

SURFACE CONSTRUCTION FROM PLANAR CONTOURS*

MICHAEL J. ZYDA,† ALLAN R. JONES, and PATRICK G. HOGAN

Naval Postgraduate School, Code 52, Dept. of Computer Science, Monterey, California 93943

Abstract—Many applications of computer graphics involve the display 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, that approximate the original three-dimensional solid. In this paper, we present an expanded algorithm that 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 the proposed solutions to those limitations.

1. INTRODUCTION

Many applications of computer graphics involve the display 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 (Fig. 1). The intersection between these two-dimensional parallel planes and the three-dimensional object forms the contours that lie along the solid's exterior and interior surfaces. The contours appear as line segments on the parallel planes and are either closed loops, open segments, or single points. The main purpose of a surface construction algorithm is the formation of surface patches between the 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 appropriate 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 (Fig. 2). This tiling operation is executed for all the line segments in the identified contours.

The initial part of this paper is a brief review of previous surface construction algorithms, concentrating mainly on their capabilities and limitations[1-6]. The remainder focuses on a new surface construction algorithm originally presented in [4]. This algorithm is more comprehensive in that it can handle multiple contours per plane and partial contour mappings. Nonetheless, that algorithm also does not provide a complete solution to the surface construction problem. Following the discussion of that algorithm, we present a further expanded algorithm which attempts to resolve each of that algorithm's limitations.

* This work has been supported by the Naval Postgraduate School Foundation Research Program, the U.S. Army Combat Developments Experimentation Center, and a grant from the Naval Ocean Systems Center, San Diego.

† Contact author.

2. 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 several articles[1-6]. 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[5]. 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[1].

Our surface construction algorithm is based on the efforts of [1-2, 4]. In order to fully understand the underlining problem of surface construction, a brief summary of the previous literature is presented. This summary focuses mainly on each of the algorithm's capabilities and limitations.

2.1 Previous surface construction algorithms

Fuchs' algorithm for surface construction is presented in [2]. His problem statement 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 (Fig. 3). 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 adjacent planes, partial contour mappings, or open (non-closed) contours (Fig. 4). The problem with multiple contours on adjacent 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 Fuchs' algorithm is that it performs a complete contour-to-contour triangulation between adjacent 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 dissimilarly sized contours. The third limi-

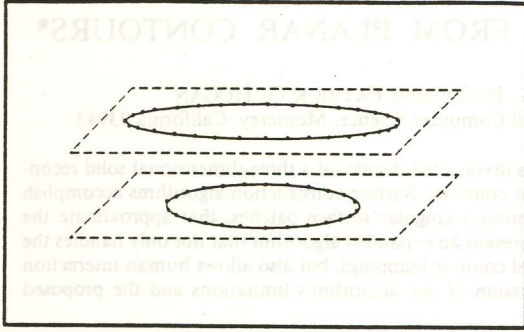


Fig. 1. Two contours on adjacent, parallel planes.

tation in Fuchs' 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 [1], we see an algorithm that is similar to Fuchs'. 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[4]. This procedure can be quite time consuming, depending on the complexity of the data base[1]. 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. (Fig. 5). 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. (Fig. 6).

3. Triangulate as usual[1].

In general, this algorithm introduces a new node between the two planar contours. This new node is used to form single, connected regions that 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 easily implemented, fast, and works well as long as the two contours/loops are mutually centered and are reasonably similar in shape and size[1]. 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 [6] is basically an extension of Fuchs' and Christiansen's algorithms. This extension includes the capability to handle contour defined objects that are highly branched and have holes. Handling of multiple contours on adjacent planes is achieved by the following:

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 [6].

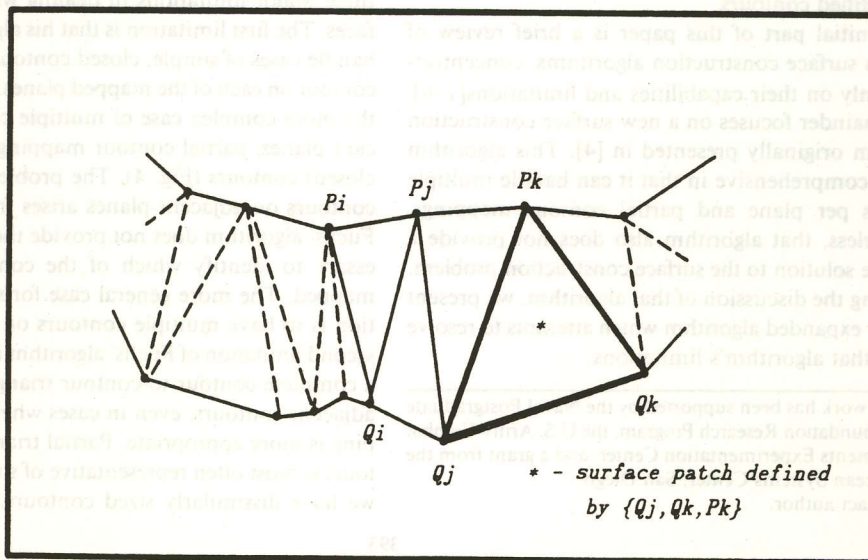


Fig. 2. Mapped connections into triangulated surface patches.

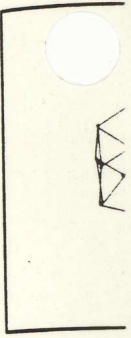


Fig. 3. Triangulation

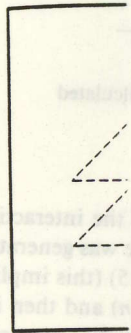
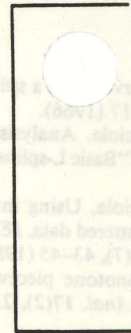
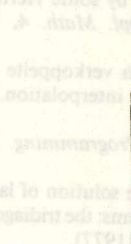


Fig. 4. I



F



REPRODUCED AT GOVERNMENT EXPENSE

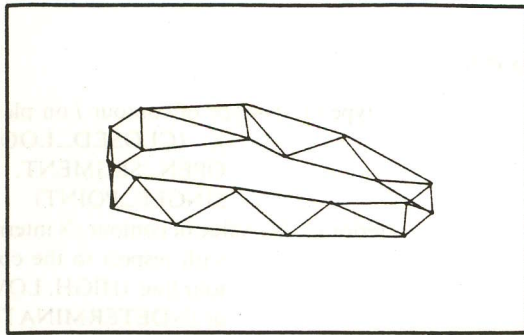


Fig. 3. Triangulated pair of simple, closed contours.

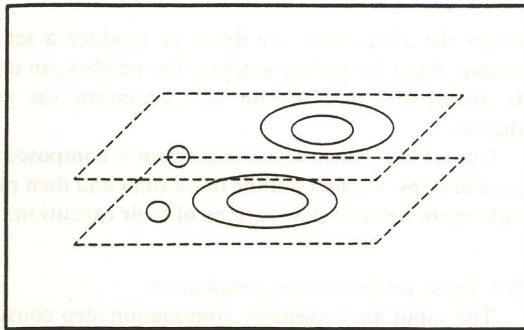


Fig. 4. Example of multiple contours per plane.

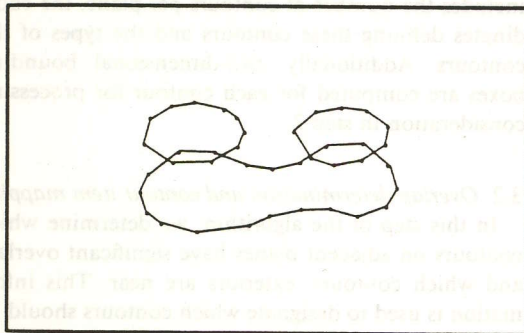


Fig. 5. Simple case of branching.

Once the concatenation process is completed, Shantz uses Fuchs' 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 some eighty 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[3] 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[4]. 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 in [4] 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 ones for handling branching as de-

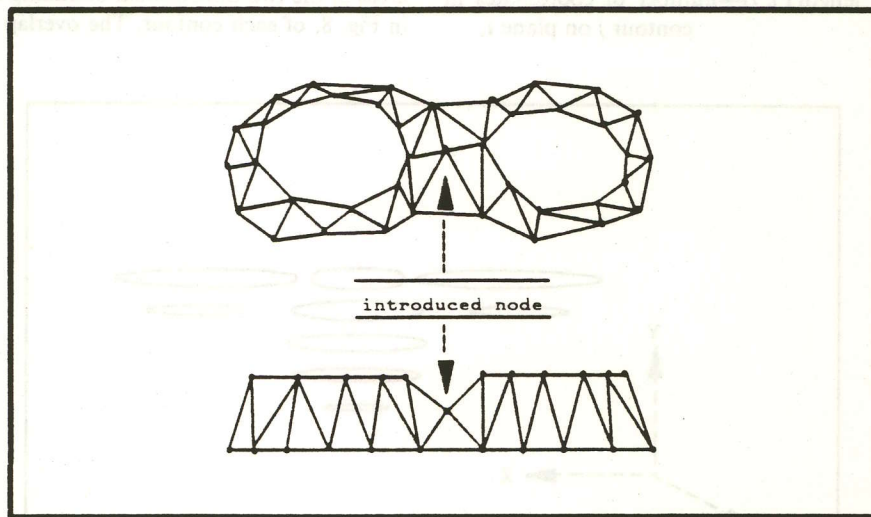


Fig. 6. Triangulation scheme for branching.

scribed in the Christiansen paper and 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 the handling of branching as described in the Christiansen paper. A discussion of our algorithm follows, with a proposed solution for handling cases involving branching.

3. SURFACE CONSTRUCTION ALGORITHMS

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 (Fig. 7) 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[4]:

- 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 .

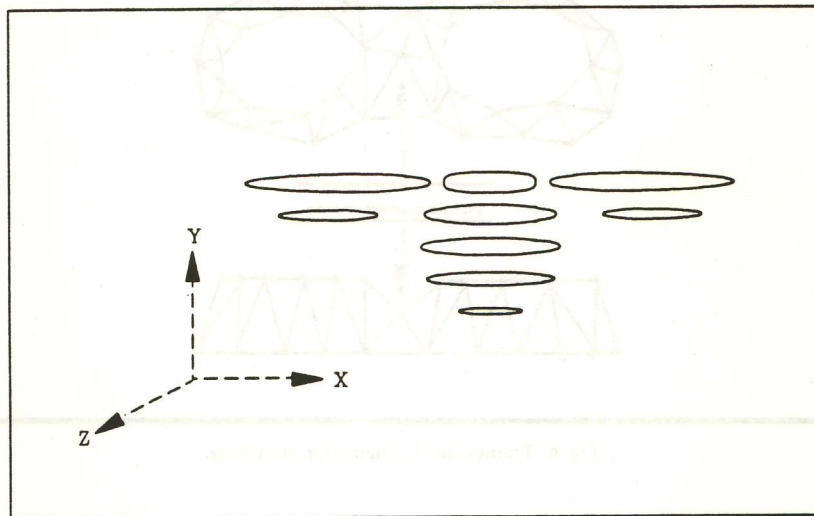


Fig. 7. A partial set of planar contours from a 3D Z^2 -orbital of a hydrogen molecule.

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 .

From the above data, we desire to produce a set of output triangular surface patches. The patches can then be stored in a database for later geometric use and display.

Our surface construction algorithm is composed of several steps. We first outline those steps and then provide more detailed descriptions of their execution.

3.1 Input and inventory compilation

The input and inventory compilation step consists of reading in the two-dimensional contours and extracting the data required by the algorithm. 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 computed for each contour for processing consideration in step 2.

3.2 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 Fig. 8, of each contour. The overlap percentage is

Fig. 8. Two-dime

used to give pr
the highest perc
In this step c
sistency checks
sistency check

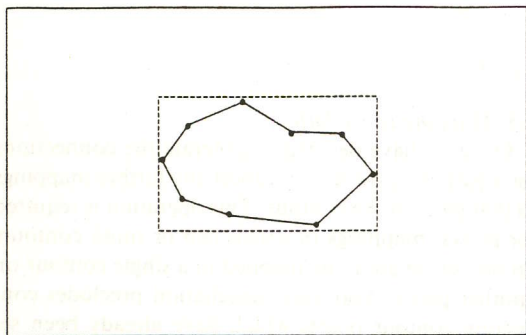


Fig. 8. Two-dimensional bounding box used for determining overlap percentage value.

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 Fig. 9. 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 to its solid interior and contour 1 on plane 1 is high valued, this mapping can be eliminated.

The contour interior specifications are also used to determine whether the mapping is interior to interior

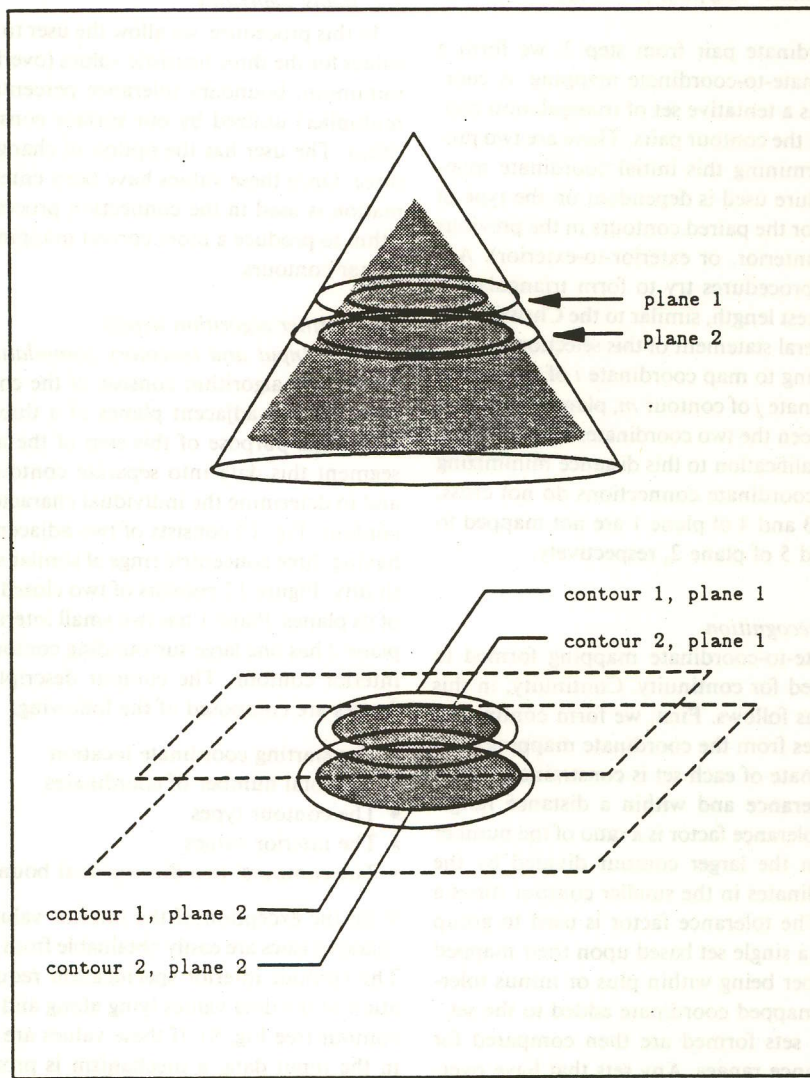


Fig. 9. Example of consistency check using item interior specifications with overlap percentage values.

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 that 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).

3.3 Form coordinate mapping for mapped contour pairs

For each coordinate pair from step 2, 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.

3.4 Continuity recognition

The coordinate-to-coordinate mapping formed in step 3 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.

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

3.6 Connection formation

We generate the coordinates for the triangulation connections specified in step 4. "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.

3.7 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 a more correct mapping between the planar contours.

3.8 Further algorithm details

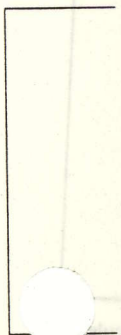
3.8.1 *Input and inventory compilation.* The input data to the algorithm consists of the contour descriptions for two adjacent planes of a three-dimensional solid. The purpose of this step of the algorithm is to segment this data into separate contour descriptions and to determine the individual characteristics of each contour. Fig. 10 consists of two adjacent planes, each having three concentric rings of similar shape and continuity. Figure 12 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 following:

- The starting coordinate location
- The total number of coordinates
- The contour types
- The interior values
- 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 contour interior specification requires an evaluation of the data values lying along and interior to the contour (see Fig. 9). 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 encoun-

tered in mul
are closely s
form of hun
which pairs c
an interior
situation is r
contours, it c
surface const
3.8.2 *Over*
The
cedure
process by v
ping assignm
that are nec
dimensional b
specification
ponent in th
tours.

The overla
procedure is
First, the two
tour on plane
dimensional
2. The coord
are the mini



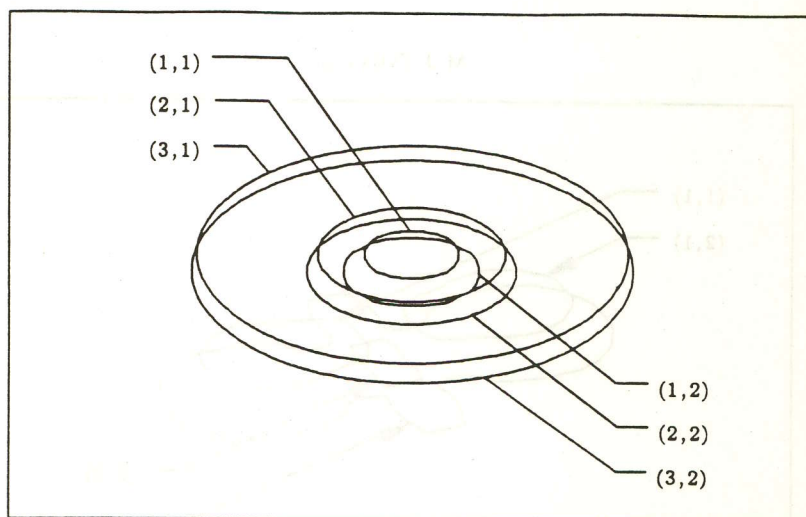


Fig. 10. Example of multiple contours per plane on adjacent planes.

tered 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, i.e., there are no concentric contours, it can be set to INDETERMINATE without surface construction degradation.

3.8.2 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 that 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 that 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 nonmatching. Finally, all contours with low overlap percentages and matching interiors are zeroed in the overlap table.

Figures 14 and 15 graphically represent the overlap determination and contour mapping for Figs. 10 and 12. Included in these figures are the overlap tables produced by this procedure. The table in Fig. 14 shows three valid overlap percentages for three different contour pairs: (1, 1)-(1, 2), (2, 1)-(2, 2), and (3, 1)-(3, 2).

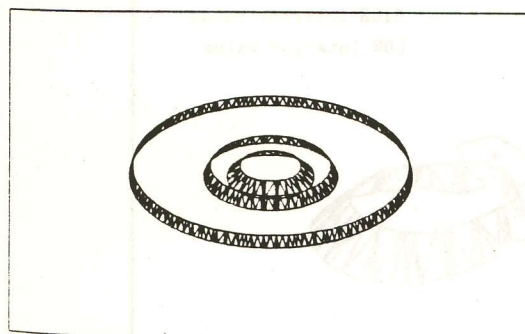


Fig. 11. Connection of Fig. 10.

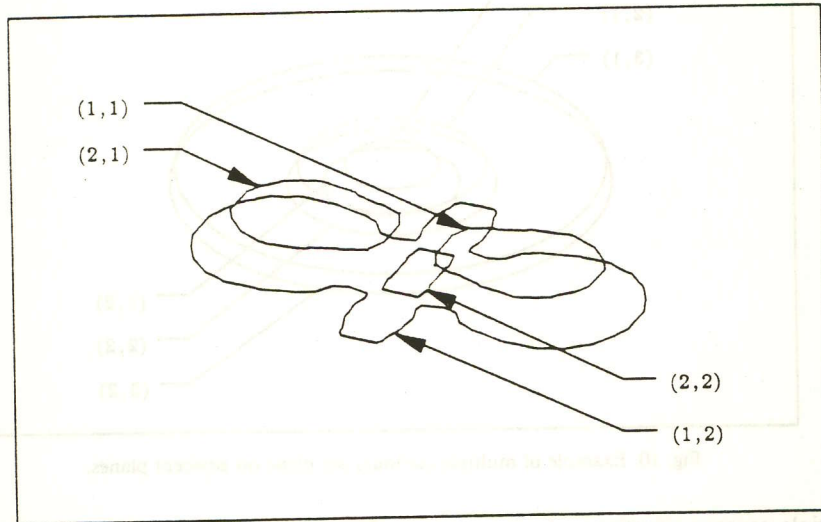


Fig. 12. Example of a set of contours requiring partial mappings and an exterior-to-exterior mapping: (1, 1) and (2, 1) to (2, 2).

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 Fig. 15 shows two high overlap percentages and two low overlap percentages. These data indicate 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).

3.8.3 *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 performed in a largest

to smallest overlap percentage order. 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

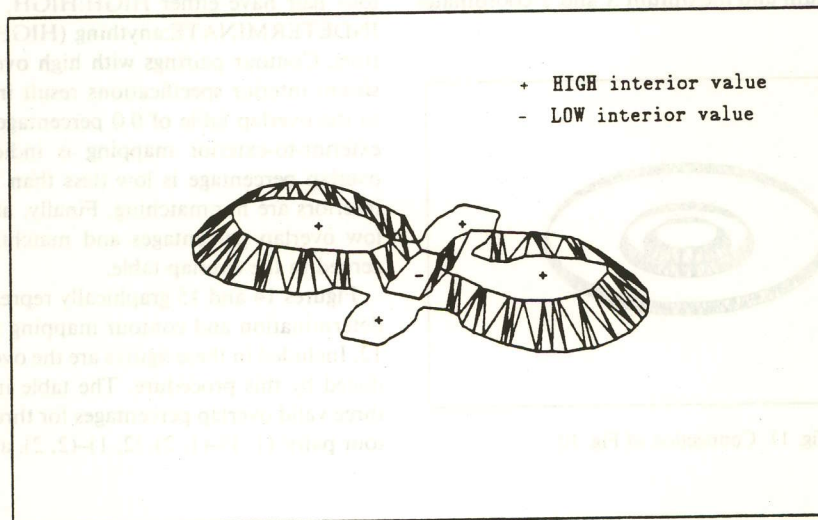


Fig. 13. Connection of Fig. 12, with contour interior values for each contour.

each coordin
which coord
vector draw
contour thro
(see Fig. 16)
coordinate s
from the cen
Also, the ou
of the vecto
ordinates sel
as the tenta
inner coord
distance fro
mapped out
c th
ta the
The tent
good. Due

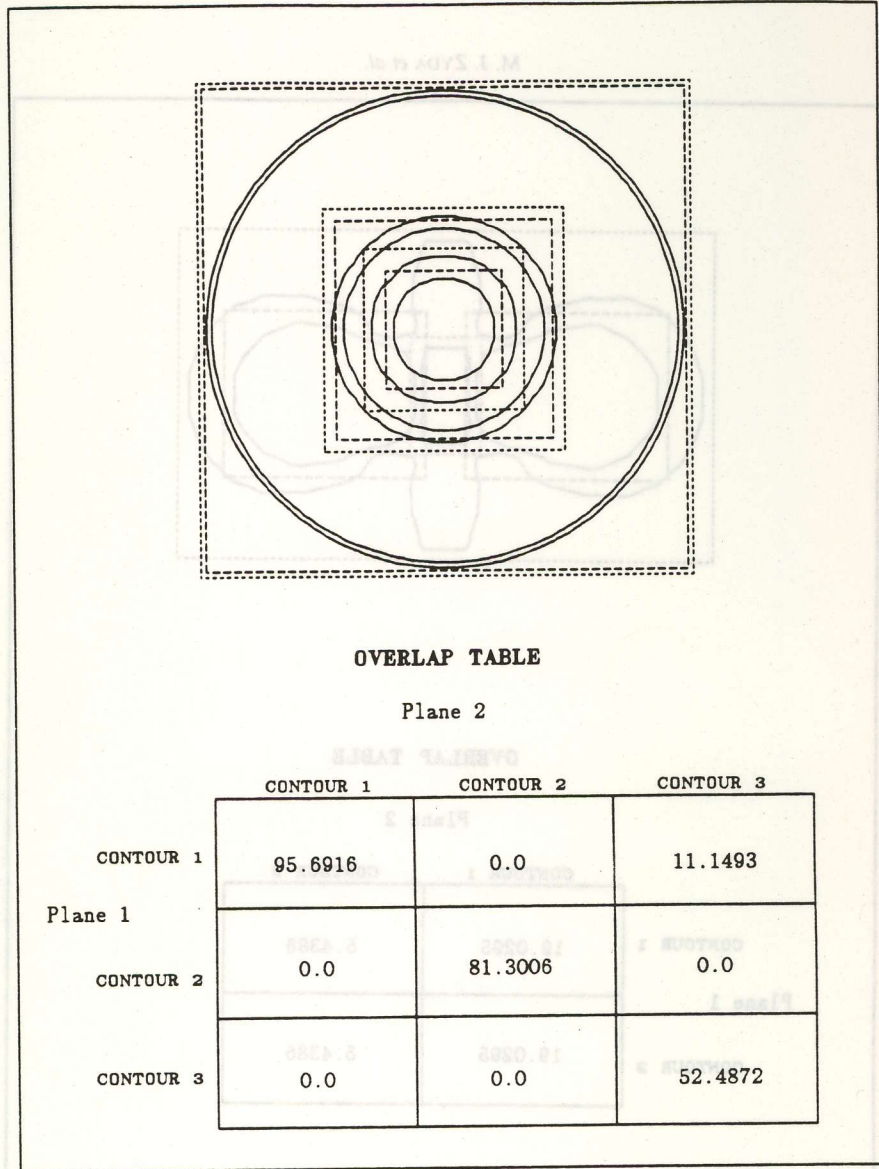


Fig. 14. Bounding boxes and overlap table produced for Fig. 10.

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 Fig. 16). 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 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 Fig. 10 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 Fig. 17 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.

3.8.3.1 *Coordinate mapping: 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.

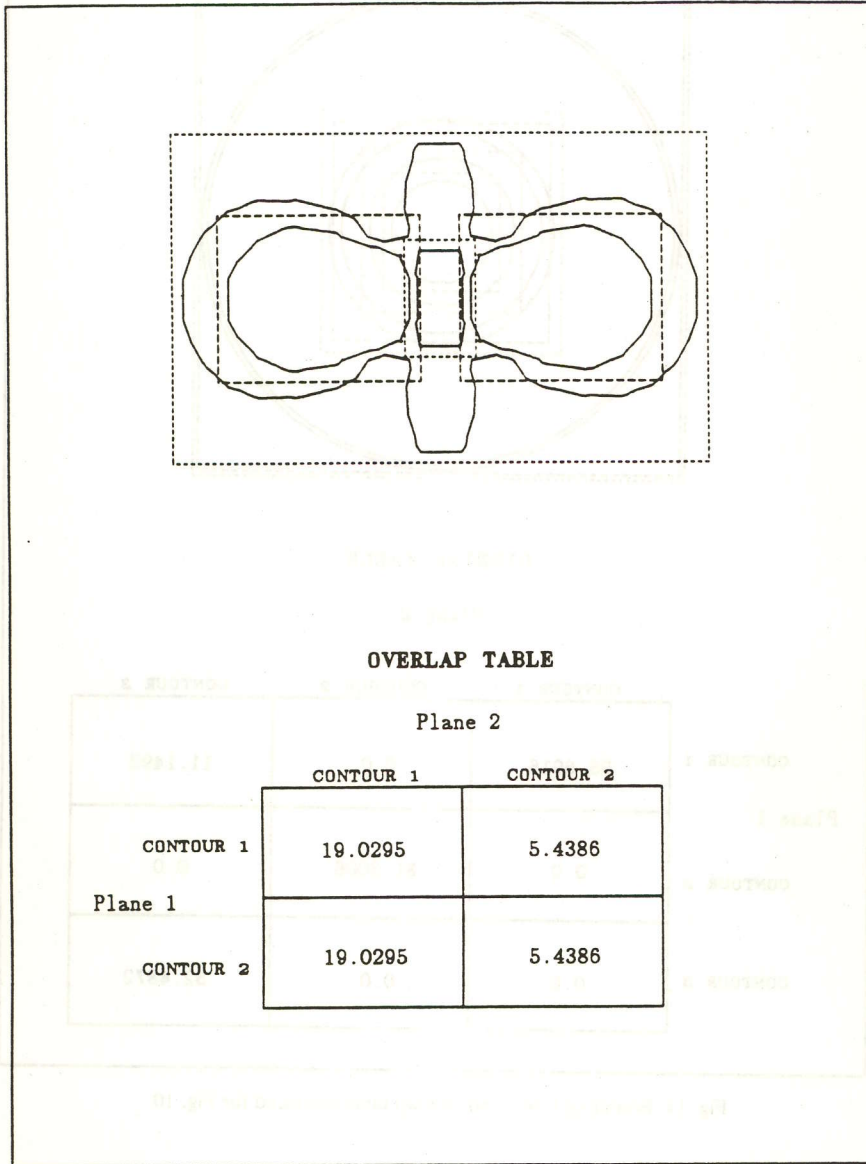


Fig. 15. Bounding boxes and overlap table produced for Fig. 12.

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 coordinate's 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 below.)

To illustrate this continuity set assignment, let us refer to the example in Fig. 17. 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.

Fig. 16.

The initial s is a fast methc In addition to the tentative r and maximur set. The inal in the The particular int dure placed a ordinate map This can be at ilar shapes ar mapping pair resulted in 5 i ranges (Fig. 1

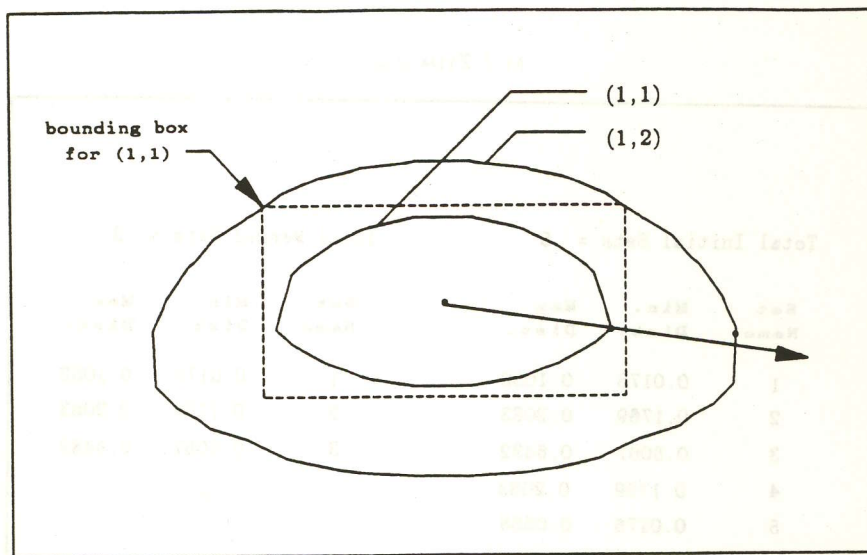


Fig. 16. Vector radiating from center coordinate through the interior coordinate towards the outer contour for tentative mapping.

The 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 Figs. 10 and 12 are of particular interest. This step of the continuity procedure placed all of the tentative mappings for the coordinate mapping pairs for Fig. 10 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 Fig. 12 resulted in 5 initial continuity sets with varying distance ranges (Fig. 18).

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 Fig. 10 due to its singular initial continuity set. Figure 18 shows the initial sets with distance ranges and the merged sets with distance ranges for the contour pairing (1, 1)-(1, 2) of Fig. 12. It is shown that the 5 initial continuity sets have been merged into 3 sets of nonoverlapping distance range.

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.

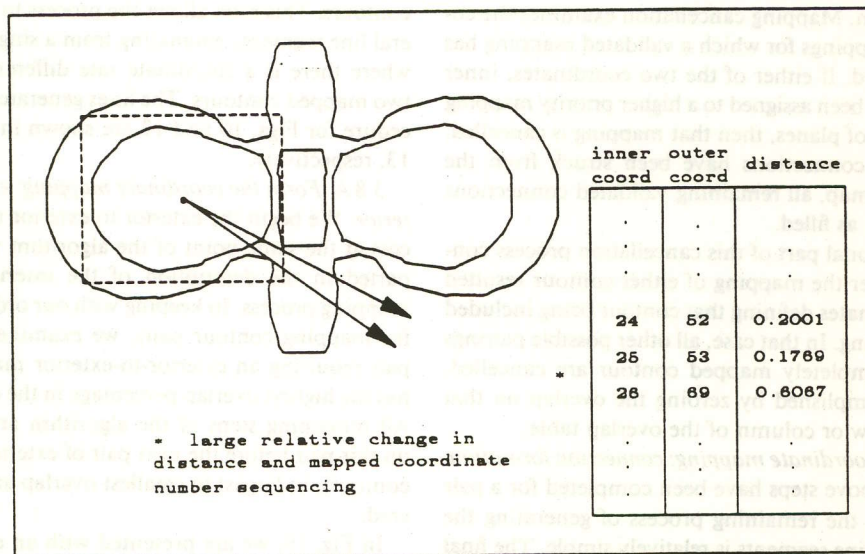


Fig. 17. Example of a case where tentative mapping coordinates and associated distances vary greatly.

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. 18. Initial continuity sets and merged continuity sets for the contour pair (1, 1)-(1, 2) of Fig. 12.

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.

3.8.3.2 Coordinate mapping: 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 part 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.

3.8.3.3 Coordinate mapping: 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 point-

ers 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 Christensen in [1]. 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 Figs. 10 and 12 are shown in Figs. 11 and 13, respectively.

3.8.4 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 Fig. 19, we are presented with an enlarged view of the bounding box overlap area of the contour pairing (1, 1)-(2, 2) of Fig. 12. This area of overlap contains all of the coordinates from both contours that are involved in the connection mapping. The first operation performed on an exterior-to-exterior mapped overlap

pair is the de-
both contour
tour with the
overlap area
mapping betw
of coordinate
contour pair
coordinate in
the shortest
simpler versi
connection s
ings. The p
map for the
sets is not ne
due to the rel
comprise the

Once we h
the same ma
formation pro
terior mappi
again uses th
uous segme
The continu
triangular su
ordinates w
formation fe
(2, 2) and (2
3.8.5 Re
heuristics p
adjust the t
construction
connections
v coordinate
first
minimum
percentage
planes. The

Fig. 19

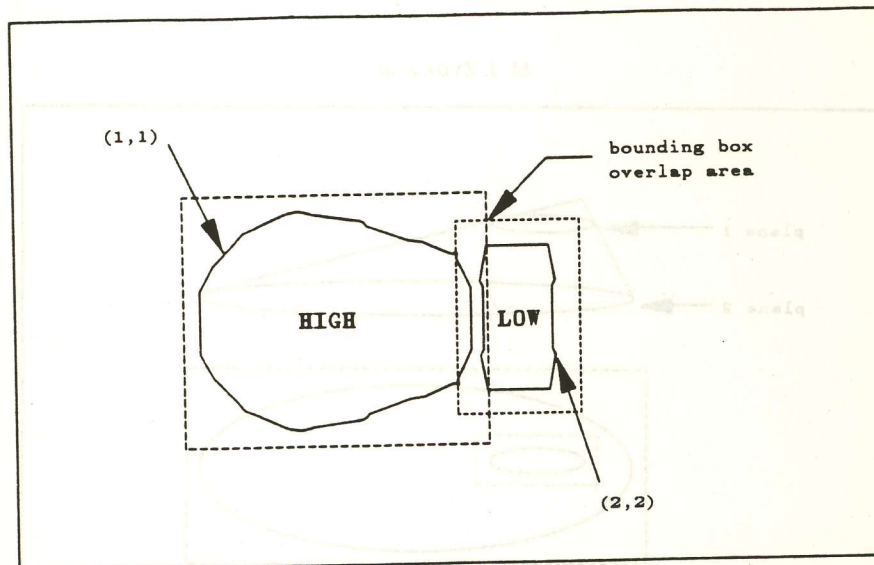


Fig. 19. Bounding box overlap for exterior-to-exterior mapping. Only the coordinates within the overlap area are mapped.

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 Fig. 12 is shown in Fig. 13.

3.8.5 Relax heuristics. The purpose of the relax heuristics procedure is to allow the user the option to adjust the three heuristic values used by the 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 the 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 nonzero percentages below the minimum, with nonmatching 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 20 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 Fig. 20, an appropriate connection can be generated.

The second heuristic value is the boundary tolerance percentage. In the initial two steps of our algorithm 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 Fig. 21 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

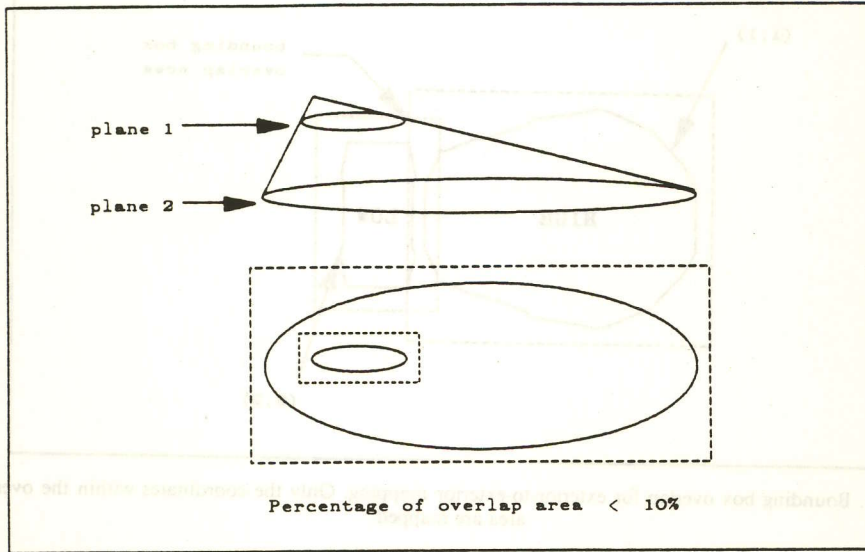


Fig. 20. Example of a contour pair which should be mapped, but would be disregarded due to overlap percentage below the minimum.

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, the algorithm utilizes a tolerance factor for the 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. The window value is a constant value used for

the selection of appropriate mapping connections. Again, by allowing the user to adjust this heuristic value, we provide opportunities to handle mapping cases that might otherwise not be included by our preset value.

3.8.6 *User interaction.* The purpose of user interaction is to allow the identification and mapping of contours that pose a problem for our surface construction algorithm. 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 can be reprocessed by our algorithm. The user can then save the

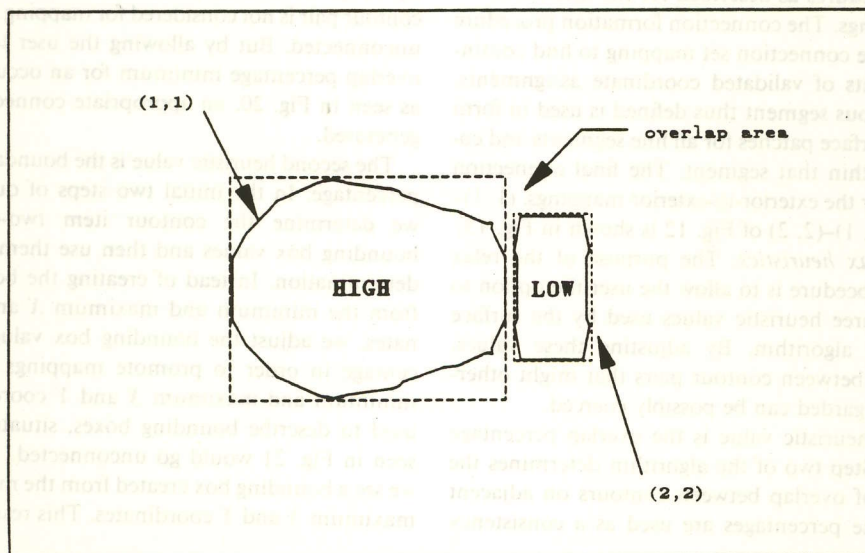


Fig. 21. Example of contours' 2D bounding boxes created strictly from the min and max X and Y coordinates. Resulting overlap = 0.

data with surface 1
 3.8.7
 presented an outline
 surface construction
 devoted to the str
 capabilities for
 partial contour
 This algorithm
 algorithms in st
 of contours. In
 user interaction
 procedure, our
 that would oth
 provided more
 algorithm, we

4.
 The first ma
 handle involve
 one plane to 1
 plane (Fig. 5).
 algorithm produ
 cause of missi
 problem is bas
 tiansen paper[
 troduce a new
 branches. The
 the ave of
 levels
 gulating as us
 mappings (Fig
 The next al
 where highly c
 rowings, are m
 with this map
 rithm's interio

data with the special cases marked for later complete surface regeneration.

3.8.7 Algorithm details summary. We have presented an outline and 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, and heuristic relaxation. This algorithm has proved to outperform all previous algorithms in surface construction via the triangulation of contours. In addition, with the incorporation of the user interaction 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.

4. ALGORITHM LIMITATIONS

The first mapping situation our algorithm cannot handle involves simple branching of one contour on one plane to two or more contours on an adjacent plane (Fig. 5). 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[1]. 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 (Fig. 6).

The next algorithm 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 tentative 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 [4].

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 algorithm 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 (Fig. 22). 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

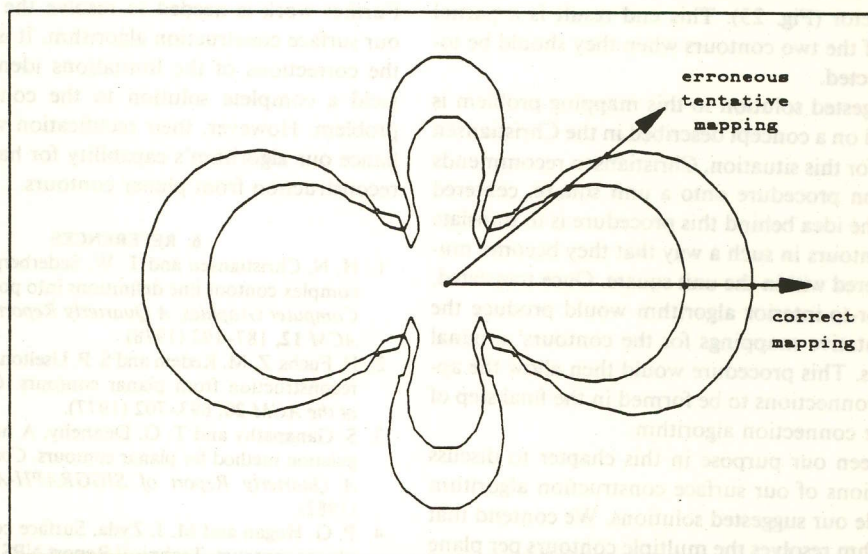


Fig. 22. Example of situation resulting in an erroneous tentative coordinate mapping where contour segment becomes near parallel with the tentative connection vector.

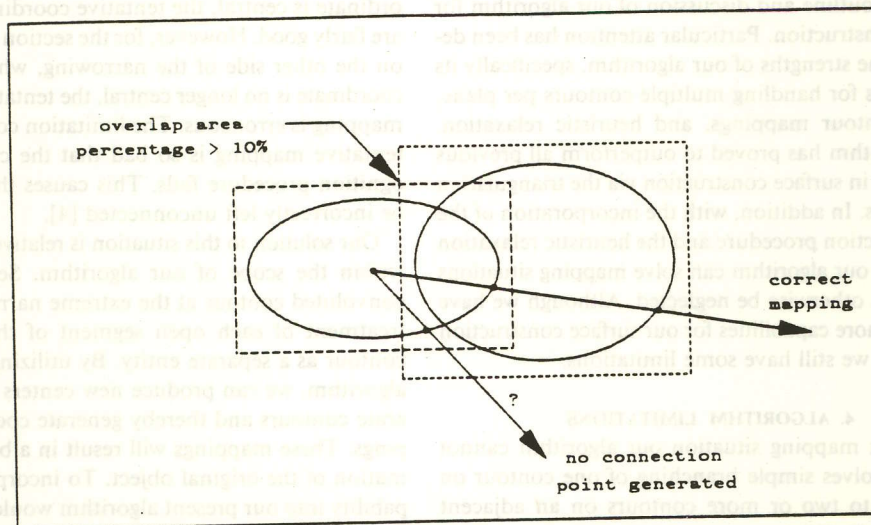


Fig. 23. Example of a situation where two contours are mapped interior-to-interior which would result in an incomplete mapping.

corrects 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 of the algorithm 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 (Fig. 23). This end result is a partial mapping of the two contours when they should be totally connected.

Our suggested solution to this mapping problem is again based on a concept described in the Christiansen paper[1]. 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 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 user interaction 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.

5. 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 to handle multiple contours per plane and partial contour mapping problems as well as user interaction and heuristic relaxation have been presented. Additionally, we have identified the limitations of our algorithm and discussed our proposed solutions for these problems.

Although we have expanded our algorithm beyond what was presented in [4], 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. It is probable 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 capability for handling surface reconstruction from planar contours.

6. REFERENCES

1. H. N. Christiansen and T. W. Sederberg, Conversion of complex contour line definitions into polygonal mosaics. *Computer Graphics: A Quarterly Report of SIGGRAPH-ACM* 12, 187-192 (1978).
2. H. Fuchs, Z. M. Kedm and S. P. Usetlon, Optimal surface reconstruction from planar contours. *Communications of the ACM* 20, 693-702 (1977).
3. S. Ganapathy and T. G. Dennehy, A new general triangulation method for planar contours. *Computer Graphics: A Quarterly Report of SIGGRAPH-ACM* 16, 69-74 (1982).
4. P. G. Hogan and M. J. Zyda, Surface construction from planar contours. Technical Report NPS52-86-002, Monterey, California: Department of Computer Science, Naval Postgraduate School (1986).
5. E. Keppel, Approximating complex surfaces by triangulation of contour lines. *IBM Journal of Research and Development* 19, 2-11 (1975).
6. M. Shantz, Surface definition for branching, contour-defined objects. *Computer Graphics: A Quarterly Report of SIGGRAPH-ACM* 15, 242-266 (1981).

Abst
are c
fixed
calcul
simp
sche
met
used

Computer-g
require a la
puter or tra
port we in
compression
or transmis

We have
by an ink
Lund[1-2].
perimposin
yellow and
mr
the

The ink jet
ink dot or
picture con
tain pictur
similar to
employed.
to a halfto
False inten
of filled dot
1 shows ex

Notice th
of black do
16. Figure
formation
the presenc
pictures ar
the basic p
ranged in r
elements.

In this i
slightly dif
One of the
pographic
name, also
CUBE," is
& J. T.
of t
I. de

The pict
graphic so
to control