NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SURFACE RECONSTRUCTION FROM PLANAR CONTOURS

by

Allan R. Jones

June 1986

Thesis Advisor: Michael J. Zyda

IRITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

REPORT SECURITY CLASSIFICATION ICLASSIFIED	1b. RESTRICTIVE MARKINGS
SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
DECLASSIFICATION/DOWNGRADING SCHEDULE	
ERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
NAME OF PERFORMING ORGANIZATION aval Postgraduate School — 6b. OFFICE SYMBOL *(If applicable)* 52	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
ADDRESS *(City, State, and ZIP Code)* onterey, CA 93943-5000	7b. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000
NAME OF FUNDING/SPONSORING ORGANIZATION — 8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
ADDRESS *(City, State, and ZIP Code)*	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. | PROJECT NO. | TASK NO. | WORK UNIT ACCESSION NO.

TITLE *(Include Security Classification)* UNCLASSIFIED
Surface Reconstruction From Planar Contours

PERSONAL AUTHOR(S)
Allan R. Jones

TYPE OF REPORT	13b. TIME COVERED	14 DATE OF REPORT *(Year, Month, Day)*	15 PAGE COUNT
aster's Thesis	FROM ______ TO ______	20 June 86	58

SUPPLEMENTARY NOTATION

COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUB-GROUP	Surface Construction, Surface Triangulation, Planar Contours

ABSTRACT *(Continue on reverse if necessary and identify by block number)*

any applications of computer graphics involve the representation of a
ree dimensional solid reconstructed from a sequence of two-dimensional
lanar contours. Surface construction algorithms accomplish this by
apping individual pairs of contours, forming triangular surface patches,
nich approximate the original three-dimensional solid. In this paper,
e present an expanded algorithm which not only handles the mappings of
ultiple contours per plane and partial contour mappings, but also
lows human interaction to resolve mapping problems. We include a
iscussion of our algorithm's limitations and proposed solutions.

DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
. NAME OF RESPONSIBLE INDIVIDUAL rof. Michael Zyda,	22b. TELEPHONE *(Include Area Code)* 408 646-2305 — 22c. OFFICE SYMBOL 527k

FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Approved for public release, distribution unlimited

Surface Reconstruction From Planar Contours

by

Allan R. Jones
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1974

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL

June 1986

ABSTRACT

Many applications of computer graphics involve the representation of a three-dimensional solid reconstructed from a sequence of two-dimensional planar contours. Surface construction algorithms accomplish this by mapping individual pairs of contours, forming triangular surface patches, which approximate the original three-dimenisonal solid. In this paper, we present an expanded algorithm which not only handles the mappings of multiple contours per plane and partial contour mappings, but also allows human interaction to resolve mapping problems. We include a discussion of our algorithm's limitations and proposed solutions.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I wish to express my thanks to the people who supported my efforts in this work. To my advisor, **Dr. Michael Zyda**, who gave me the opportunity, guidance, and patience necessary to complete this thesis project. But most of all I would like to thank my wife, ***Kathy,*** who has been most supportive and understanding in all my endeavours while at NPS and who gives me the reason for carrying on.

I. INTRODUCTION

Many applications of computer graphics involve the representation of a three-dimensional solid reconstructed from a sequence of two-dimensional planar contours. These contours are obtained by some electronic sensor that records data from the original three-dimensional object along a finite number of parallel planes. The intersection between these two-dimensional parallel planes and the three-dimensional object forms these contours which lie along the solid's exterior and interior surfaces. These contours appear as line segments on the parallel planes and are represented as either closed loops, open segments, or single points. The main purpose of surface construction algorithms is the formation of surface patches between these contours on adjacent planes in order to approximate the original three-dimensional solid.

The problem of surface construction from two-dimensional parallel planes is characterized by mapping and triangulating pairs of planar contours into surface patches that form a display. The surface construction algorithm identifies the appropiate contours, including the specific portions of those contours, that should be mapped. Then connections are formed by building triangular tiles between individual line segments from one contour and a single point from the end of a line segment on the other mapped contour. This tiling operation is executed for all the line segments in the identified contours.

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. [FUCHS,1977]

This specification is mutually described in all the public literature on surface construction [GANAPATHY,1982] [CHRISTIANSEN,1978] [SHANTZ,1981] [HOGAN,1985] [FUCHS,1977]. Each of the papers uses the notation to expand upon the initial algorithm originally proposed by Fuchs.

The initial action of this paper is a brief review of all previous surface construction algorithms, concentrating mainly on their capabilities and limitations. The main part focuses on the algorithm presented by Hogan. This algorithm is more comprehensive in that it can handle multiple contours per plane and partial contour mappings. Nonetheless, it also does not provide a complete solution to the surface construction problem. Following the discussion of the Hogan algorithm we present a further expanded algorithm which attempts to resolve each of that algorithm's limitations.

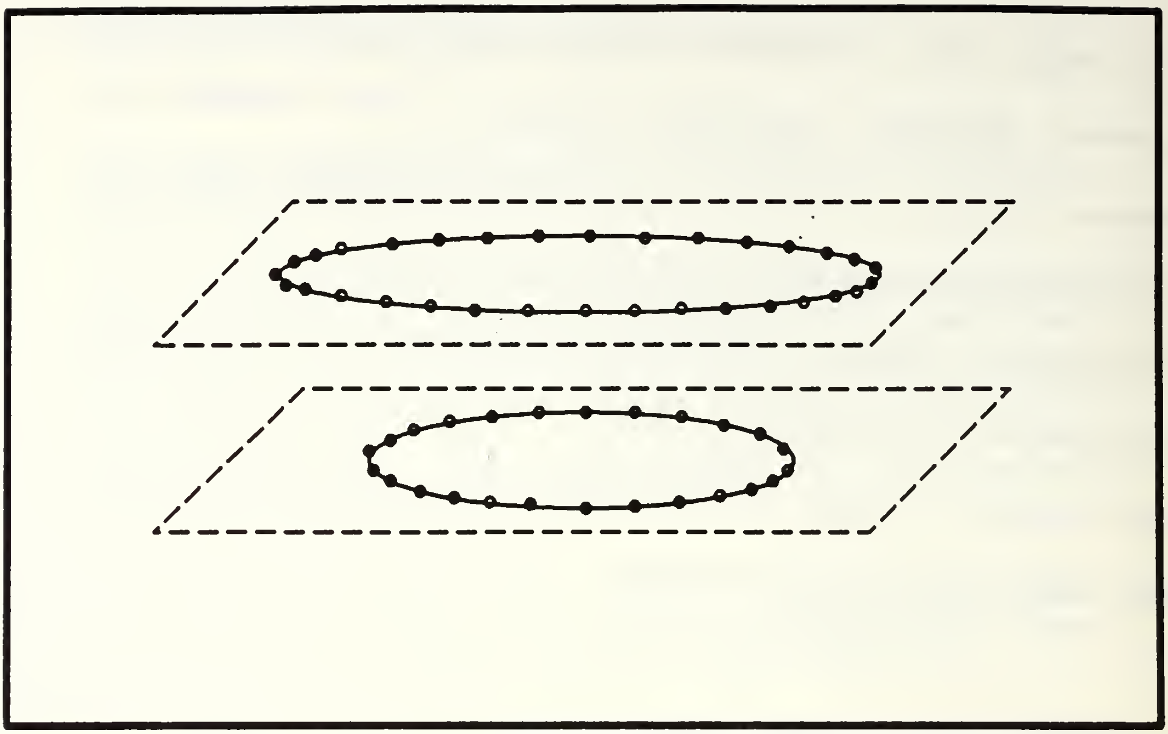


Fig 1.1 - Two contours on adjacent, parallel planes.

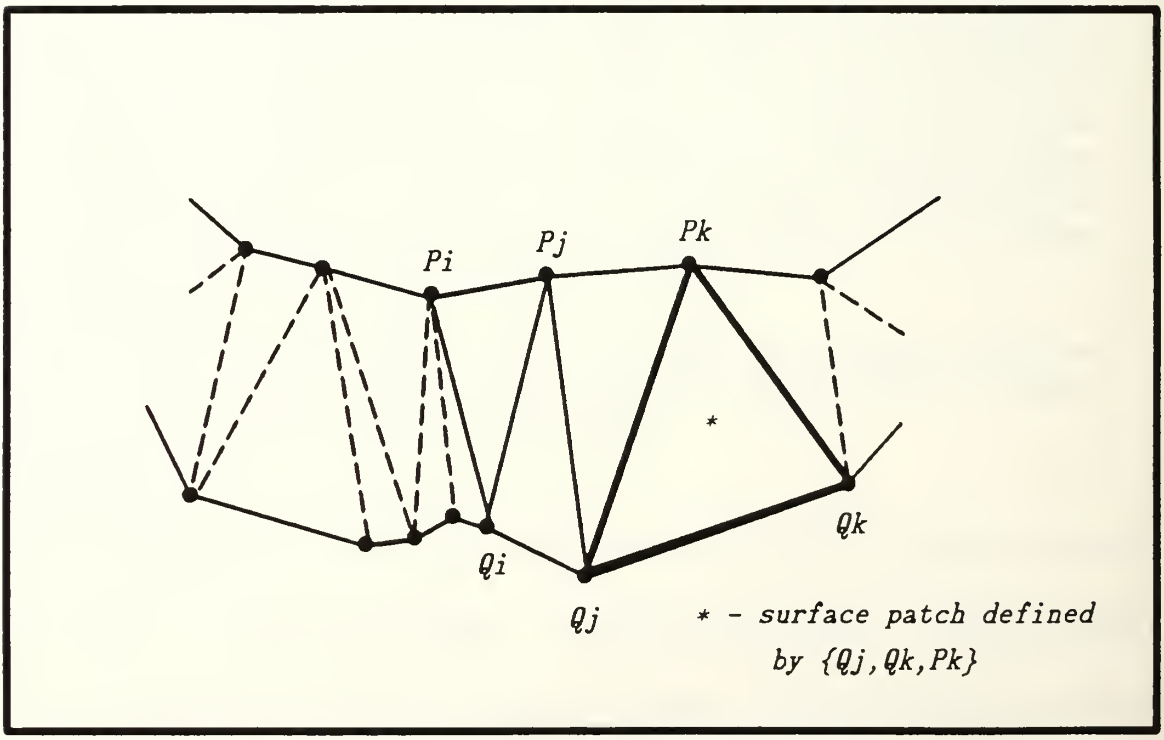


Fig 1.2 - Mapped connections into triangulated surface patches.

II. LITERATURE REVIEW

The method for finding an approximation by triangulation of a surface defined by a set of contour lines has been the subject of past articles written by [KEPPEL,1975], [FUCHS,1977], [CHRISTIANSEN,1978], [SHANTZ,1981], and [GANAPATHY,1982]. Each author has addressed different aspects of the problem. However to date, no reliable algorithm has been published which can successfully handle triangulating complex surfaces in all cases. The reason for this is that insufficient information is obtained from the contour lines regarding the gradients associated with the surface they describe [KEPPEL,1975]. Contour lines of an irregular surface, such as found in nature, do not lend themselves to curve fitting, or other attempts at precise mathematical descriptions [CHRISTIANSEN,1978].

Our surface construction algorithm is based on the efforts of Fuchs, Christiansen, and Hogan. In order to fully understand the underlining problem of surface construction, a brief summary of all previous literature is presented. This summary focuses mainly on each of the algorithm's capabilities and limitations.

Fuchs algorithm for surface construction was presented in [FUCHS,1977]. His problem statement, stated in our introduction, is the basis of all subsequent literature. The main contributions of that paper are the concise statement of the surface construction problem and a method for connecting simple, closed contours (Figure2.1 and [HOGAN,1985]).

Fuchs' algorithm contains three major limitations in dealing with complex surfaces. The first limitation is that his algorithm can only handle cases of simple, closed contours, with only one contour on each of the mapped planes. It cannot handle the more complex case of multiple contours on adajcent planes, partial contour mappings, or open (non-closed) contours (Figure2.2 and [HOGAN,1985]). The problem with multiple contours on adajcent planes, arises from the fact that Fuchs' algorithm does not provide the mechanics necessary to identify which of the contours should be mapped. The more general case for surface construction is to have multiple contours on each plane. The second limitation of Fuch's algorithm is that it performs a complete contour-to-contour triangulation between adajcent contours, even in cases where a partial mapping is more appropriate. Partial triangulation of contours is most often representative of situations in which we have dissimiliarly sized contours. The third limitation in Fuch's algorithm is in its inability to handle open contours. This is the direct result of his algorithm's lack of generality. A method designed to handle the partial contour mappings is also capable of handling open contours.

In [CHRISTIANSEN,1978] we see an algorithm that is similiar to Fuch's. The major difference is a mechanism which allows human interaction to resolve mapping ambiguities. This mechanism allows the user to determine the relative connection points in the mapping process for highly convoluted contour cases [HOGAN,1985]. This procedure can be quite time consuming, depending on

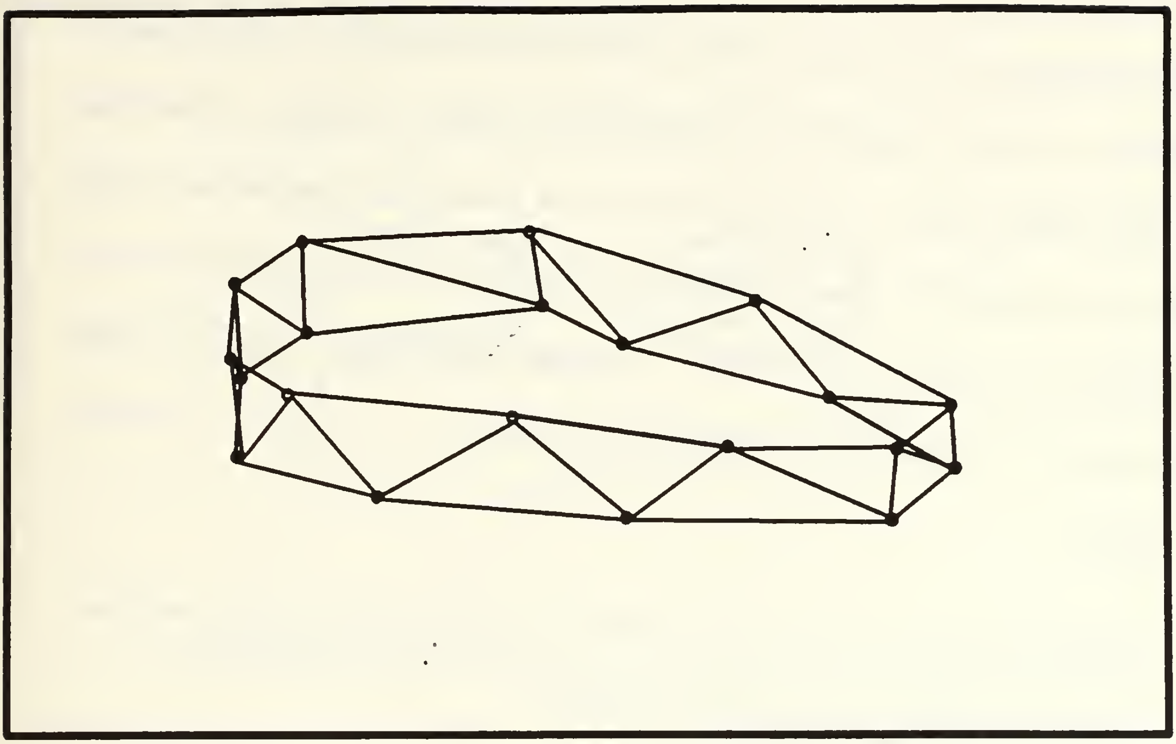


Fig. 2.1 - Triangulated pair of simple, closed contours.

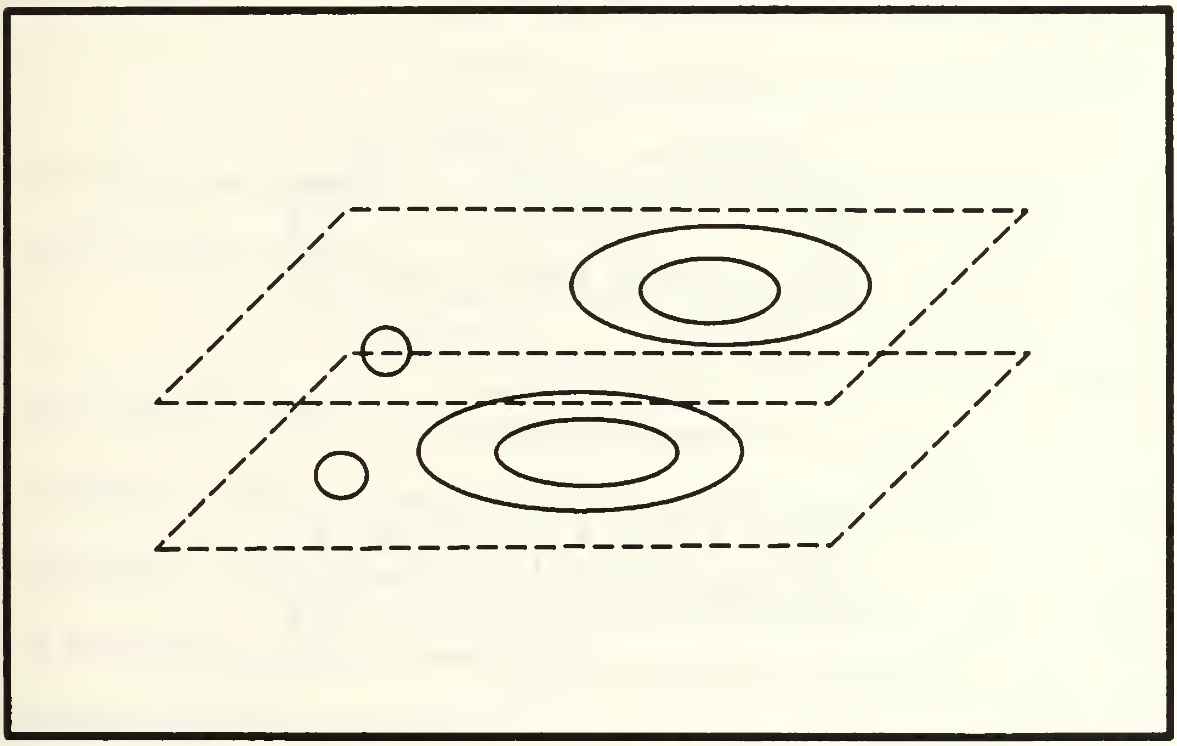


Fig. 2.2 - Example of multiple contours per plane.

the complexity of the data base [CHRISTIANSEN,1978]. Christiansen's algorithm has the capability of handling some simple branching. Branching normally results from a pair of contours in one plane being mapped to a single contour on an adjacent plane, (Figure 2.3). This branching capability allows the algorithm to handle simple cases of multiple contours on adjacent planes. Christiansen accomplishes the branching capability by utilizing the following procedure.

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. (Figure2.4)

3. Triangulate as usual. [CHRISTIANSEN,1978]

In general this algorithm introduces a new node between the two, planar contours. This new node is used to form single, connected regions which are then processed by the original surface construction algorithm.

The problems with Christiansen's algorithm are its inability to handle open contours and its inability to handle complex cases of multiple contours on adjacent planes, except through the use of expensive human interaction. Christiansen interestingly shuns the optimality seen by Fuchs as important by utilizing the heuristic of choosing the "shortest diagonal" in forming triangular tiles instead of minimal triangular area. As stated in his article, this heuristic is

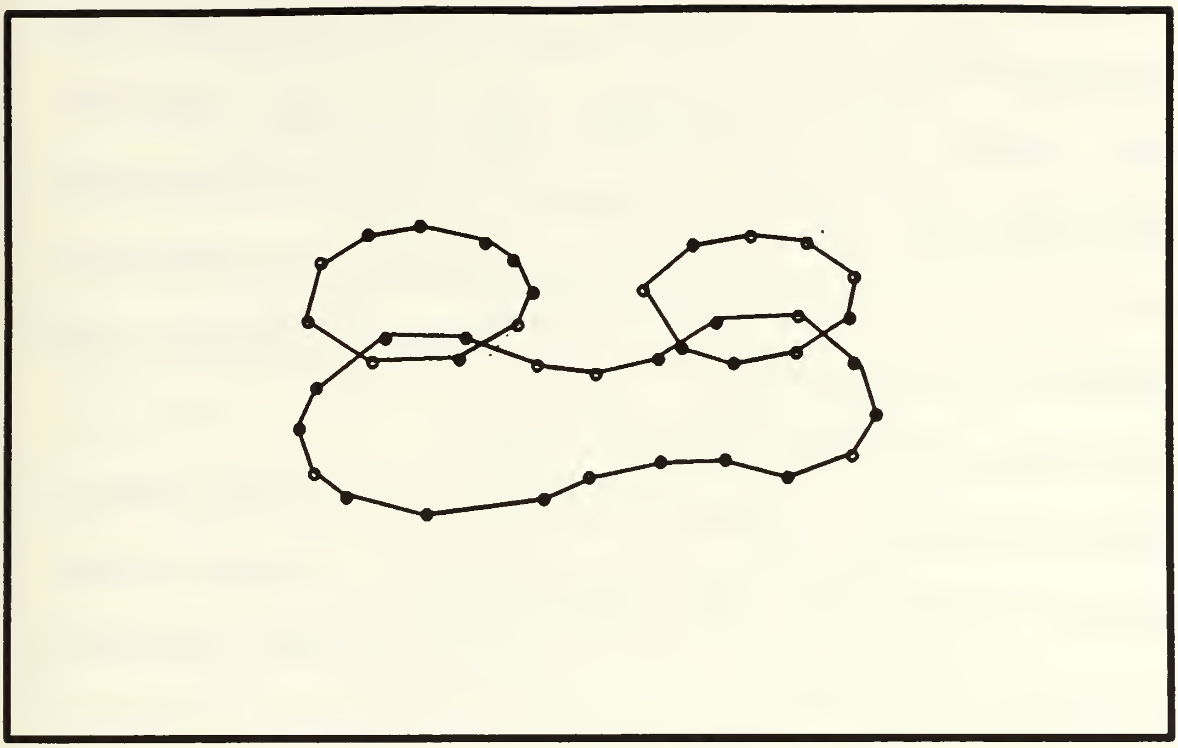


Fig. 2.3 - Simple case of branching.

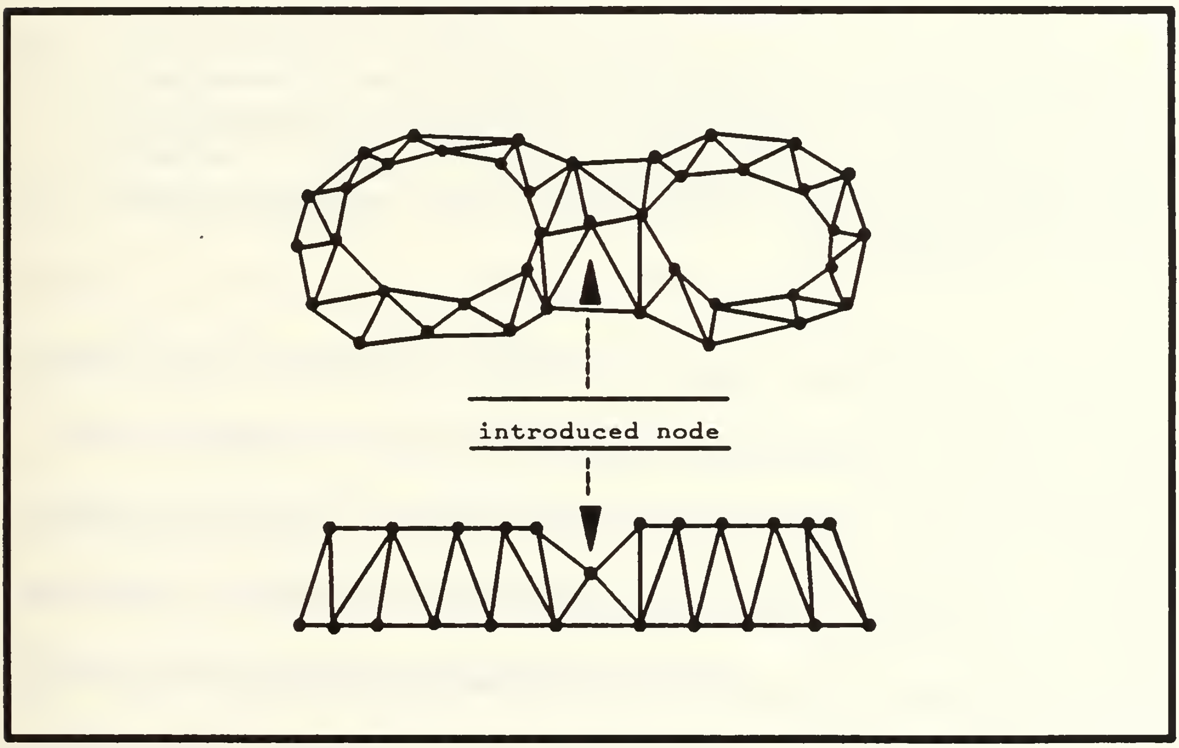


Fig. 2.4 - Triangulation scheme for branching.

easily implemented, fast, and works well as long as the two contours/loops are mutually centered and are reasonably similar in shape and size [CHRISTIANSEN,1978]. The process considers the next two nodes of each contour as candidates for triangulation. After determining the lengths of all possible diagonals for the surface patch, nodal selection for triangulation results from the surface patch exhibiting the shortest diagonal.

The algorithm proposed in [SHANTZ,1981] is basically an extension of Fuch's and Christiansen's algorithms. This extension includes the capability to handle contour defined objects which are highly branched and have holes. Handling of multiple contours on adjacent planes is achieved by:

> For branching contours where n contours in section i are connected to m contours in section i+1, the surfaces are mapped by first concatenating the section i contours into a single large contour using a minimum number of minimum distance links, similarly concatenating the section i+1 contours, then performing the one to one mapping between the resulting composite contours [SHANTZ,1981].

Once the concatenation process is completed, Shantz uses Fuch's closed contour mechanism to formulate the connections between the composite contours. After the connections have been formed, any extraneous connections resulting from the concatenation process are removed. The resolution of ambiguities arising from multiple contour cases requires human interaction and similar to the Christiansen algorithm, Shantz states that this is extremely labor intensive.

Shantz cites a specific case in which a set of contours from the Livingston brain database required many hours of contour splitting with an interactive cursor.

The main limitations of the Shantz algorithm are its inability to handle cases of open contours and partial contour mappings. Additionally it can only handle cases of multiple contours on adjacent planes when a composite contour can be formed. or the ambiguities can be resolved by human interaction.

The algorithm described by Ganapathy [GANAPATHY,1982] is a further improvement on the Fuchs' and Christiansen's methods of handling simple, closed contours. This improvement results from using a more computationally expedient heuristic for triangulations [HOGAN,1985]. However, Ganapathy's algorithm does not include the capabilities introduced and discussed by Shantz. Instead he simply assumes a complete mapping of paired contours. which is not always the case.

The problem with the Ganapathy algorithm is that it represents a general solution for handling only simple cases of surface construction. Capabilities for handling multiple contour mappings, partial contour mappings, or human interaction are not provided and their issues are not addressed in his presentation.

The algorithm presented by Hogan [HOGAN,1985] is more complete than its predecessors in that it not only handles the simple cases of contour mapping. but additionally provides a more comprehensive procedure for resolving the multiple contours per plane and partial mapping problems. The only capabilities lacking from the Hogan algorithm are the one for handling branching as described in the

Christiansen paper and the one for human interaction for the resolution of highly ambiguous mappings.

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 handling multiple contours per plane, partial contour mappings, and which supports simple cases of branching. In addition the algorithm should provide a mechanism for human interaction for the resolution of highly ambiguous mappings.

The surface construction algorithm we present handles the cases of simple contour mappings, multiple contours per plane, partial mappings, and in addition provides a mechanism for human interaction to deal with cases involving highly ambiguous mappings. The only capability lacking from our algorithm is that for handling branching as described in the Christiansen paper. A discussion of our algorithm follows, with a proposed solution for handling cases involving branching.

III. SURFACE CONSTRUCTION ALGORITHM

In the preceding section, we presented a discussion of previous surface construction algorithms. Here, we present a detailed discussion of our algorithm by first specifying the known input/output data structures.

Surface construction of an object between a set of planar contours (Figure 3.1) can be reduced to constructing the surface triangulations between two adjacent planes. The specification of the problem can be best seen by listing the known input data structures [HOGAN,1985]:

* 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 [HOGAN,1985]:

* 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 [HOGAN,1985]:

* 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.

Our surface construction algorithm is composed of the following outlined steps:[1]

A. 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 these 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 is used to create the data structures required for surface construction.

[1] The bulk of this discussion is drawn from [ZYDA,1984] and [HOGAN,1985]

B. 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 would result in an erroneous approximation 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.

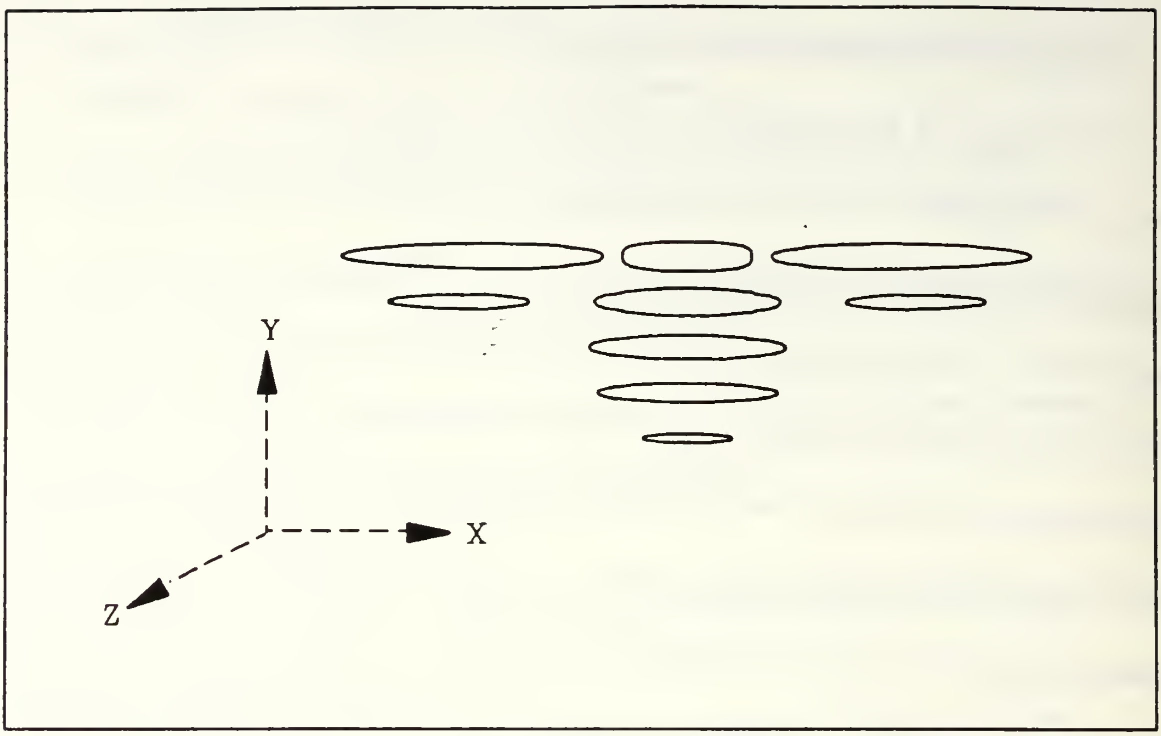


Fig. 3.1 - A partial set of planar contours from a 3D Z^2-orbital of a hydrogen molecule.

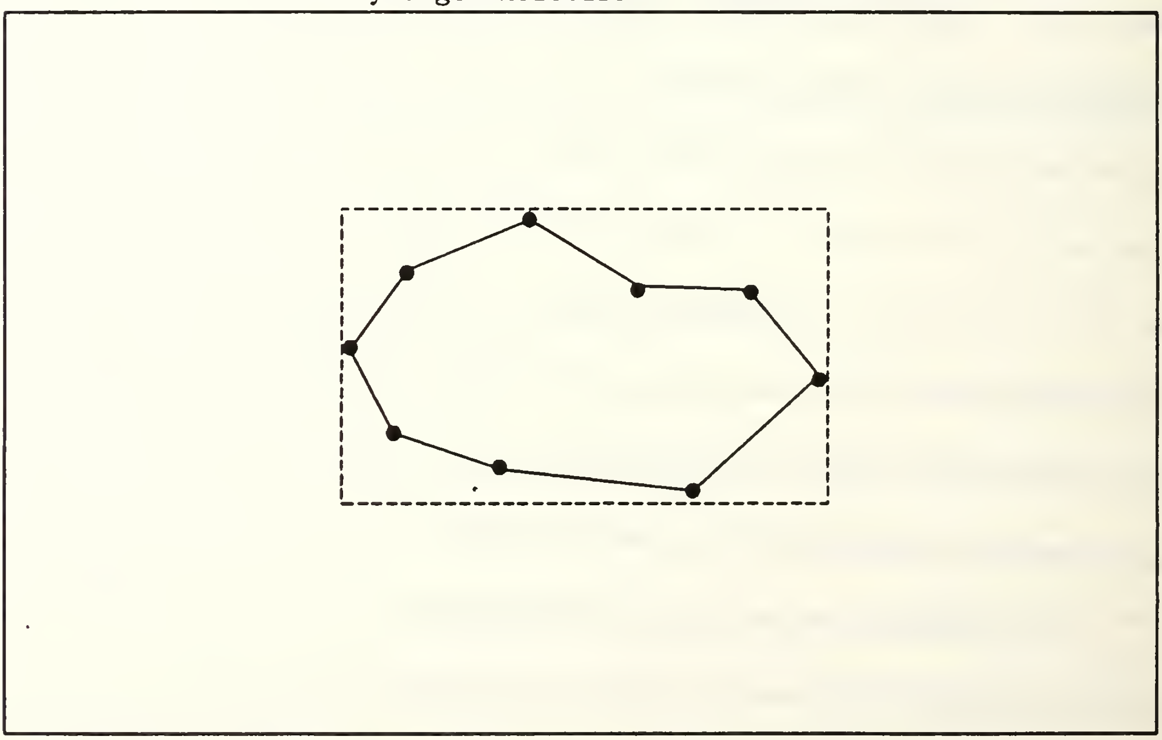


Fig. 3.2 - Two dimensional bounding box used for determining overlap percentage value.

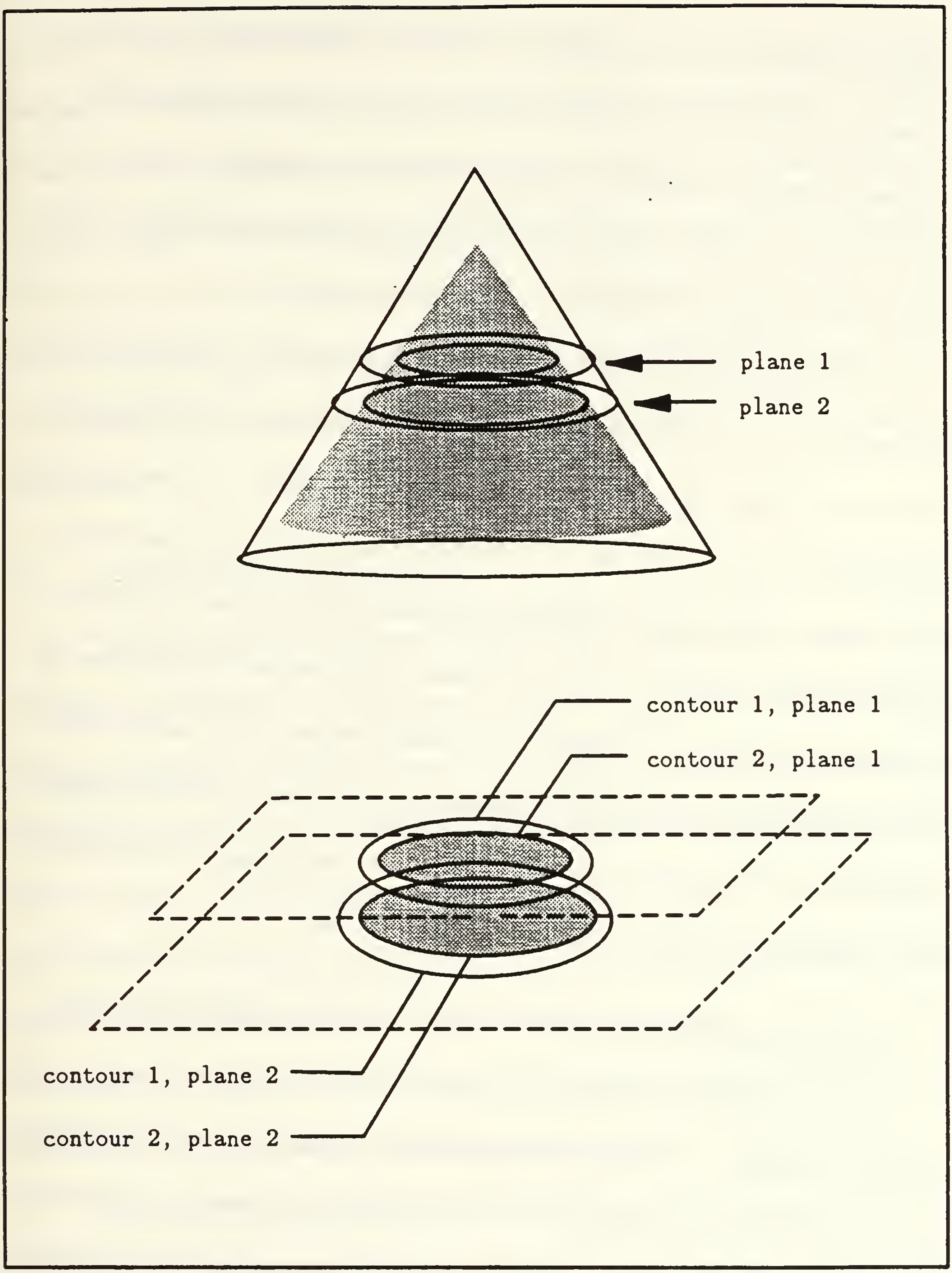


Fig. 3.3 - Example of consistency check using item interior specifications with overlap percentage values.

However, since contour 2 on plane 2 is low valued with respect to its solid interior and contour 1 on plane 1 is high valued, this mapping can be eliminated.

These values 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).

C. FORM COORDINATE MAPPING FOR MAPPED CONTOUR PAIRS

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.

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

E. 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. Also, this cancellation precludes connecting contour points which have already been selected for connection.

F. 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.

G. EDIT CONTOURS

We extract the contour coordinates from the input data file and use them to create contour defined objects. These contour defined objects generated are then available for the user to remove or save for reprocessing by the surface construction algorithm as necessary.

H. RELAX HEURISTICS

In this procedure, we allow the user to input his own values for the three heuristic values (overlap percentage minimum, boundary tolerance percentage, tolerance multiplier) utilized by our surface construction algorithm. The user has the option of changing one or all three. Once these values have been entered, the information is used in the connection process of our algorithm to produce more a correct mapping between the planar contours.

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 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.

As noted, the contour interior specification is the only piece of data which is difficult to obtain. It 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 is provided to allow for user specification of contour interior values. The range of interior values is HIGH, LOW or INDETERMINATE. The problem that occurs without this value concerns the contour pairing problem encountered in multiple contour situations where contours are closely spaced and of similar shape. Here, 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 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.

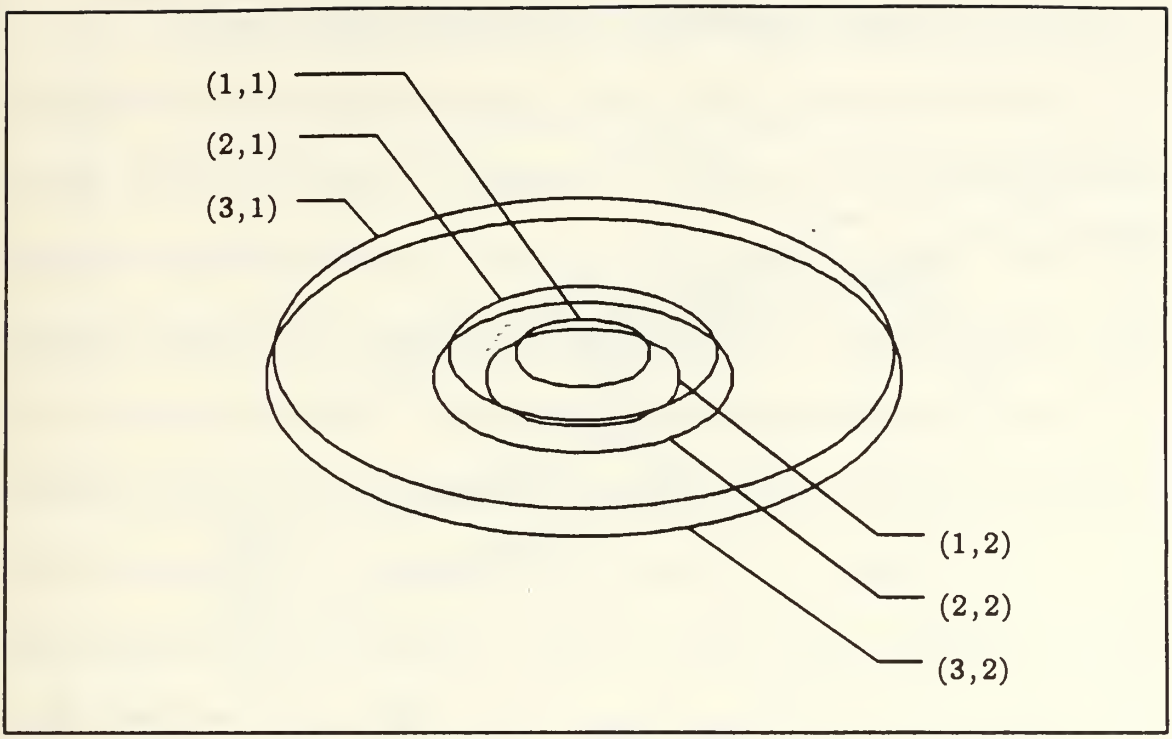


Fig. 3.4 - Example of multiple contours per plane on adjacent planes.

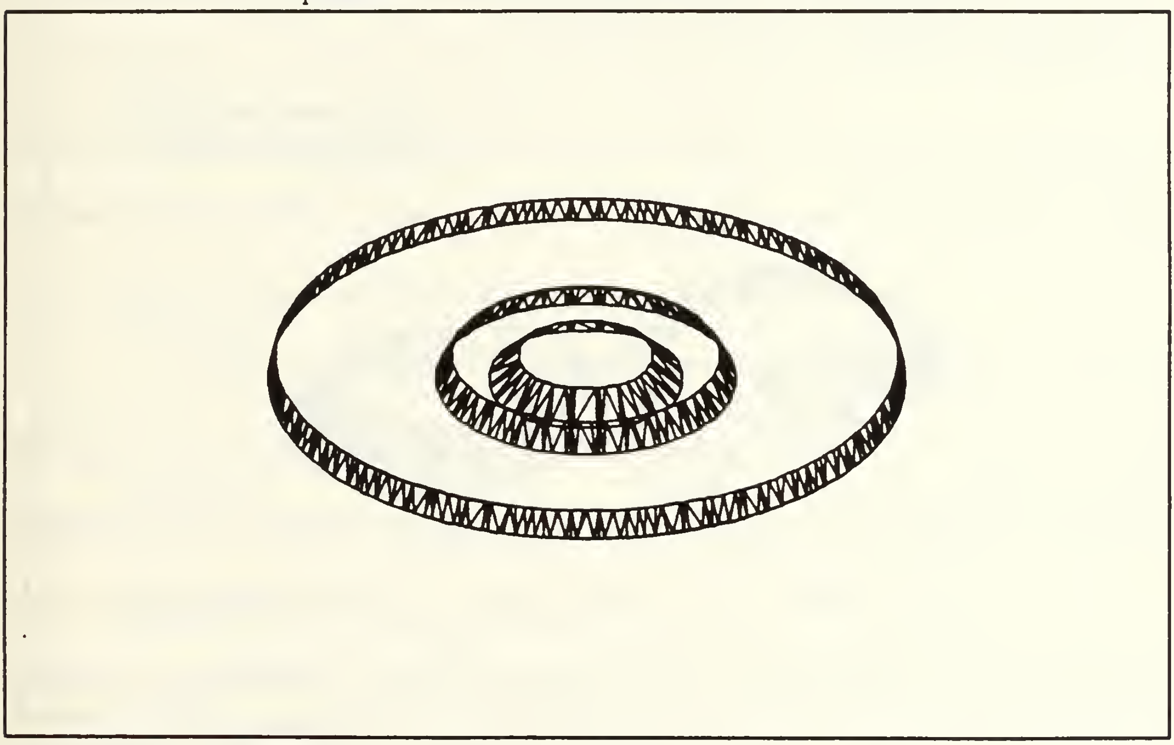


Fig. 3.5 - Connection of Figure 3.4.

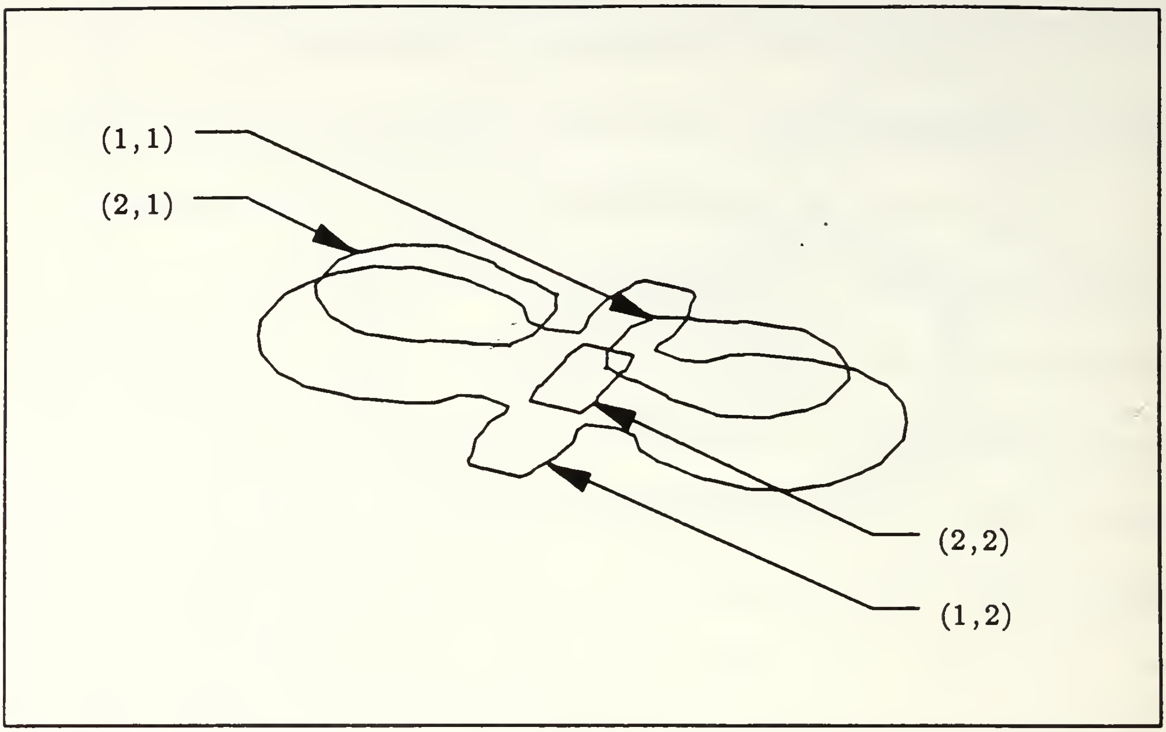


Fig. 3.6 - Example of a set of contours requiring partial mappings and an exterior to exterior mapping; (1,1) and (2,1) to (2,2).

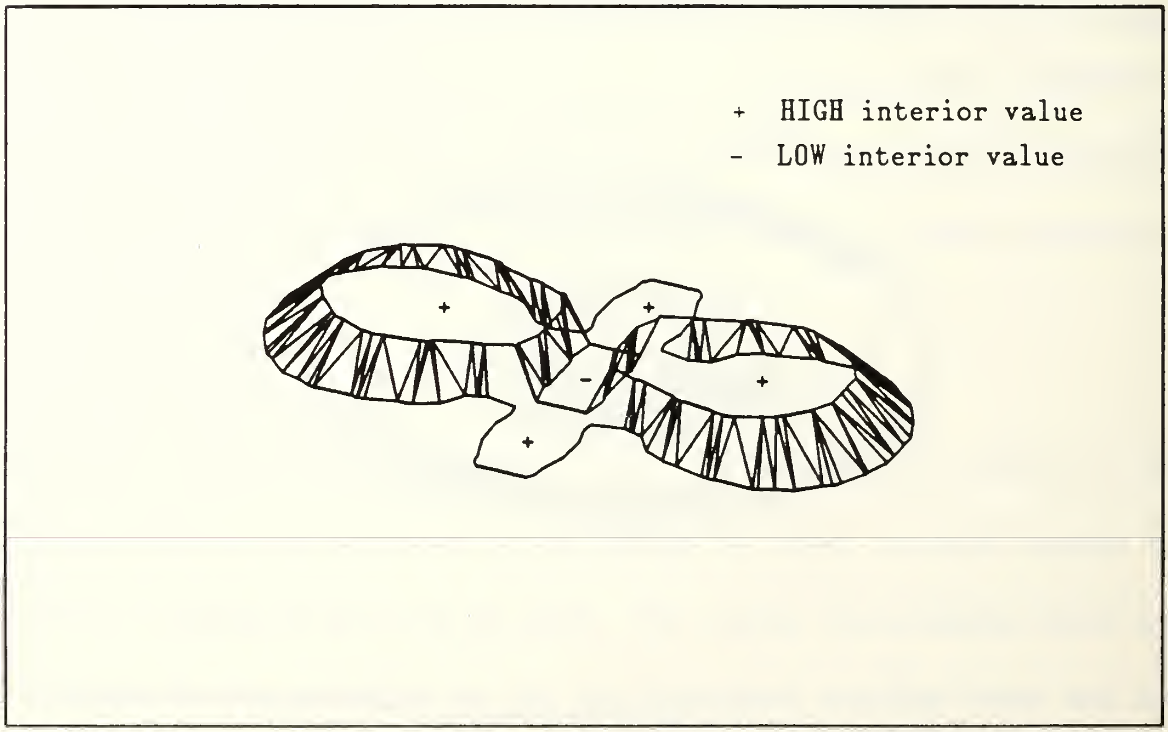


Fig. 3.7 - Connection of Figure 3.6, with contour interior values for each contour.

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.

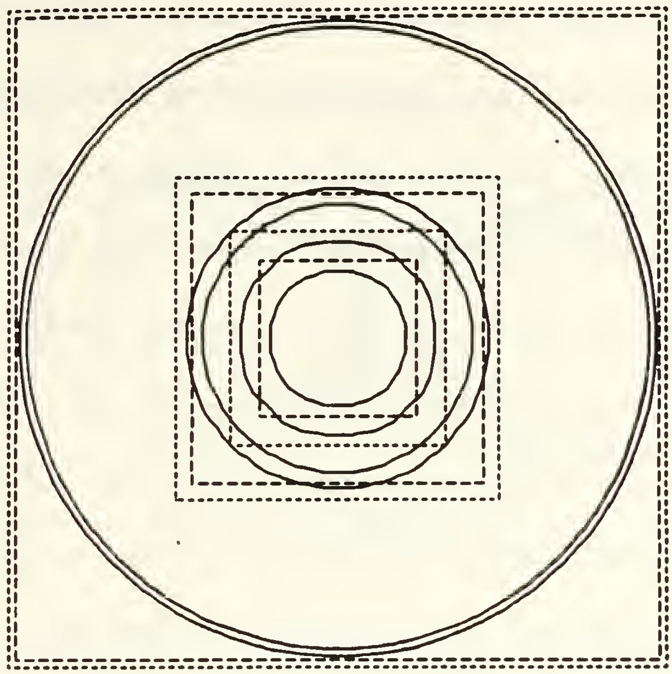
After the overlap percentage has been computed for a contour pairing, it is used in conjunction with the interior specifications to determine the mapping type for the contour pair. An interior to interior mapping is indicated when a high percentage of overlap (greater than 10%) exists for a pair of contours. 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).

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

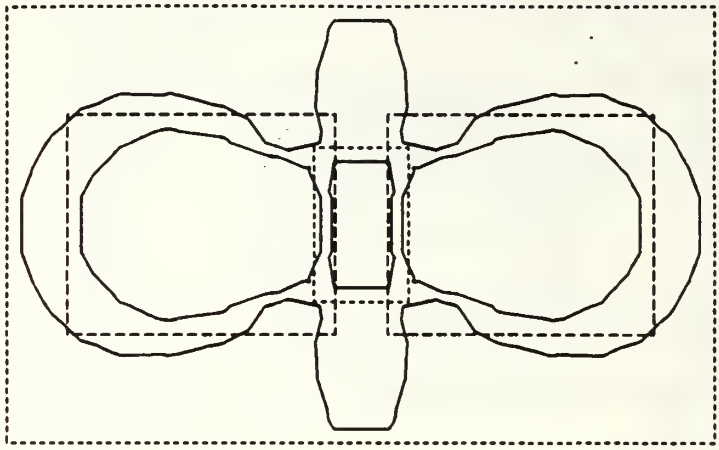


OVERLAP TABLE

Plane 2

Plane 1		CONTOUR 1	CONTOUR 2	CONTOUR 3
	CONTOUR 1	95.6916	0.0	11.1493
	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
Plane 1	CONTOUR 1	19.0295	5.4386
	CONTOUR 2	19.0295	5.4386

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

considered for mapping. Mapping paired contours is on a largest to smallest overlap percentage criteria. 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 as interior with the smaller area. 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. 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

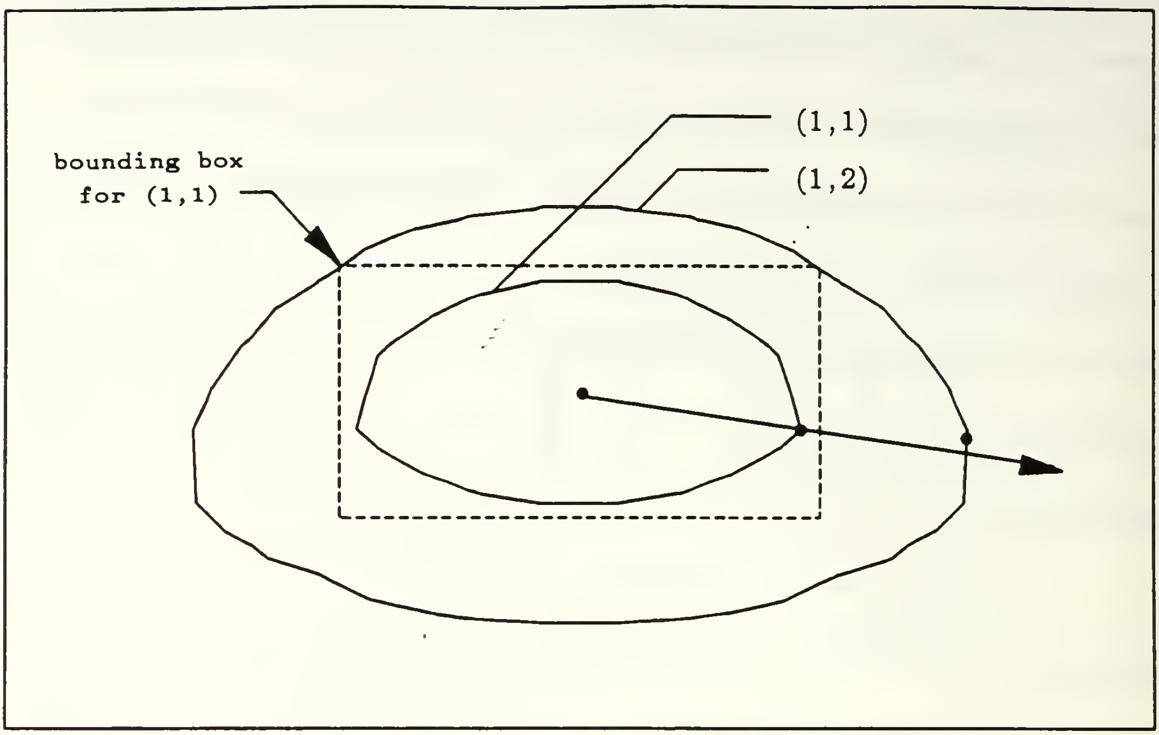
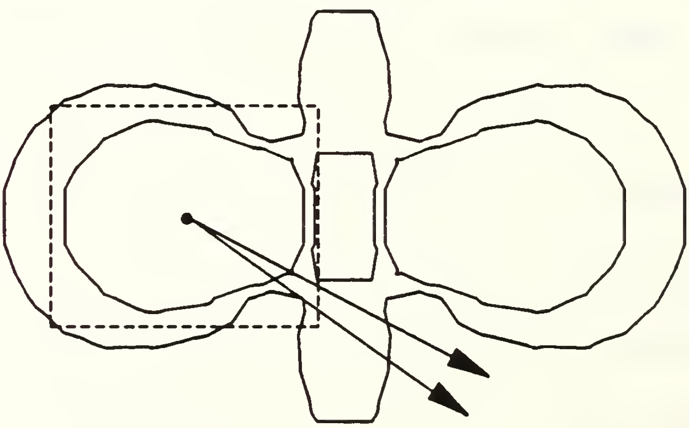


Fig. 3.10 - Vector radiating from center coordinate through the interior coordinate towards the outer contour for tentative mapping



inner coord	outer coord	distance
.	.	.
.	.	.
.	.	.
24	52	0.2001
25	53	0.1769
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.

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 noted that all of the inner coordinates may not necessarily 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 will be 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 and the last coordinate considered was 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 well outside of tolerance with the last coordinate added and 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 maping, 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 (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 merging 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. It is shown that the 5 initial continuity sets have been merged into 3 sets of non-overlapping distance range.

Total Initial Sets = 5

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

Total Merged Sets = 3

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

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

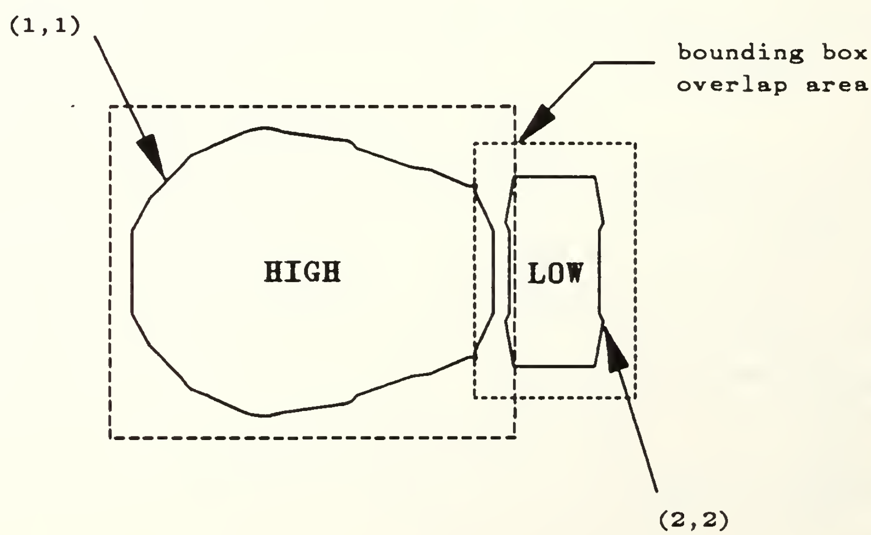


Fig. 3.13 - Bounding box overlap for exterior to exterior mapping. Only the coordinates within the overlap area are mapped.

After we have merged continuity sets, we need to determine which of those sets of coordinates 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.

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 [CHRISTIANSEN,1978]. 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 where 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.

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 will be 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 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.

5. Edit Contours

The purpose of the edit contour process is to allow user interaction in identifying the planar contours that pose a problem for our surface construction algorithm. The contour coordinates are obtained from the input data file and used to create triangulated surfaces. Once the triangulated surfaces are generated, we can utilize the picking mechanism of the IRIS-2400 graphics system for editing.

With this process, the user can remove contours that produce valid connections. The user can then concentrate his efforts on the contours that produce invalid results. After the problem contours are identified and selected by the user, they are saved to a file for later reprocessing by our algorithm. The user can then recall the file containing the problem contours and in conjunction with the "relax heuristics" procedure possibily force a valid connection by adjusting the heuristic values used by our surface construction algorithm.

6. Relax Heuristics

The purpose of the relax heuristics procedure is to allow the user the option to adjust the three heuristic values used by our surface construction algorithm. By adjusting these values, connections between contour pairs that might otherwise be disregarded can be possibly coerced.

The first heuristic value is the overlap percentage minimum. Step two of our algorithm determines the percentage of overlap between contours on adjacent planes. These percentages are used as a consistency check for matching interior specifications. We apply our overlap percentage heuristic in the final phase of this pairing procedure. Contour pairs having an overlap percentage minimum, with matching interior specifications, are mapped interior to interior. Contour pairs having non-zero percentages below the minimum, with non-matching interior specifications, are mapped exterior to exterior. All other contour pairs are disregarded.

The value that is preset in our algorithm for the overlap percentage minimum is ten percent. This value, through experimentation, results in the greatest number of correct contour pairings. However, some contour pairs which should be mapped are disregarded because of this selection for the overlap percentage minimum. Figure 3.14 is an example of such a situation. In that figure, we have a pair of contours with matching interior specifications (HIGH:HIGH), and in addition having an overlap percentage of less than ten percent. By our preset overlap percentage minimum value, this contour pair is not considered for mapping and remains unconnected. But by allowing the user to adjust the overlap percentage minimum for an occurrence such as seen in Figure 3.14, an appropriate connection can be generated.

The second heuristic value is the boundary tolerance percentage. In the initial two steps of our algorthm we determine the contour item two-dimensional bounding box values and then use them for overlap determination. Instead of creating the bounding box from the minimum and maximum X and Y coordinates, we adjust the bounding box values by a percentage in order to promote mappings. If only the minimum and maximum X and Y coordinates were used to describe bounding boxes, situations such as seen in Figure 3.15 would go unconnected. In that case, we see a bounding box created from the minimum and maximum X and Y coordinates. This results in a zero percentage overlap and no connections are generated. This is an unsatisfactory situation since the contours should be mapped. By allowing the user to adjust this heuristic value, opportunities are now available for user intervention to handle mapping situations that would otherwise be neglected by our algorithm.

The last heuristic value is the tolerance multiplier. When handling an interior to interior mapping, our algorthm utilizes a tolerance factor for the

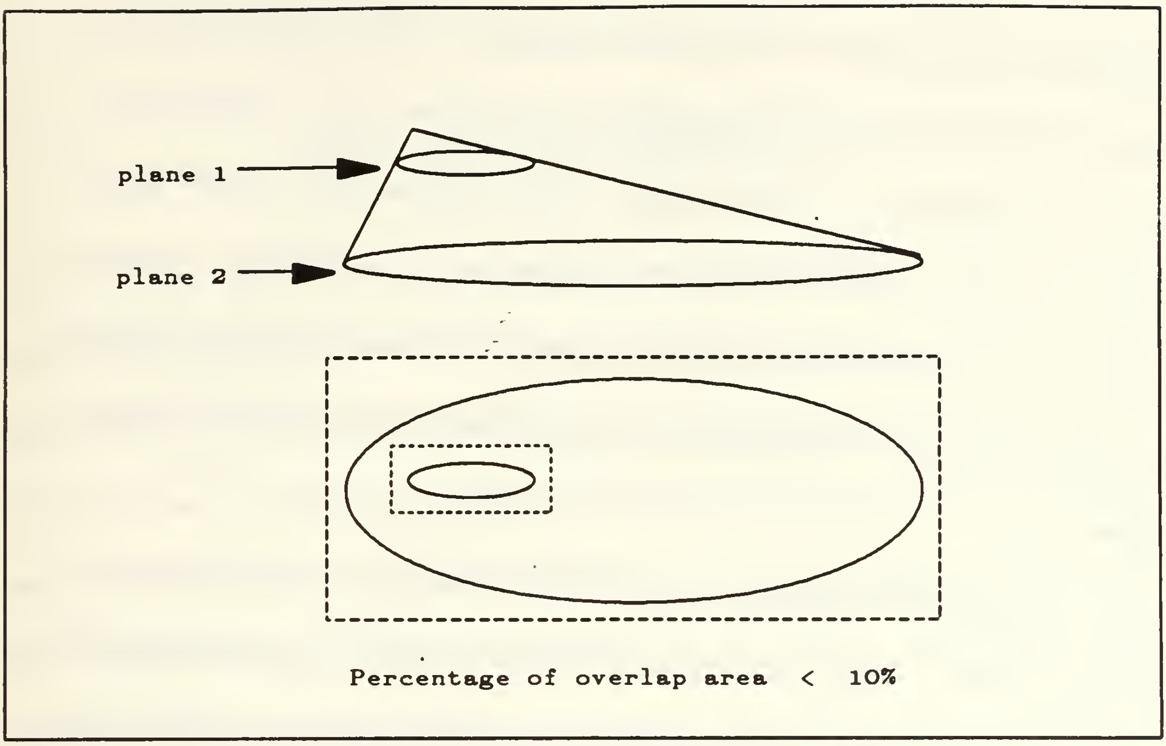


Fig. 3.14 - Example of a contour pair which should be mapped, but would be disregarded due to overlap percentage below the minimum.

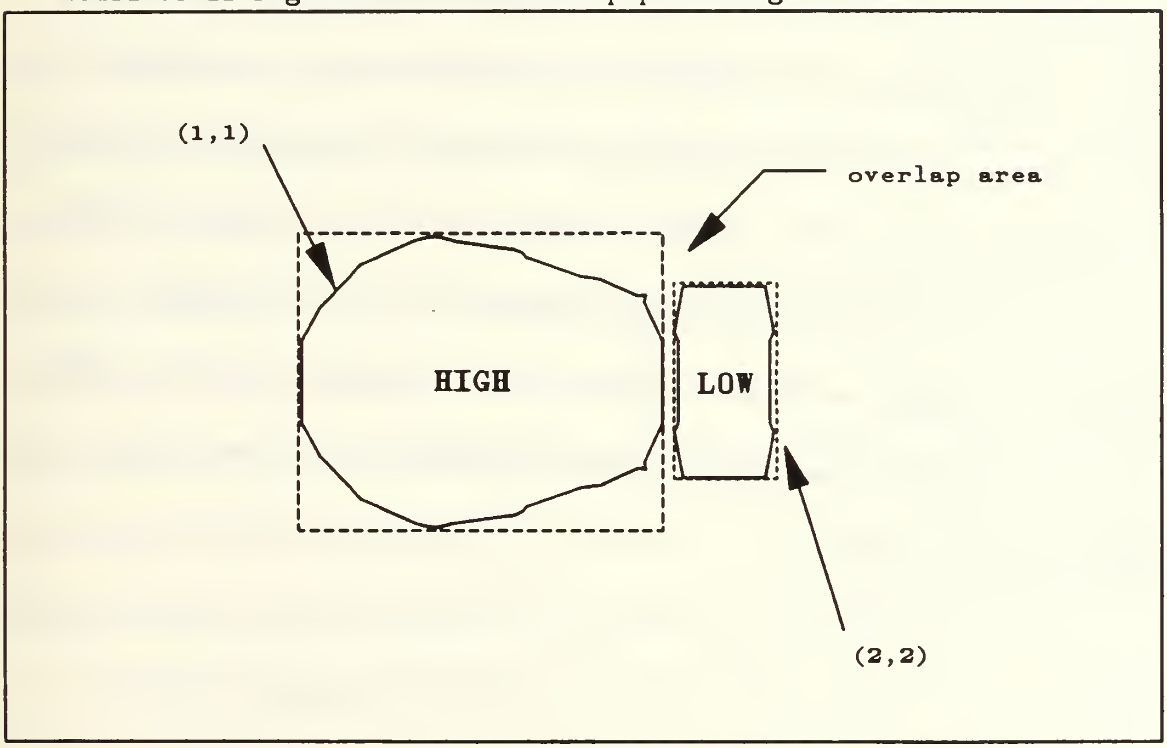


Fig. 3.15 - Example of contours' 2D bounding boxes created strictly from the min and max X and Y coordinates. Resulting overlap = 0.

determination of the initial continuity set assignments. This tolerance factor is based on 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. This window value is a constant value used for the selection of appropriate mapping connections. Again by the allowing user to adjust this heuristic value, we provide opportunities to handle mapping cases that might otherwise not be included by our preset value.

We have presented a thorough discussion of our algorithm for surface construction. Particular attention has been devoted to the strengths of our algorithm, specifically its capabilities for handling multiple contours per plane, partial contour mappings, editing contours, and relaxing the heuristic values. This algorithm has proved to outperform all previous algorithms in surface construction via the triangulation of contours. In addition, with the incorporation of the edit contour procedure and the heuristic relaxation procedure, our algorithm can solve mapping situations that would otherwise be neglected. Although we have provided more capabilities for our surface construction algorithm, we still have some limitations. In the next chapter, we address those limitations.

IV. ALGORITHM LIMITATIONS

In the previous chapter, we discussed the capabilities of our surface construction algorithm, emphasizing its handling of multiple contours per plane and partial contour mappings. Additionally, we described its newest feature of providing user intervention in editing contours and adjusting the three heuristic values utilized by our algorithm. However, there still exist contour mapping situations which our surface construction algorithm can not handle. These situations together with suggested solutions are described below.

The first mapping situation involves simple branching of one contour on one plane to two or more contours on an adjacent plane (Figure 2.3). When presented with this case, our algorithm produces an incomplete contour mapping because of missing data. Our suggested solution to this problem is based on a concept described in the Christiansen paper [CHRISTIANSEN,1978]. A procedure could be created to introduce a new node between the closest nodes of the branches. The Z coordinate of the new node would be the average of the Z coordinates of the two contour levels involved. Once the new node is in place, triangulating as usual will produce the desired contour mappings (Figure 2.4).

The next mapping limitation occurs in situations where highly convoluted contours, with extreme narrowings, are mapped interior to interior. The problem with this mapping situation comes from our algorithm's interior to interior dependence on the overlap region bounding box's center coordinate for the tenative coordinate mapping. For the section of the contour near the coordinate center, where the center coordinate is central, the tentative coordinate mappings are fairly good. However, for the section of the contour on the other side of the narrowing, where the center coordinate is no longer central, the tentative coordinate mapping is erroneous. The limitation comes when the tentative mapping is so bad that the continuity recognition procedure fails. This causes the contours to be incorrectly left unconnected [HOGAN,1985].

Our solution to this situation is relatively simple and within the scope of our algorithm. Segmenting the convoluted contour at the extreme narrowings, allows treatment of each open segment of the convoluted contour as a separate entity. By utilizing our existing algorithm, we can produce new centers for these separate contours and thereby generate coordinate mappings. These mappings will result in a better approximation of the original object. To incorporate this capability into our present algorithm would only require a means for partitioning the convoluted contour. This partitioning method can be achieved either through user intervention or through some automatic mechanism.

The next mapping limitation also deals with interior to interior contour mapping situations. In cases where sections of a contour are closely parallel with the connection vector drawn from the center coordinate of the inner contour, erroneous mappings are produced (Figure 4.1). Appropriate connections are generated for segments of the outer contour which are nearly perpendicular to the tentative connection vector; however, the tentative connections start to falter as the contour segment nears parallel with the connection vector.

The same solution recommended for handling highly convoluted contours with extreme narrowings will correct this problem. The quality of the tentative coordinate mapping can be greatly improved by partitioning the original contour into open segments and mapping them separately.

The final limitation concerns an interior to interior mapping in which the inner contour is not contained in the outer contour. This situation is indicative of contour data taken from a toroidal object. The limitation of our algorithm in this case is caused by using a tentative connection vector originating from the center of the inner contour. Since the two contours are not mutually centered, the displacement between the two center coordinates results in only generating mappings for that section of the outer contour which is on the same side of the tentative connection vector (Figure 4.2). The end result is a partial mapping of the two contours which really should be totally connected.

Our suggested solution to this mapping problem is again based on a concept described in the Christiansen paper [CHRISTIANSEN,1978]. For this situation, Christiansen recommends a translation procedure onto a unit square, centered at (0,0). The idea behind this procedure is to translate the two contours in such a way that they become mutually centered within the unit square. Once translated, our interior to interior algorithm would produce the desired tentative mappings for the contours' original coordinates. This procedure would then allow

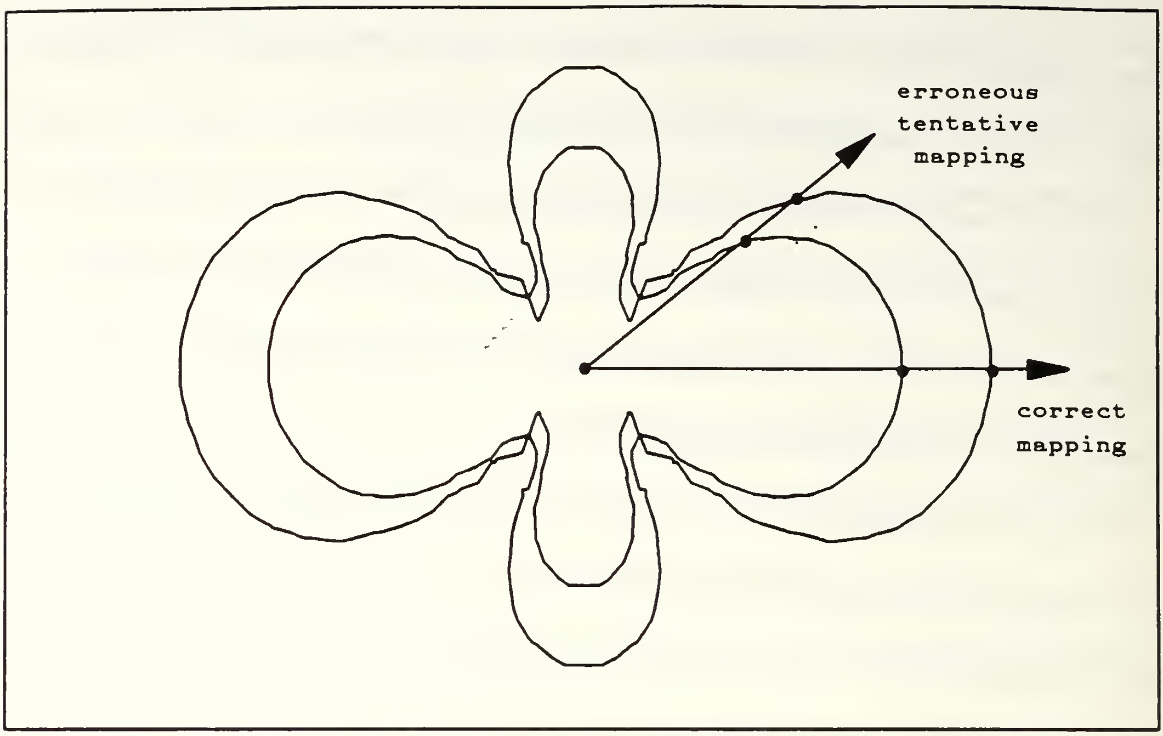


Fig. 4.1 - Example of situation resulting in an erroneous tentative coordinate mapping where contour segment becomes near parallel with the tentative connection vector.

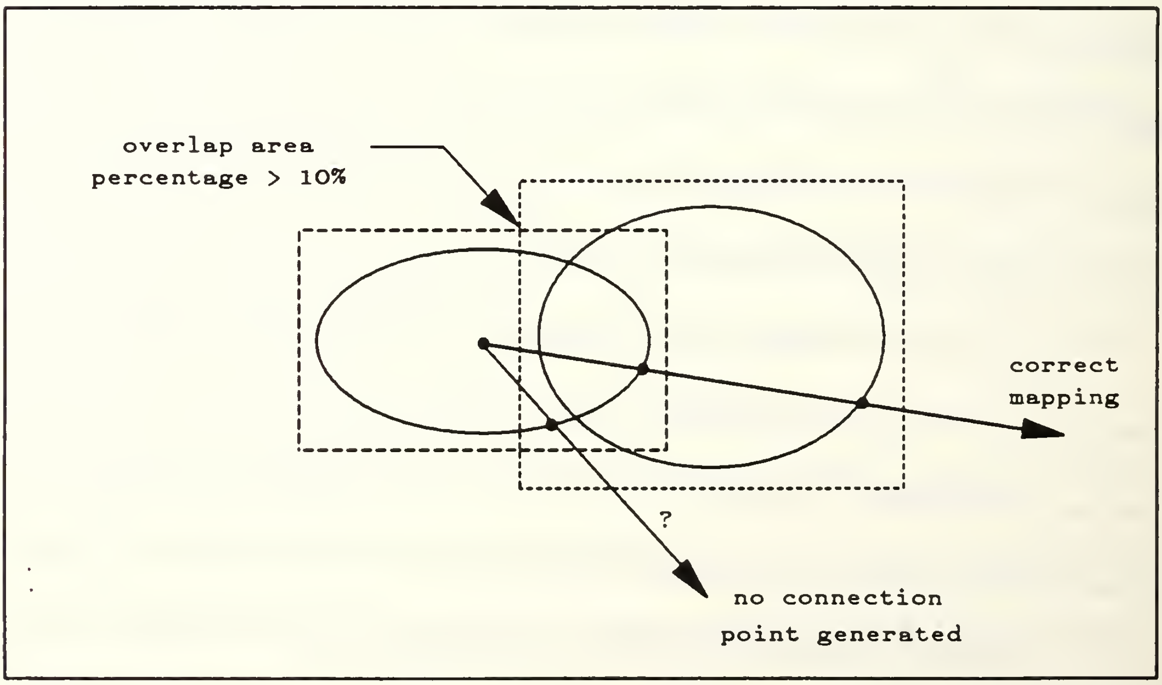


Fig. 4.2 - Example of a situation where two contours are mapped interior to interior which would result in an incomplete mapping.

the appropriate connections to be formed in the final step of our surface connection algorithm.

It has been our purpose in this chapter to discuss the limitations of our surface construction algorithm and provide our suggested solutions. We contend that our algorithm resolves the multiple contours per plane and partial mapping problems. Additionally, with the added features of contour editing and heuristic relaxation our algorithm can handle mapping situations that would otherwise be neglected. However, we must concede that our algorithm is not a total solution to the surface construction from planar contour data problem.

V. CONCLUSION

It has been our purpose in this paper to present an expanded algorithm for the surface construction of a three-dimensional object from a set of its planar contours. The main thrust of this paper has been devoted to the capabilities of our surface construction algorithm. Specifically, our algorithm's ability in handling multiple contours per plane and partial contour mapping problems as well as user interaction procedures for editing contours and relaxing heuristics have been presented. Additionally, we identified the limitations of our algorithm and discussed our proposed solutions for these problems.

Although we have expanded our algorithm beyond what was presented by Hogan [HOGAN,1985], we still have not provided a complete solution to the contour mapping problem. Further work is needed to resolve the limitations of our surface construction algorithm as described in Chapter IV. It is quite possible that the corrections of the limitations identified will not yield a complete solution to the contour mapping problem. However, their rectification will greatly enhance our algorithm's ability in handling surface reconstruction from planar contours.

LIST OF REFERENCES

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

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

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

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

Keppel, E., "Approximating Complex Surfaces by Triangulation of Contour Lines", *IBM Journal of Research and Development,* Vol. 19, January 1975, pp. 2-11.

Hogan, P. G., "Surface Construction from Planar Contours", Master's Thesis, Department of Computer Science, Naval Postgraduate School, Monterey, California, December 1985.

Zyda, M. J., "Algorithm Directed Architecture for Real-Time Surface Display Generation", Ph. D. Dissertion, Department of Computer Science, Sever Institute of Washington University, Saint Louis, Missouri, January 1984.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Superintendent Attn: Library (Code 0142) Naval Postgraduate School Monterey, California 93943	2
3.	Chairman (Code 52) Department of Computer Science Naval Postgraduate School Monterey, California 93943	1
4.	Computer Technology Curricular Officer (Code 37) Naval Postgraduate School Monterey, California 93943	1
5.	Michael J. Zyda (Code 52Zk) Department of Computer Science Naval Postgraduate School Monterey, California 93943	2
6.	C. Thomas Wu (Code 52Wq) Department of Computer Science Naval Postgraduate School Monterey, California 93943	1
7.	LCDR Allan R. Jones USN 3912 Plum Lane Chesapeake, Virginia 23321	6
8.	LCDR Patrick G. Hogan USN 102 Borden Avenue Wilmington, North Carolina 28403	1

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

10. CAPT Roger Diehl USMC 1
1105 Richmond Drive
Stafford, Virginia 22554

218766

Thesis
J677 Jones
c.1 Surface reconstruction from planar contours.