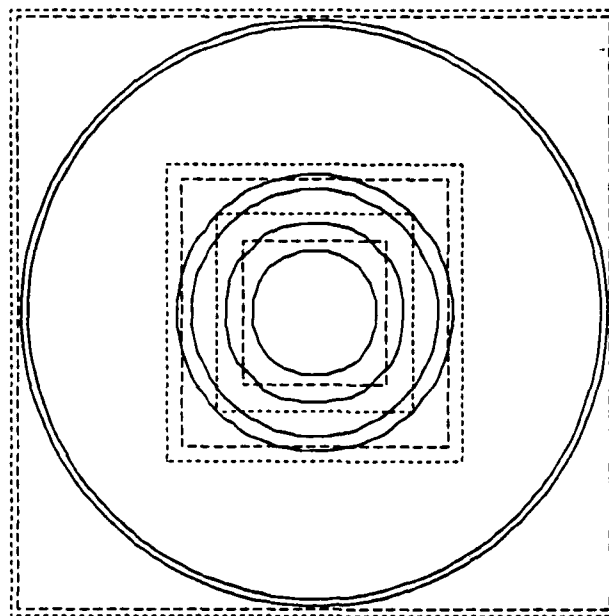
OVERLAP TABLE

Plane 2

		CONTOUR 1	CONTOUR 2	CONTOUR 3
	CONTOUR 1	95.6916	0.0	11.1493
Plane 1	CONTOUR 2	0.0	81.3006	0.0
	CONTOUR 3	0.0	0.0	52.4872

Fig. 3.8 - Bounding boxes and overlap table produced for Figure 3.4
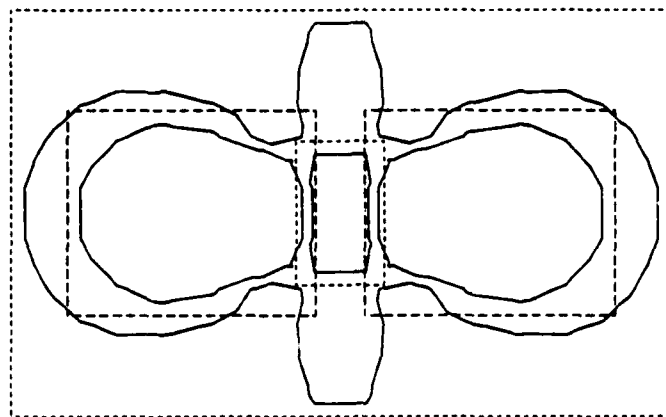
OVERLAP TABLE

Plane 2

	CONTOUR 1	CONTOUR 2
CONTOUR 1	19.0295	5.4386
Plane 1		
CONTOUR 2	19.0295	5.4386

Fig. 3.9 - Bounding boxes and overlap table produced for Figure 3.6

34

3. Form the Coordinate Mapping: Interior to Interior

The coordinate mapping formation procedure for each coordinate pair having a non-zero overlap (in the overlap table) begins with the pair having the largest overlap percentage. All remaining steps in the surface construction algorithm are carried out on this pair before the next pair of contours is considered for mapping. Mapping paired contours is on a largest to smallest overlap percentage ordering. Since exterior to exterior mappings are indicated only in situations where the overlap percentage is low, they are considered for mapping only after all interior to interior mappings have been performed. This study follows that ordering and completes the description of the interior to interior mapping process before considering the separate process necessary for exterior to exterior mappings.

The first operation performed on an interior to interior overlap pair is the determination of which contour is interior to the other. This assignment is accomplished by comparing bounding box areas for the contour pair and designating the contour with the smaller area as interior. Once the interior contour assignment has been made, the center coordinate of that contour's bounding box is computed.

The knowledge of the center coordinate of the interior contour is used in the following manner. For each coordinate of the inner contour, we determine which coordinate of the outer contour is closest to a vector drawn from the center coordinate of the inner contour through the coordinate of the inner contour (see Figure 3.10). We add the qualification that the outer coordinate selected by this procedure must be farther from the center coordinate than the inner coordinate.

35
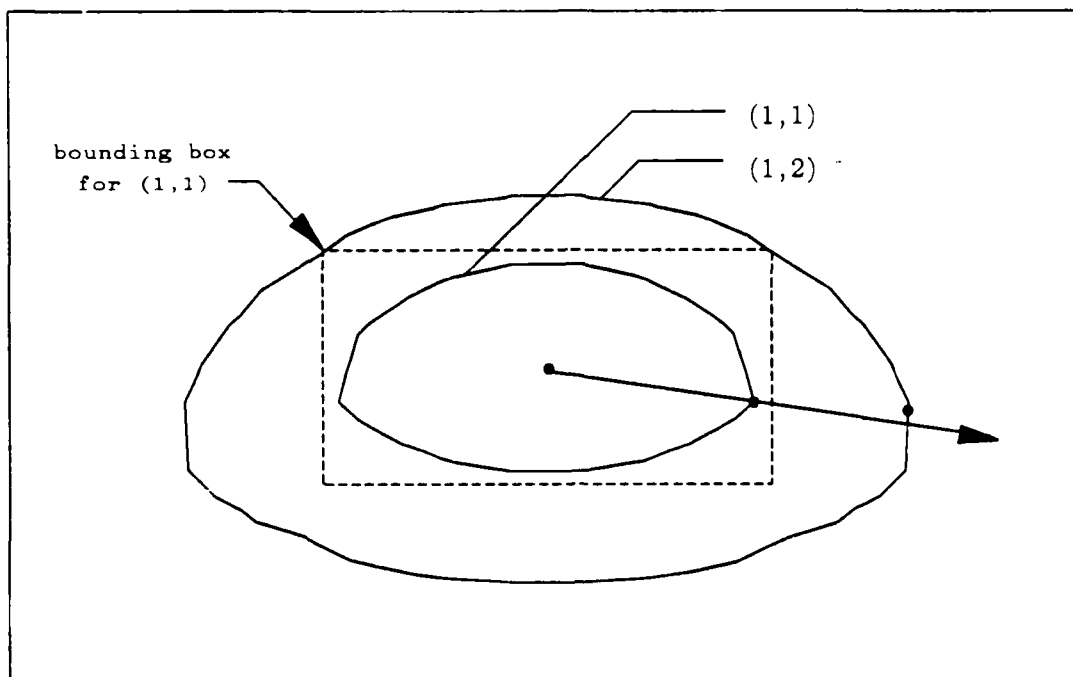
Fig. 3.10 - Vector radiating from center coordinate through the interior coordinate towards the outer contour for tentative mapping

inner coord	outer coord	distance
•	•	•
•	•	•
•	•	•
24	52	0.2001
25	53	0.1789
26	69	0.6067
•	•	•
•	•	•
•	•	•

• large relative change in distance and mapped coordinate number sequencing
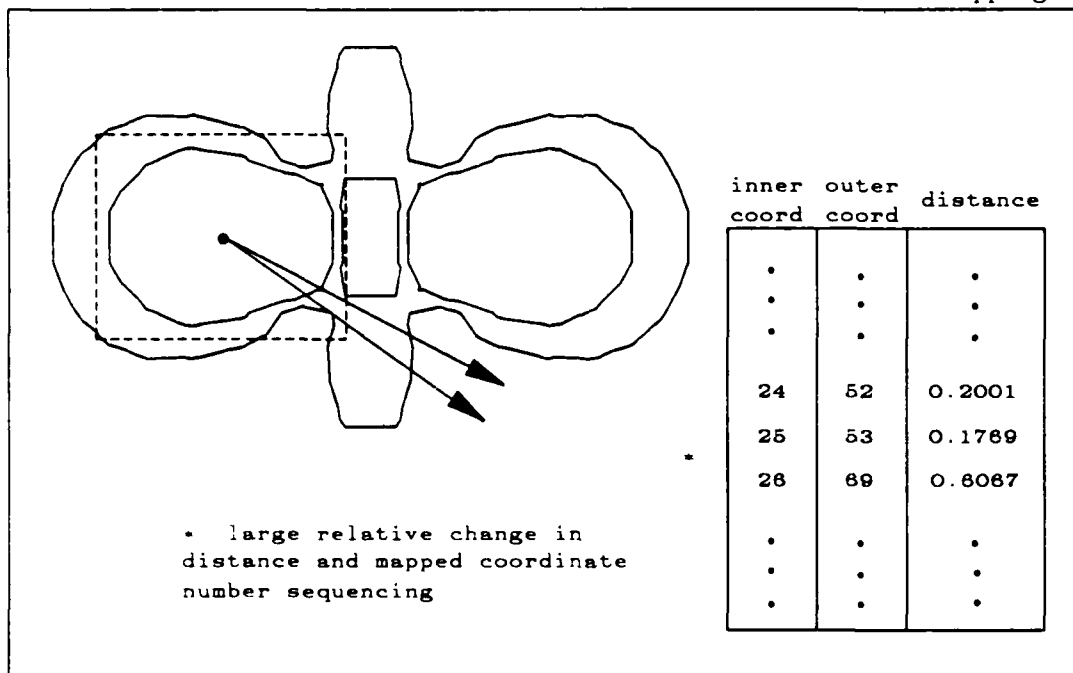
Fig. 3.11 - Example of a case where tentative mapping coordinates and associated distances vary greatly.

Also, the outer coordinate must be on the same side of the vector as the inner coordinate. The outer coordinates selected by this mapping process are recorded as the tentative coordinate map coordinate for each inner coordinate. We also record the two-dimensional distance from each inner coordinate to its tentatively mapped outer coordinate. The resulting data structure contains the mapped outer coordinates with the distance to the inner coordinate to which it is mapped.

The tentative connection map for Figure 3.4 is very good. Due to the similarity in size and shape of the mapped contour pairs, there is very little variation in the mapped distance values and the coordinates selected for mapping appear sequential. On the other hand, it can be seen in Figure 3.11, that large variations in distance values result from this tentative mapping process, and mapped outer coordinates appear with large gaps in the sequencing. This is due to the dissimilarity of the contour pair; the inner contour is relatively simple and much smaller than the convoluted outer contour. The procedure used to delineate a correct mapping from this tentative mapping is described below.

a. Continuity Recognition

The continuity recognition procedure uses the tentative connection map and associated distances for a pair of contours to determine the set of coordinate mappings that should be made for that pair. In the previous step of the algorithm, we produced the tentative connection map for all of the coordinates of the inner contour. This provides a rough approximation of the final mapping, but it must be

37

noted that not all of the inner coordinates need be involved in the final mapping for that pair. The continuity recognition procedure builds sets of coordinate mappings that are both continuous and of similar mapped distance range. These continuity sets are then used to determine the coordinate sequences that should comprise the final connection mapping.

The first step in this procedure is to assign each coordinate pairing of the tentative connection map to an initial continuity set. This is accomplished by stepping through the coordinates of the inner contour in sequence and comparing each coordinates' mapped outer coordinate to the last coordinate added to the last created continuity set. If that coordinate is within a tolerance factor of the last coordinate added, it is added to that set. If the coordinate in question is not within tolerance, a new set is created with that coordinate mapping as its start. The tolerance factor used is a ratio of the number of coordinates in the outer contour divided by the number of coordinates in the inner contour times a window value. (The window value is discussed in the next chapter.)

To illustrate this continuity set assignment, let us refer to the example in Figure 3.11. Here, the tolerance factor is 10 coordinates. The last coordinate considered is inner coordinate number 24. The next coordinate considered is coordinate 25, which is mapped to outer coordinate 53. This coordinate is within the tolerance factor of 10 and is added to the last created continuity set. Inner coordinate number 26 is mapped to outer coordinate 69. This outer coordinate is outside of tolerance with the last coordinate added and

38

therefore, a new continuity set is created with this coordinate mapping as its start.

This initial step of the continuity recognition process is a fast method for aggregating coordinate map pairs. In addition to building the initial continuity sets for the tentative mapping, we keep track of the minimum and maximum mapped distances for each continuity set. These values are used for merging continuity sets in the next step of the process.

The initial sets generated for Figures 3.4 and 3.6 are of particular interest. This step of the continuity procedure placed all of the tentative mappings for the coordinate mapping pairs for Figure 3.4 into a single set. This can be attributed once again to the contours' similar shapes and sizes. On the other hand, coordinate mapping pairs for the mapping (1,1) - (1,2) of Figure 3.6 resulted in 5 initial continuity sets with varying distance ranges (see Figure 3.12).

Once the initial continuity sets have been created for a contour pairing, we merge any sets that have overlapping mapped distance ranges. This merge process reduces the total number of sets and further aggregates the coordinate pair mappings to sets with coordinate number continuity and distance range similarity. In reference to our examples, no continuity set merge was required for Figure 3.4 due to its singular initial continuity set. Figure 3.12 shows the initial sets with distance ranges and the merged sets with distance ranges for the contour pairing (1,1) - (1,2) of Figure 3.6. In that figure, the 5 initial continuity sets have been merged into 3 sets of non-overlapping distance range.

39

Total Initial Sets = 5   Total Merged Sets = 3

Set Name	Min. Dist.	Max. Dist.		Set Name	Min. Dist.	Max. Dist.
1	0.0176	0.1052		1	0.0176	0.1052
2	0.1769	0.2083		2	0.1769	0.2083
3	0.6067	0.6482		3	0.6067	0.6482
4	0.1769	0.2083				
5	0.0176	0.0688				

Fig. 3.12 - Initial continuity sets and merged continuity sets for the contour pair (1,1) - (1,2) of Figure 3.6.
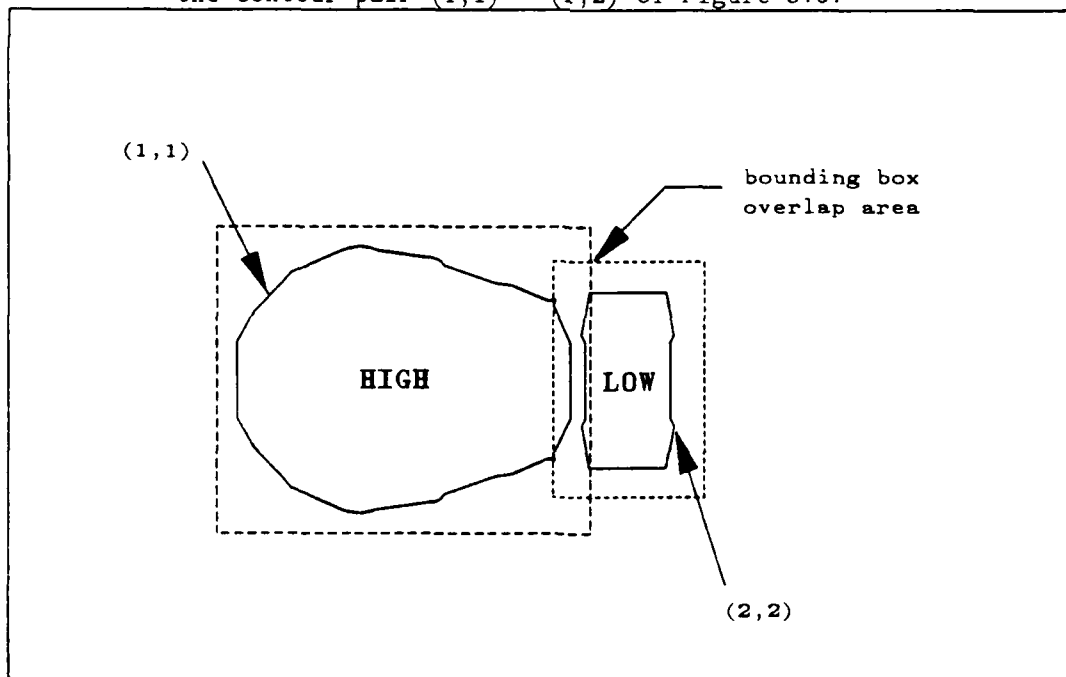

Fig. 3.13 - Bounding box overlap for exterior to exterior mapping. Only the coordinates within the overlap area are mapped.

After we have merged continuity sets, we need to determine which of those sets of coordinate mappings is the one that should be used for connection formation. The choice is clearly the set with the smallest distance range. With this decision, we validate all coordinate pairings that are members of this smallest distance set, and cancel all other coordinate pairings for that set of contours.

b. Mapping Cancellation

The validated coordinate connection map for the contour pair has significance beyond indicating which coordinates need to have connection segments generated. It also indicates "filled" connection positions. By filled we mean that once we have formed connections to a coordinate segment of a contour, that segment should not be reused for any further mapping that occurs for the two current, adjacent planes. This mapping is both checked and recorded at this stage of the algorithm. Mapping cancellation examines the coordinate mappings for which a validated mapping has been assigned. If either of the two coordinates, inner or outer, has been assigned to a higher priority mapping for this pair of planes, then that mapping is cancelled. Once these connections have been struck from the connection map, all remaining validated connections are recorded as filled.

An additional tasking of this cancellation process concerns whether the mapping of either contour resulted in all coordinates defining that contour being included in the mapping. In that case, all other possible pairings with the completely mapped contour are cancelled. This is accomplished by zeroing the overlap on that contour's row or column of the overlap table.

41

c. Connection Formation

When the above steps have been completed for a pair of contours, the remaining process of generating the appropriate line segments is relatively simple. The final coordinate mapping for the inner contour is examined for continuous segments of validated connections. When a continuous segment is defined, the beginning and ending coordinates of that segment (for both the inner and outer contours) are used as boundary pointers for connection formation. The coordinates in between those pointers are stepped through one at a time by a process whose purpose is to generate the minimum area triangular surface patch, as defined in our introduction. The surface patch is formed by using a line segment from one contour as the triangle's base, and a coordinate from the other contour for the triangle's third point. The minimum area selection is accomplished by a procedure that chooses the next line segment between the contours that is both the shortest and within the mapping specified for the two contours. This is identical to the heuristic used by Christiansen in [CHRIS]. Differing coordinate rates between the two contours are taken care of by using the coordinate ratio (from the continuity tolerance factor) between the contours. This ratio allows the process to generate several line segments emanating from a single coordinate when there is a coordinate rate differential between two mapped contours. The lines generated by this procedure for Figures 3.4 and 3.6 are shown in Figures 3.5 and 3.7, respectively.

42

4. Form the Coordinate Mapping: Exterior to Exterior

We begin the exterior to exterior mapping process at the same point of the algorithm where we departed in the description of the interior to interior mapping process. In keeping with our ordering criteria for mapping contour pairs, we examine the contour pair requiring an exterior to exterior mapping which has the highest overlap percentage in the overlap table. All remaining steps of the algorithm are carried out on this pair before the next pair of exterior to exterior contours, in largest to smallest overlap area, is considered.

In Figure 3.13, we are presented with an enlarged view of the bounding box overlap area of the contour pairing $(1,1) - (2,2)$ of Figure 3.6. This area of overlap contains all of the coordinates from both contours which are involved in the connection mapping. The first operation performed on an exterior to exterior mapped overlap pair is the determination of the set of coordinates in both contours that is within the overlap area. The contour with the smaller number of coordinates in the overlap area is used in the formation of a connection mapping between the contour with the larger number of coordinates in the overlap area. The basis for this connection map is the determination for each coordinate (in the smaller coordinate set contour) of the coordinate in the other contour coordinate set that is the shortest distance away. This determination is a simpler version of the distance minimizing process for connection set assignment of interior to interior mappings. The product of this process is the connection map for the pair of contours. The use of continuity sets

43

is not necessary for exterior to exterior mappings due to the relatively small number of coordinates which comprise the connection set.

Once we have generated this connection set, we use the same mapping cancellation and connection formation procedures as described for the interior to interior mappings. The connection formation procedure again uses the connection set mapping to find continuous segments of validated coordinate assignments. The continuous segment thus defined is used to form triangular surface patches for all line segments and coordinates within that segment. The final connection formation for the exterior to exterior mappings, $(1,1) - (2,2)$ and $(2,1) - (2,2)$ of Figure 3.6, are shown in Figure 3.7.

IV. ALGORITHM HEURISTICS AND LIMITATIONS

In the preceeding chapter, we presented an explanation of our algorithm for surface construction. Particular attention was devoted to our algorithm's handling of the multiple contours per plane and partial contour mapping problems. It must be emphasized, however, that our algorithm does not provide a complete solution for all sets of contour surface data. In this chapter, we investigate some of the limitations of our algorithm. In order to do that, we must first discuss the heuristics employed by that algorithm.

A. HEURISTICS

Our algorithm utilizes three heuristics which are essential for the correct connection of planar contours. These heuristics were presented briefly in the last chapter, but we feel it is necessary to explain more fully their application and interaction regarding the contour mapping problem.

1. Overlap Percentage Minimum

In step two of our algorithm, we determine the percentage of overlap between contours on adjacent planes. These percentages are then considered in a consistency check for matching contour interior specifications. The heuristic in question, the overlap percentage minimum, is applied in the final phase of this contour pairing procedure. Contour pairs having an overlap percentage value above the overlap percentage minimum, with matching interior specifications, are designated for interior to interior mapping. Contour pairs having

45

non-zero percentages below the overlap percentage minimum, with non-matching interior specifications, are designated for exterior to exterior mapping. All other contour pairs are disregarded.

The value we have utilized for the overlap percentage minimum is ten percent. We found, through experimentation, that the assignment of this value resulted in the greatest number of correct contour pairings. Some contour pairs which should be mapped, however, are disregarded for mapping because of this selection (of 10%) for the overlap percentage minimum. In Figure 4.1, we are presented with an example of such a situation. In that figure, we have a pair of contours with matching interior specifications (HIGH:HIGH), and having an overlap percentage less than ten percent. By our heuristic, this contour pair would not be considered for mapping, and would remain unconnected.

One possible solution to this problem would be a mechanism which used a relaxation procedure to force a mapping between the pair of contours. This mechanism could be selected by the user to designate contour pairs for mapping which would otherwise be disregarded. If applied to the mapping situation of Figure 4.1, an appropriate connection could be generated.

2. Boundary Tolerance Percentage

The next heuristic to be discussed comes into play in the initial two steps of our algorithm. Specifically, the two operations involved are the determination of contour item two-dimensional bounding box values, and the usage of those values for overlap determination. As previously discussed, exterior to exterior contour mappings are
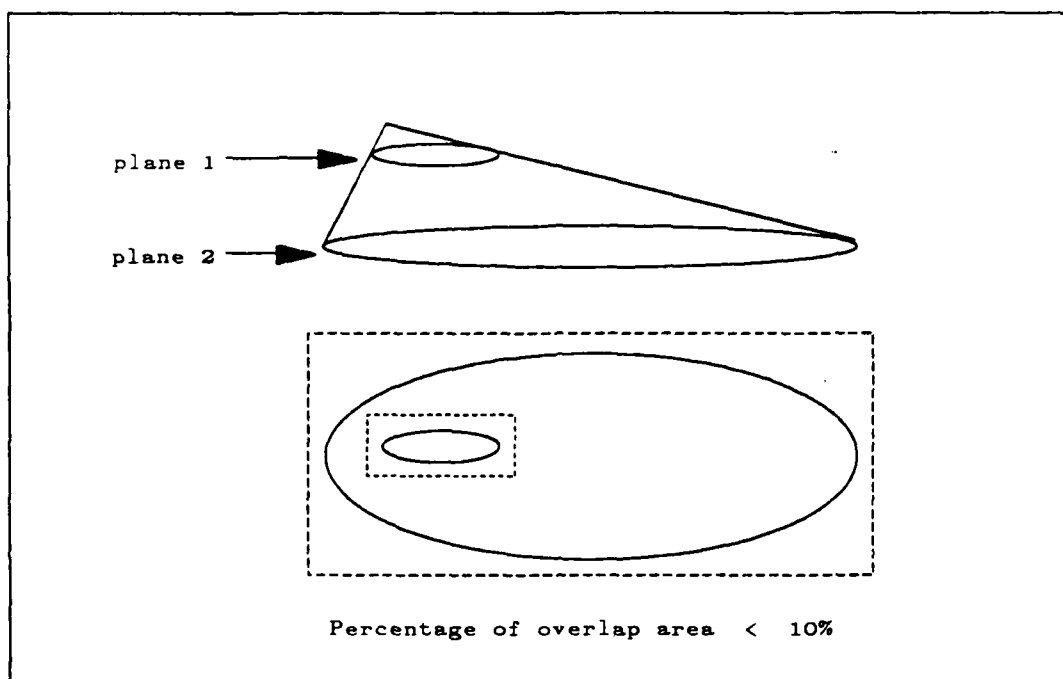
46

Percentage of overlap area < 10%

Fig. 4.1 - Example of a contour pair which should be mapped, but
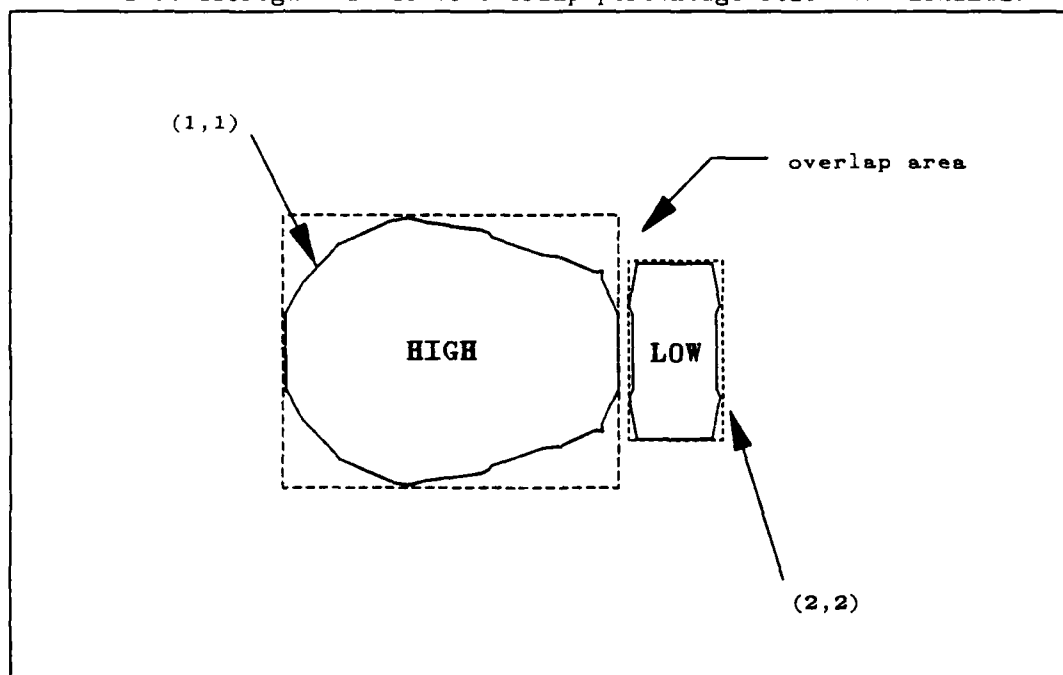would be disregarded due to overlap percentage below the minimum.

Fig. 4.2 - Example of contours' 2D bounding boxes created strictly
from the min and max X and Y coordinates. Resulting overlap = 0.

indicated for pairs of contours with a low percentage of overlap and non-matching interior specifications. In the initial development of our algorithm, we utilized the minimum and maximum X and Y coordinates of the contour to describe its bounding box. We found, however, that in the majority of cases, these values resulted in zero percentage of overlap between contours which should be mapped. An example of this limiting of bounding box values can be seen in Figure 4.2. In that figure, we are presented with the contour pair from Figure 3.13. In this example, it can be seen that limiting the bounding boxes for these two contours to their respective minimum and maximum X and Y coordinate values results in zero percentage of overlap. This is an unsatisfactory situation since the contours should be mapped.

To remedy this situation, we adjust the bounding box values by a percentage to promote mappings in situations similar to that of Figure 4.2. Once again, we are presented with the opportunity to utilize a relaxation procedure, prompted via user intervention, for mapping situations not included by this heuristic. A mechanism could be provided allowing the user to designate the bounding boxes for individual contours, and thereby force a mapping between the desired set of contours.

3. Tolerance Multiplier

In an interior to interior mapping situation, a tolerance factor is used for the determination of the initial continuity set assignments. This tolerance factor is a ratio of the number of coordinates in the outer contour divided by the number of coordinates in the inner contour times a window value. The window value is a

constant which we found necessary for the selection of appropriate mapping connections. We chose to utilize a tolerance factor in this step of our algorithm, as well as in the connection formation procedure, because it provides an inexpensive means for restricting the search space in the selection of mapping connections.

B. LIMITATIONS

In the preceeding chapter, we demonstrated the capabilities of our algorithm, with emphasis on its handling of the problems of multiple contours per plane and partial contour mappings. We have found, however, that there exist contour mapping situations which cannot be handled by our algorithm.

The first mapping situation concerns simple branching of one contour on one plane to two or more contours on an adjacent plane (see Figure 2.3). In this situation, we found that the application of our algorithm produces an incomplete contour mapping due to missing data. One possible solution to this mapping problem is the inclusion of a procedure for creating an introduced node similar to that described in the Christiansen [Ref. 2] paper. This special case procedure could be selected automatically, or initiated via user interaction.

The next limitation of our algorithm manifests itself in situations where highly convoluted contours, with extreme narrowings, are mapped interior to interior. The problem here is due to the interior to interior algorithm's dependence on the overlap region bounding box's center coordinate for the tentative coordinate mapping. For the portion of the contour near the center coordinate, the tentative

49

coordinate is fairly good. For the portion of a contour on the other side of a narrowing, where the center coordinate is no longer central, the tentative mapping is erroneous. The problem comes when the tentative mapping is so bad that the continuity recognition procedure fails, and contour segments are incorrectly left unconnected.

The solution to this problem is fairly simple and within the purview of our algorithm. If the convoluted contour is segmented at the extreme narrowings, it is possible to treat each open segment of the original contour as a separate contour. Using the original algorithm, we can generate centers for each new contour, and hence coordinate mappings, which result in a more correct approximation of the original three-dimensional object. The only capability lacking from our present algorithm is a mechanism for partitioning the original convoluted contour. This mechanism could be either user specified or automatic. The user specified option is favored due to the computational expense involved for automatic contour segmentation.

The next limitation also concerns interior to interior contour mappings. In situations where sections of a contour tend to be near parallel with the vector drawn from the center coordinate of the inner contour, erroneous mappings result. An example of this situation can be seen in Figure 4.3. For those segments of the outer contour which are nearly perpendicular to the tentative connection vector, an appropriate connection map is generated. As the contour segment becomes more parallel to this vector, the tentative connections generated begin to falter.
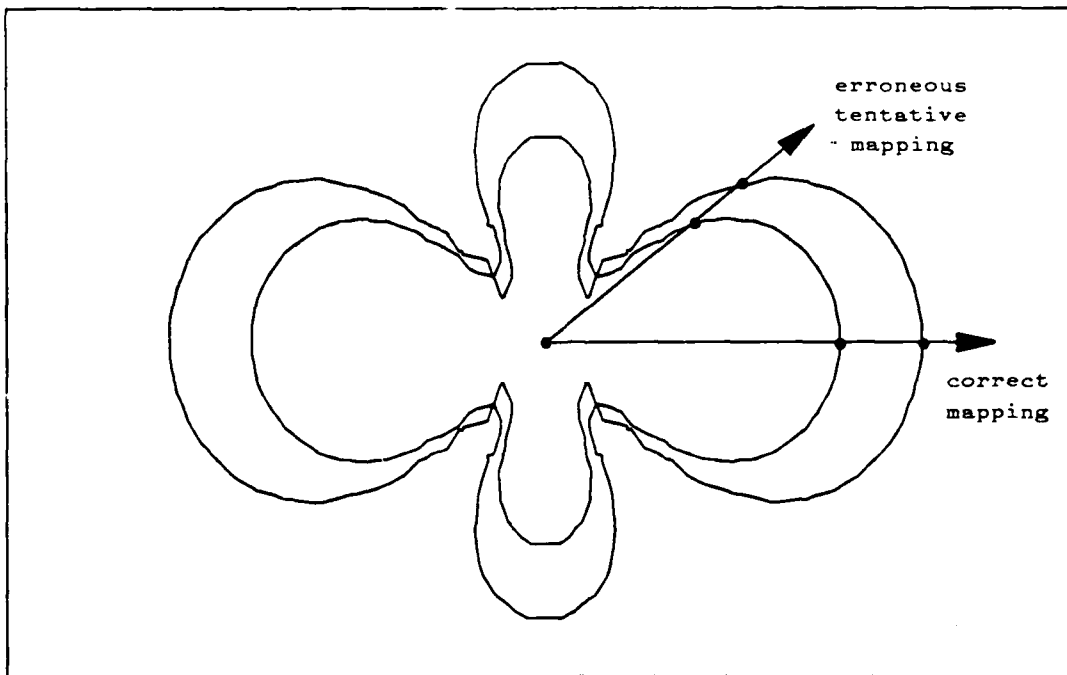
Fig. 4.3 - Example of situation resulting in an erroneous tentative
coordinate mapping where contour segment becomes near parallel
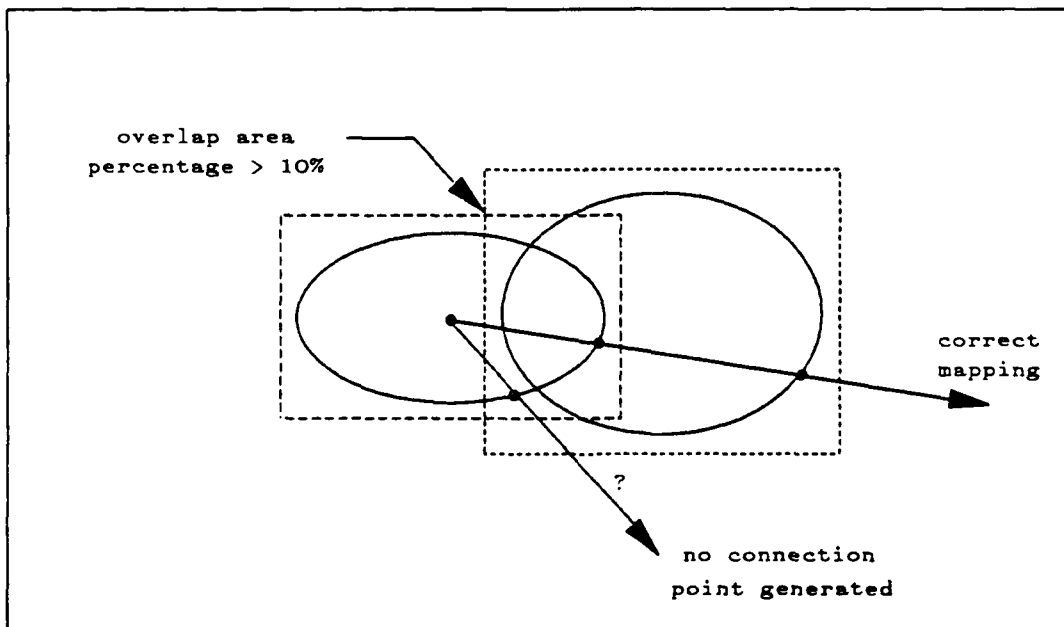with the tentative connection vector.

Fig. 4.4 - Example of a situation where two contours are mapped
interior to interior which would result in an incomplete mapping.

The remedy to this problem is very similar to that for the previous situation involving highly convoluted contours with extreme narrowings. Segmentation of the original contour into several open segments, which could be mapped separately, would greatly improve the quality of the tentative coordinate mapping. Once again, user intervention is the preferred method of contour segmentation.

The final problem situation to be discussed concerns interior to interior mappings where the inner contour is not contained in the outer contour. This situation would result from contour data taken from a torus, such as a doughnut. An example is illustrated in Figure 4.4. The problem with this mapping situation results from the use of the tentative connection vector emanating from the center of the inner contour. Since the center coordinate of the inner contour is displaced from the center coordinate of the outer contour, tentative mappings are generated only for that section of the outer contour which is on the same side of the tentative connection vector (see Figure 4.4). The net result is a partial mapping of two contours which should be totally connected.

A practical solution to this mapping problem, which could be readily adapted to our algorithm, is described in the Christiansen paper [Ref. 2]. In mapping situations where contours to be mapped are not mutually centered, Christiansen recommends a translation procedure onto a unit square, centered at $(0,0)$. The principle of this process is to translate the two contours in such a manner that they become mutually centered within the unit square. Application of the interior to interior algorithm at this point would result in the desired

52

mappings. Tentative mappings would be generated for the contours'
original coordinates, thus allowing the appropriate connections to be
formed in the final step of the algorithm.

C. SUMMARY

It has been the purpose of this chapter to investigate the
limitations of our algorithm, and provide practical solutions where
possible. Additionally, to a lesser degree, the heuristics employed by
our algorithm have been explained to increase the understanding of the
reader. We feel that our algorithm provides a solution to the multiple
contours per plane and partial mapping problems, but must concede that
it is not a total solution to the problem of surface construction from
planar contour data.

V. CONCLUSION

It has been the goal of this paper to describe a new algorithm for the surface construction of a three-dimensional object from a set of that object's planar contours. The greatest part of this paper has been devoted to the capabilities of our algorithm, specifically, its handling of the multiple contours per plane and partial contour mapping problems. We have included a discussion of the limitations encountered thus far by our algorithm for specific problem mapping situations.

In view of the limitations presented, we must comment that our algorithm does not, in its present form, provide a complete solution to the contour mapping problem. Further development is required to alleviate the problem areas discussed in Chapter IV. It is probable, however, that the correction of these algorithmic shortcomings will not ensure a complete solution to the contour mapping problem. We foresee that in some situations either user interaction or an alternative approach may be required.

APPENDIX - PSEUDO-CODE ALGORITHM DESCRIPTICN

A pseudo-code description of our surface construction algorithm
follows:

FACE_IT
{
   Input the coordinates for two adjacent planes. Make a local copy
   of the coordinates.

   DELINEATE_INVENTORY
   {
      Take inventory of the contours in the coordinate sets. This
      inventory determines the total number of contours for each plane
      and records where each contour begins and ends.
   }

   TYPE_INVENTORY
   {
      We determine the contour type of each contour in each plane.
      There are three possible types: CLOSED_LCCP, OPEN_SEGMENT, and
      SINGLE_POINT.
   }

   BOUND_INVENTORY
   {
      Determine the rectangular, two-dimensional boundary of each
      contour. Increase those boundaries by a constant to increase
      the possibility of detecting appropriate exterior to exterior
      mappings.
   }

   INTERIOR_DETERMINATION
   {
      Determine whether the interior of each contour is HIGH or LCW
      valued with respect to the current contour level. This value
      can be assigned interactively in cases where the information
      to make this determination is not available. These values
      are used in a consistency check for selection of contour pairs
      for mapping.
   }

   OVERLAP_DETERMINATION
   {
      Compute the overlap table for the contours of both planes. The
      values in the table are the percentage of overlap for each
      possible contour pair on the adjacent planes. If there is no
      overlap, a value of 0.0 is recorded.

Contour mapping types are also assigned at this step of the
algorithm. Contour pairs with a HIGH percentage of overlap,
matching interior specifications (HIGH:HIGH, LOW:LOW, or
INDETERMINATE:anything) are assigned interior to interior type
mapping. Those pairs with a non-zero overlap percentage, below
10%, with non-matching interiors are assigned exterior to
exterior mappings. All other contour pairings are zeroed.
}

CONNECTION_DETERMINATION
{
    This step of the algorithm orders the pairs to be mapped, and
    forms connections for the assigned types of contour mappings.
    This step is detailed below.
}

}   /* end of FACEIT */

CONNECTION_DETERMINATION
{
    while .true.
    {

    Find the largest overlap percentage in the overlap table. If the
    largest value = 0.0 then QUIT.

    If the contour mapping indicated by this largest overlap value is
    exterior to exterior . . . .

    EXTERIOR_TO_EXTERIOR_MAPPING
    {
        Determine the set of coordinates in each contour that are in the
        overlap area.

        For the contour of the overlap pair that has the least number of
        coordinates, find the minimum distanced coordinate of the other
        contour.

        Assign all coordinates within the overlap region to the connec-
        tion set.

    }   /* end of EXTERIOR_TO_EXTERIOR_MAPPING */

else
        /* perform an interior to interior mapping */

    INTERIOR_TO_INTERIOR_MAPPING
    {
        Determine which contour of the pair is interior. This assignment

56

is based upon which contours' bounding box is smallest.

Compute the center coordinate of the inner contour's bounding box. Check to make sure that this point is inside the contour. If it is not, the contour needs to be partitioned.

For each coordinate of the inner contour, determine the coordinate of the outer contour which is closest to a vector drawn from the center coordinate through the coordinate of the inner contour. Store the coordinate as the connection map coordinate for the inner contour. Also, record the mapped distance from each inner coordinate to its mapped outer coordinate.

```
RECOGNIZE_CONTINUITY
{
    /* Determine continuity sets in the two contours using the
       the connection map and associated distances. */

    INITIAL_CONTINUITY_SETS
    {
        Assign the coordinates of the connection map to a con-
        tinuity set based upon whether each consecutive coordinate
        is within a coordinate tolerance factor. This tolerance
        factor is a ratio of the number of coordinates in the
        outer contour divided by the number of coordinates in the
        inner contour.
    }

    INITIAL_SET_DISTANCE_RANGES
    {
        Determine the minimum and maximum distance ranges for each
        of the continuity sets.
    }

    CONTINUITY_SET_MERGE
    {
        Merge any continuity sets that have overlapping distance
        ranges, maintaining the distance range for any merged set.
    }

    CONNECTION_SET_ASSIGNMENT
    {
        Assign coordinate connections for the coordinates of the
        merged continuity set that contains the smallest distance.
        All other continuity sets are left unconnected.
    }

}   /* end of RECOGNIZE_CONTINUITY */

}   /* end of INTERIOR_TO_INTERIOR_MAPPING */
```

```
MAPPING_CANCELLATION
{
    Examine the coordinate mappings for which a connection has been
    assigned.  If either of the two coordinates, inner contour or
    outer contour, has been used in a previous, higher priority
    mapping for this pair of planes, that coordinate mapping is
    cancelled.  Once these filled connections have been struck from
    the connection map, all remaining validated connections are
    recorded as filled.
}


CONNECTION_FORMATION
{
    Generate the connections for the validated coordinate map.  This
    is accomplished by stepping through the connection map and
    forming coordinate connections where indicated.  In between
    coordinates, those not directly mapped but within the tolerance
    factor for the connection mapping, are also added to the picture.
    The goal of the connection process is to form minimum area'd
    triangular surface patches.
}

}   /* end while .true. of CONNECTION_DETERMINATION */

}   /* end of CONNECTION_DETERMINATION */
```

58

LIST OF REFERENCES

1. Fuchs, H., Kedem, Z. M. and Uselton, S. P., "Optimal Surface Reconstruction from Planar Contours," Communications of the ACM, v. 20, pp. 693-702, October 1977.

2. Christiansen, H. N. and Sederberg, T. W., "Conversion of Complex Contour Line Definitions into Polygonal Mosaics," Computer Graphics: A Quarterly Report of SIGGRAPH-ACM, v. 12, pp. 187-192, August 1978.

3. Shantz, M., "Surface Definition for Branching, Contour-Defined Objects," Computer Graphics: A Quarterly Report of SIGGRAPH-ACM, v. 15, pp. 242-266, July 1981.

4. Ganapathy, S. and Dennehy, T. G., "A New General Triangulation Method for Planar Contours," Computer Graphics: A Quarterly Report of SIGGRAPH-ACM, v. 16, pp. 69-74, July 1982.

BIBLIOGRAPHY

Aho, A. V., Hopcroft, J. E. and Ullman, J. D., The Design and Analysis of Computer Algorithms, Reading, Massachusetts: Addison-Wesley Publishing Co., 1974.

Boissonnat, J., "Geometric Structures for Three-Dimensional Shape Representation," ACM Transactions on Graphics, v. 3, October 1984.

Foley, J. D. and Van Dam, A., Fundamentals of Interactive Computer Graphics, Reading, Massachusetts: Addison-Wesley Publishing Co., 1984.

Pavlidis, T., Algorithms for Graphics and Image Processing, Rockville, Maryland: Computer Science Press, 1982.

Zyda, M. J., Algorithm Directed Architecture for Real-Time Surface Display Generation, Ph.D. Dissertation, Sever Institute of Washington University, Saint Louis, Missouri, 1984.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center   2
    Cameron Station
    Alexandria, Virginia 22304-6145

2. Superintendent   2
    Attn: Library (Code 0142)
    Naval Postgraduate School
    Monterey, California 93943-5002

3. Chairman (Code 52)   1
    Department of Computer Science
    Naval Postgraduate School
    Monterey, California 93943-5004

4. Computer Technology Programs (Code 37)   1
    Naval Postgraduate School
    Monterey, California 93943-5004

5. Michael J. Zyda (Code 52Zk)   2
    Department of Computer Science
    Naval Postgraduate School
    Monterey, California 93943-5004

6. Thomas Wu (Code 52Wq)   2
    Department of Computer Science
    Naval Postgraduate School
    Monterey, California 93943-5004

7. Dr. Henry Fuchs   1
    208 New West Hall (035A)
    University of North Carolina
    Chapel Hill, North Carolina 27514

8. Lt. Cdr. Patrick G. Hogan, USN   10
    102 Borden Avenue
    Wilmington, North Carolina 28403

61

END

FILMED

4-86

DTIC