

Optimal Surface Reconstruction from Planar Contours

H. Fuchs, Z.M. Kedem, and S.P. Uselton
The University of Texas at Dallas

In many scientific and technical endeavors, a three-dimensional solid must be reconstructed from serial sections, either to aid in the comprehension of the object's structure or to facilitate its automatic manipulation and analysis. This paper presents a general solution to the problem of constructing a surface over a set of cross-sectional contours. This surface, to be composed of triangular tiles, is constructed by separately determining an optimal surface between each pair of consecutive contours. Determining such a surface is reduced to the problem of finding certain minimum cost cycles in a directed toroidal graph. A new fast algorithm for finding such cycles is utilized. Also developed is a closed-form expression, in terms of the number of contour points, for an upper bound on the number of operations required to execute the algorithm. An illustrated example which involves the construction of a minimum area surface describing a human head is included.

Key Words and Phrases: surface reconstruction, contour data, serial sections, three-dimensional computer graphics, minimum cost paths, continuous tone displays

CR Categories: 5.25, 5.32, 8.2

Copyright © 1977, Association for Computing Machinery, Inc. General permission to republish, but not for profit, all or part of this material is granted provided that ACM's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Association for Computing Machinery.

This research was supported in part by NSF Grant MCS77-03905.

Authors' address: The University of Texas at Dallas, Richardson, TX 75080.

A version of this paper was presented at SIGGRAPH 77: The Fourth Annual Conference on Computer Graphics and Interactive Techniques, San Jose, California, July 20-22, 1977.

The remaining papers appear in Computer Graphics (ACM SIGGRAPH newsletter), Vol. 11, No. 2 (Summer 1977). This special issue is available prepaid from ACM, P.O. Box 12105, Church Street Station, New York, NY 10249; ACM or SIGGRAPH members \$12.00, all others \$16.00.

Introduction

Many scientific and technical endeavors involve interactions with solids and surfaces. In biological research, medical diagnosis and therapy, architecture, and automobile and ship design, for instance, these structures are often so detailed and the interaction with them so extensive that automated assistance of some kind is almost a necessity. Such assistance is seriously hampered by the difficulty of effectively defining these three-dimensional structures to the computer.

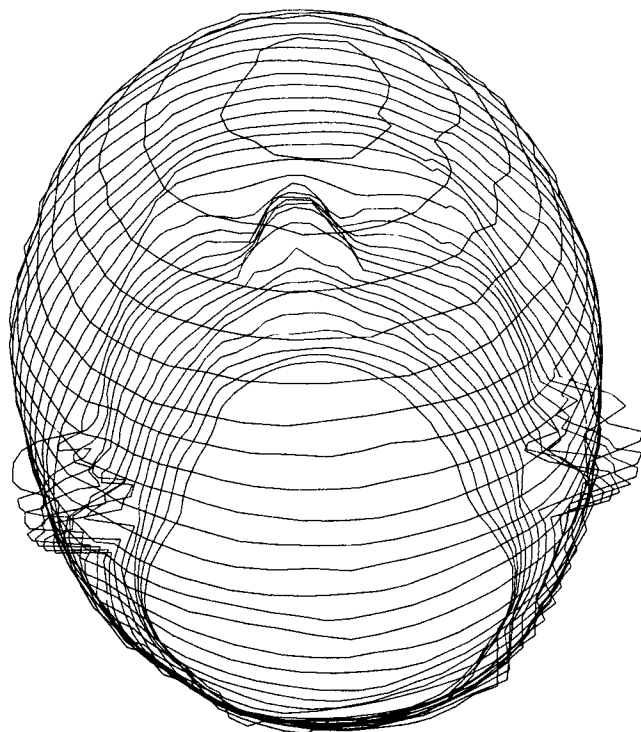
In microscopy, for instance, one often deals with three-dimensional objects, but, because light microscopes of any significant magnification are only monocular, the three-dimensional structure has to be reconstructed from a sequence of two-dimensional images. Two methods are often employed to obtain these images. If the specimen is sufficiently transparent, the microscope can simply be focused at various levels into the structure. The image obtained at one such setting consists primarily of those parts of the solid which are in focus, namely a cross-section of the solid at a given depth. These images are then recorded for further study. This method, however, cannot be used for opaque specimens. Such specimens are cut into thin slices; then each slice's microscopic image is examined and recorded individually.

The cross-sectional images are then used to reconstruct the three-dimensional structure. A simple manual method is sometimes employed. The images are transferred to photographic transparencies sized for table-top observation and are stacked in sequence with transparent spacers of appropriate thickness. The resulting semitransparent stack roughly approximates the original three-dimensional structure and can be examined from various angles.

Since the outside surface of the three-dimensional structure is often of greatest significance, an intermediate step of determining, either manually or automatically, the boundary at each cross-section is often included [11]. The reconstructed structure then appears similar to a set of wire-frame contours over which the surface "skin" is thought to lie [9]. Since the display of the structure is most often in the form of these wire-frame contours, the task of creating the surface is left to the viewer's imagination. This is a nontrivial mental exercise when dealing with any but the simplest structures (Figure 1).

Although wide interest has been indicated in the development of an automatic procedure for defining a surface over wire-frame contours, to the best of our knowledge, only a single object-constructing method has appeared in the literature, a volume-approximating procedure due to E. Keppel [8]. This method introduced a reduction of the problem to finding a path in a directed graph and used certain heuristics for choosing an appropriate path. The method to be presented here also reduces the problem to one in graph theory, but does not utilize any heuristics, thus allowing a general

Fig. 1. Contour data of human head.



development with a variety of possible options for choosing optimizing criteria. (A simple procedure utilizing local optimization has been previously developed by one of the authors[4].)

Statement of Problem

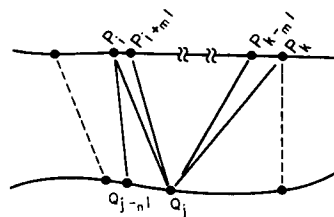
An unknown three-dimensional solid is intersected by a finite number of specified parallel planes. (The method can easily be extended to handle more general cases in which the contours do not necessarily lie on parallel planes. In the interest of simplicity of presentation, these generalizations are not considered in the sequel.) 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*. The sequence of contour segments lying on one of the parallel planes, which we assume also form a simple closed curve, is called a *contour*. The sequence of contours is used to construct a piecewise planar approximation to the original object surface. The approximating surface is constructed in a way which assures that its intersections with the parallel planes are identical to the contours lying on them.

We reduce the problem of constructing such an approximation to one of constructing a sequence of partial approximations, each of them connecting two contours lying on adjacent planes. Each of these problems can be described as follows:

Let one contour be defined by the sequence of m distinct contour points P_0, P_1, \dots, P_{m-1} , and let the other contour be defined by the sequence of n distinct contour points Q_0, Q_1, \dots, Q_{n-1} . We note that P_0 follows P_{m-1} and that Q_0 follows Q_{n-1} , and so indices of P are modulo m and indices of Q are modulo n . ($+_k$ will be used to denote addition modulo k .) We wish to create a surface between the contours P and Q . The surface is constructed of triangular tiles between these two contours. The vertices of these tiles are contour points, with the vertices 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 $\{P_i, P_k, Q_j\}$ or $\{Q_i, Q_k, P_j\}$.

We next consider a simplifying observation. Assume that the tile $\{P_i, P_k, Q_j\}$ is in the approximating surface. Then we know that the line segment $\overline{P_i P_k}$ (or $\overline{P_k P_i}$), which belongs to the intersection of the approximating surface with one of the parallel planes, has to be a part of the contour. Thus the points $P_i, P_{i+m1}, \dots, P_k$ (or the points $P_k, P_{k+m1}, \dots, P_i$) all lie on a straight line. Therefore, without changing the approximating surface, the tile $\{P_i, P_k, Q_j\}$ can be replaced by the tiles $\{P_i, P_{i+m1}, Q_j\}, \{P_{i+m1}, P_{i+m2}, Q_j\}, \dots, \{P_{k-m1}, P_k, Q_j\}$ (or it can be replaced by $\{P_k, P_{k+m1}, Q_j\}, \{P_{k+m1}, P_{k+m2}, Q_j\}, \dots, \{P_{i-m1}, P_i, Q_j\}$). (The case where $\{Q_i, Q_k, P_j\}$ belongs to the approximating surface is treated similarly.) Thus, without loss of generality, we assume that all tiles belonging to an approximating surface will be *elementary tiles*, triangles either of the form $\{P_i, P_{i+m1}, Q_j\}$ or the form $\{Q_j, Q_{j+n1}, P_i\}$. Thus 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 (Figure 2).

Fig. 2. Reduction to "elementary" tiles.



We orient the elementary tiles by writing a tile of the form $\{P_i, P_{i+m1}, Q_j\}$ as $\langle P_i, Q_j, P_{i+m1} \rangle$ and by writing a tile of the form $\{Q_j, Q_{j+n1}, P_i\}$ as $\langle Q_{j+n1}, P_i, Q_j \rangle$. The order of the three points in the sequences describing the tiles was chosen in a way which guarantees consistent orientation of the tiles' surfaces. Span $\overline{P_i Q_j}$ will be considered the left span of $\langle P_i, Q_j, P_{i+m1} \rangle$,

and span $\overline{P_{i+m1}Q_j}$ will be considered its right span. ($\overline{P_iQ_j}$ and $\overline{P_iQ_{j+n1}}$ are, respectively, the left and right spans of $\langle Q_{j+n1}, P_i, Q_j \rangle$.)

There are of course many sets of elementary tiles which could be defined over the points of the two contours. As we want the tiles which will define a surface to "fit" together, we restrict consideration to those sets of tiles which satisfy the following two conditions:

- (1) Each contour segment appears in exactly one tile in the set.
- (2) If a span appears as a left (right) span of some tile in the set, then it also has to appear as a right (left) span of at least one tile in the set.

A set of tiles satisfying these two conditions is called an *acceptable surface*.

There are still many sets of tiles which satisfy the above conditions; therefore additional criteria will be used to choose the most appropriate surfaces from among these sets.

Solution

The problem is reduced to one in graph theory. In the sequel, the terms used, unless defined, are those of [5]. We define a directed graph $G = \langle V, A \rangle$, in which the vertices correspond to the set of all possible spans between the points P_0, P_1, \dots, P_{m-1} and the points Q_0, Q_1, \dots, Q_{n-1} , and the arcs correspond to the set of all the possible elementary tiles. An arc in the graph will be incident *from* the vertex which corresponds to the tile's left span, and will be incident *to* the vertex which corresponds to the tile's right span. Thus

$V = \{v_{ij} | i = 0, 1, \dots, m-1; j = 0, 1, \dots, n-1\}$
and v_{ij} corresponds to the span $\overline{P_iQ_j}$. Furthermore

$A = \{ \langle v_{kl}, v_{st} \rangle \mid \text{either } s = k \text{ and } t = l + n1 \\ \text{or } s = k + m1 \text{ and } t = l \},$

and thus $\langle v_{kl}, v_{st} \rangle$ corresponds to the elementary tile with the left span $\overline{P_kQ_l}$ and the right span $\overline{P_sQ_t}$. G is a toroidal graph (see Figure 3). We refer to the vertex v_{ij} as being in the row i and in the column j ; similarly we say that the arc $\langle v_{ij}, v_{i+m1,j} \rangle$ is a *vertical* arc between

the rows i and $i+m1$, and the arc $\langle v_{ij}, v_{i,j+n1} \rangle$ is a *horizontal* arc between the columns j and $j+n1$.

Any set of elementary tiles can be viewed as a subgraph of G spanned by the arcs corresponding to the elementary tiles (Figure 4). For any subgraph S of G and a vertex v of G we denote by **indegree** $_S(v)$ and **outdegree** $_S(v)$ the number of arcs in S which are incident to v or from v , respectively. We next characterize those subgraphs of G , referred to in the sequel as *acceptable subgraphs*, which correspond to acceptable surfaces. Let S be a subgraph of G which corresponds to an acceptable surface. The previously stated conditions (1) and (2) are equivalent to the following conditions on S :

- (1') For every $i, i = 0, 1, \dots, m-1$, there is exactly one vertical arc in S between the rows i and $i+m1$ (which corresponds to an elementary tile which includes the contour segment $\overline{P_iP_{i+m1}}$); and for every $j, j = 0, 1, \dots, n-1$, there is exactly one horizontal arc in S between the columns j and $j+n1$ (which corresponds to an elementary tile which includes the contour segment $\overline{Q_jQ_{j+n1}}$).
- (2') For a vertex v_{ij} of G , either the **indegree** $_S(v_{ij}) = \text{outdegree}_S(v_{ij}) = 0$ or **indegree** $_S(v_{ij}) > 0$ and **outdegree** $_S(v_{ij}) > 0$.

We remind the reader that a directed graph is *weakly connected* if and only if it is connected when considered as an undirected graph, ignoring the direction on the arcs. A directed graph splits into one or more maximal weakly connected subgraphs, which are referred to as *weak components*.

LEMMA 1. *An acceptable subgraph of G is weakly connected.*

PROOF. Let S be an acceptable subgraph of G and assume by contradiction that S has at least two weak components. Consider one of its components S_0 . There is at least one arc of S which is not in S_0 , and assume, without loss of generality, that it is horizontal. We claim that there exist i and j such that the vertex v_{ij} is in S_0 , but S_0 does not contain a horizontal arc between the columns $j-n1$ and j . Indeed, let us consider the two possible cases:

- (1) S_0 has no horizontal arcs. As S_0 is not empty we choose an arbitrary vertex of S_0 to be v_{ij} (which is then incident with a vertical arc in S_0).

Fig. 3. Toroidal graph representation.

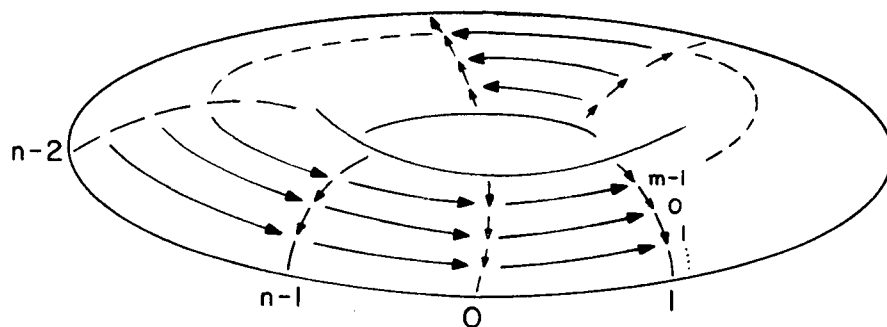
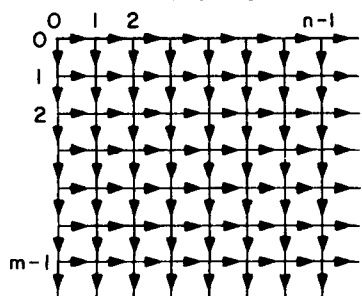
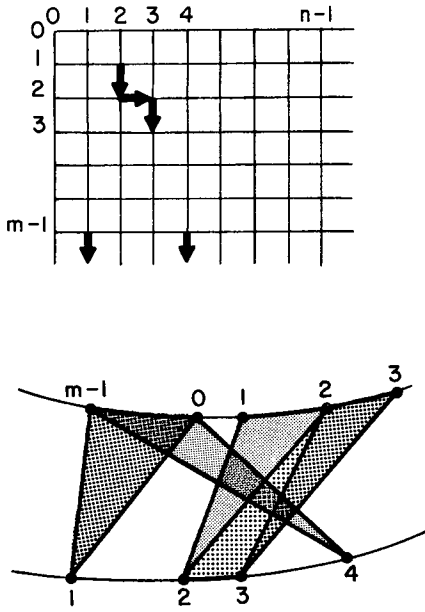


Fig. 4. Correspondence between a set of tiles and a subgraph.



(2) S_0 has at least one horizontal arc. As S contains exactly one horizontal arc between any two adjacent columns and as we have assumed there is a horizontal arc of S which is not in S_0 , it follows that for some j , S_0 has a horizontal arc between the columns j and $j+n1$ but has no horizontal arc between the columns $j-n1$ and j . Choose v_{ij} to be any vertex in the j th column which is in S_0 .

We now show that in both of these cases all the arcs $\langle v_{kj}, v_{k+m1,j} \rangle$, $k = 0, 1, \dots, m-1$, are in S_0 . As v_{ij} is in S , it follows that $\text{indegree}_S(v_{ij}) > 0$, and, as $\langle v_{i,j-n1}, v_{ij} \rangle$ is not in S_0 , $\langle v_{i-m1,j}, v_{ij} \rangle$ must be in S_0 (Figure 5). Therefore $v_{i-m1,j}$ is in S_0 , $\text{indegree}_S(v_{i-m1,j}) > 0$, and, as $\langle v_{i-m1,j-n1}, v_{i-m1,j} \rangle$ is not in S_0 , $\langle v_{i-m2,j}, v_{i-m1,j} \rangle$ must be in S_0 . Proceeding in this way, we see that for all k , $\langle v_{kj}, v_{k+m1,j} \rangle$ are in S_0 .

As S contains a horizontal arc between any two adjacent columns, it follows that for some k , $\langle v_{k,j-n1}, v_{kj} \rangle$ is in S (Figure 6). This arc shares a vertex with the arc $\langle v_{kj}, v_{k+m1,j} \rangle$, which is in S_0 , and thus $\langle v_{k,j-n1}, v_{kj} \rangle$ must be in S_0 . As we have assumed before that S_0 contains no arc between the columns $j-n1$ and j , it follows by contradiction that $S = S_0$, S is weakly connected. \square

LEMMA 2. *If v_{ij} is a vertex of S such that $\text{indegree}_S(v_{ij}) + \text{outdegree}_S(v_{ij}) \geq 3$, then $\text{indegree}_S(v_{ij}) = \text{outdegree}_S(v_{ij}) = 2$, and for every other vertex v_{st} of S $\text{indegree}_S(v_{st}) = \text{outdegree}_S(v_{st}) = 1$.*

PROOF. By the assumption of the lemma, v_{ij} is incident with at least three arcs of S . Without loss of generality, we may assume that it is incident with the arcs $\langle v_{i,j-n1}, v_{ij} \rangle$, $\langle v_{ij}, v_{i,j+n1} \rangle$, and $\langle v_{i-m1,j}, v_{ij} \rangle$. S does not contain any other horizontal arcs between the columns $j-n1$ and j and between the columns j and $j+n1$. Thus by reasoning similar to that used in the proof of Lemma 1, it can be shown that S contains all the arcs of the form $\langle v_{kj}, v_{k+m1,j} \rangle$ for $k = 0, 1, \dots, m$

Fig. 5.

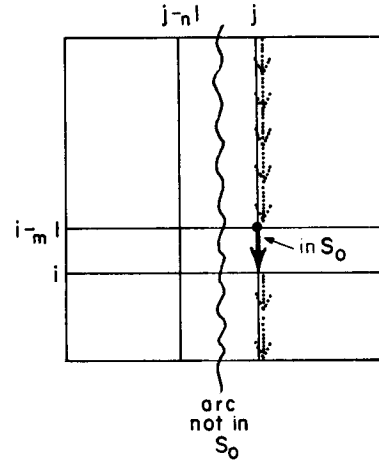


Fig. 6.

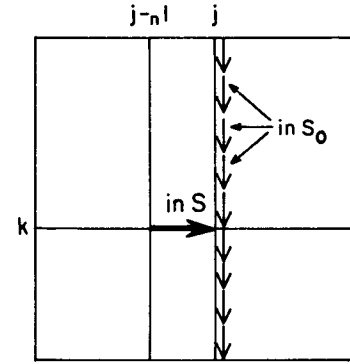
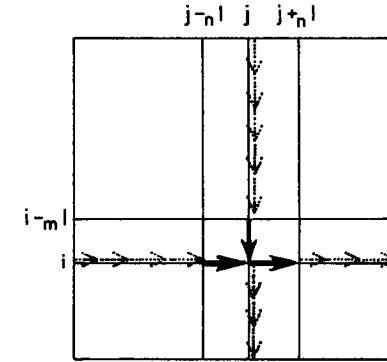


Fig. 7.



-1 (Figure 7). S contains the arcs $\langle v_{i-m1,j}, v_{ij} \rangle$ and $\langle v_{ij}, v_{i+m1,j} \rangle$ and therefore contains no other vertical arcs between the rows $i-m1$ and i and between the rows i and $i+m1$. Thus, as S is weakly connected and contains a horizontal arc between any two adjacent columns, it has to contain all the arcs of the form $\langle v_{i,k}, v_{i,k+n1} \rangle$ for $k = 0, 1, \dots, n-1$. Furthermore S contains no additional arcs, as we have already listed all its vertical and horizontal arcs.

We have already shown that v_{ij} is incident with the arcs $\langle v_{i,j-n1}, v_{ij} \rangle$, $\langle v_{ij}, v_{i,j+n1} \rangle$, $\langle v_{i-m1,j}, v_{ij} \rangle$, and $\langle v_{ij}, v_{i+m1,j} \rangle$ in S . Any other vertex of S is either v_{ik} or v_{kj} for some k . v_{ik} is incident only with $\langle v_{i,k-n1}, v_{ik} \rangle$ and $\langle v_{ik}, v_{i,k+n1} \rangle$ in S , and v_{kj} is incident only with $\langle v_{k-m1,j}, v_{kj} \rangle$ and $\langle v_{kj}, v_{k+m1,j} \rangle$ in S . \square

We remind the reader that a directed graph is *eulerian* if and only if it can be traversed by a closed walk in which every arc of the graph occurs exactly once. Such a walk is called an *eulerian trail*. It is known that a subgraph S of a directed graph is *eulerian* if and only if it is weakly connected and for every vertex v in S $\text{indegree}_S(v) = \text{outdegree}_S(v)$.

We now state a set of necessary and sufficient conditions satisfied by an acceptable subgraph.

THEOREM 1. *Let S be a subgraph of G . Then S corresponds to an acceptable surface if and only if (1) S contains exactly one horizontal arc between any two adjacent columns and exactly one vertical arc between any two adjacent rows and (2) S is eulerian.*

PROOF. Let S correspond to an acceptable surface. Part (1) is just a restatement of condition (1'). By condition (2') and by Lemma 2 for every vertex v of S , $\text{indegree}_S(v) = \text{outdegree}_S(v)$, and by Lemma 1, S is weakly connected. Thus S is eulerian. Assume now that S satisfies parts (1) and (2). Condition (1') follows from part (1), and condition (2') follows from part (2). Thus S corresponds to an acceptable surface. \square

REMARK. *If S is an acceptable subgraph, for some $i, i = 0, 1, \dots, m-1, v_{i0}$ is in S (and for some $j, j = 0, 1, \dots, n-1, v_{0j}$ is in S).*

PROOF. From part (1) of Theorem 1 we know that S must contain a horizontal arc of the form $\langle v_{i0}, v_{i1} \rangle$ (and a vertical arc of the form $\langle v_{0j}, v_{1j} \rangle$). Thus v_{i0} (and v_{0j}) must be in S . \square

An acceptable subgraph S can be described as a closed trail (u_1, u_2, \dots, u_r) , where the u_k are vertices of G such that $u_1 = u_r = v_{i0}$ for some i , and S contains exactly all the distinct arcs $\langle u_1, u_2 \rangle, \langle u_2, u_3 \rangle, \dots, \langle u_{r-1}, u_r \rangle$. Note that $r = m + n + 1$ and S contains $m + n$ arcs. Such a trail is obtained by simply writing out the eulerian trail describing S starting with an appropriate v_{i0} . In the sequel we refer to acceptable subgraphs as *acceptable trails*.

We note that S is in one of two forms: either $\text{indegree}_S(v_{ij}) = \text{outdegree}_S(v_{ij}) = 1$ for every v_{ij} of S , or $\text{indegree}_S(v_{st}) = \text{outdegree}_S(v_{st}) = 2$ for one vertex of S and $\text{indegree}_S(v_{ij}) = \text{outdegree}_S(v_{ij}) = 1$ for every other vertex of S . These two cases correspond to two forms of an acceptable surface: in the first case it is homeomorphic to a cylinder (Figure 8), and in the second it is homeomorphic to two cones "glued" along the span $\overline{P_i Q_j}$ (Figure 9). (In some applications one may consider the latter form inappropriate and accept only those surfaces which are homeomorphic to a cylinder.)

We note that the number of acceptable surfaces is an exponentially growing function of $\max(m, n)$. To see this, consider all surfaces which correspond to trails beginning with v_{00} . Each such trail can be described as $v_{00}, I_1, I_2, \dots, I_{m+n}$, where each I_k may be thought of as one of the two instructions: "down" or "right." There are exactly m "down" and exactly n "right" instructions in the sequence I_1, I_2, \dots, I_{m+n} ; these instructions can appear in all possible permuta-

tions. Thus there are $(m+n)!/(m!n!)$ possible "programs," each of them corresponding to a distinct acceptable surface which includes the span $\overline{P_0 Q_0}$. As $(m+n)!/(m!n!)$ is already an exponentially growing function of $\max(m, n)$, the total number of acceptable surfaces (most of which do not include the span $\overline{P_0 Q_0}$) is exponential.

Fig. 8. An acceptable surface homeomorphic to a cylinder.

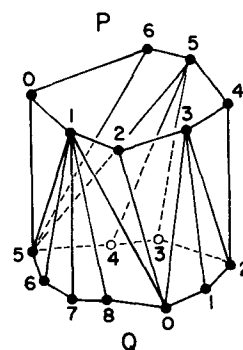
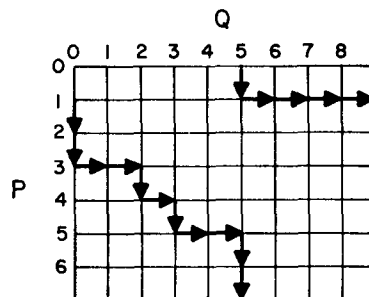
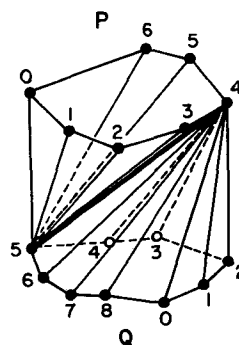
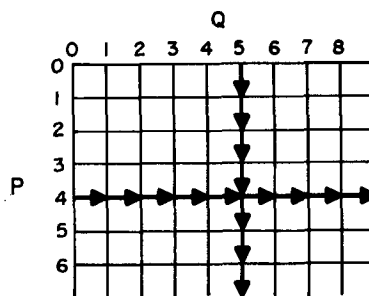


Fig. 9. An acceptable surface homeomorphic to two cones.



As there are many acceptable surfaces, we use additional criteria to choose optimal surfaces from among them. To this end we have to define the "quality" of an allowable surface, and then we look for a surface of "best" quality.

We describe this measure of "quality" by assigning costs to the tiles. We associate with each arc $\langle v_{kl}, v_{st} \rangle$ of G a cost $C(\langle v_{kl}, v_{st} \rangle)$ chosen from some set; in the sequel this will be the set of the real numbers. The cost of a trail is defined as usual to be the sum of the costs of the arcs traversed by it. A surface of best "quality" is one whose corresponding trail is of minimum cost. Using Theorem 1, we can now state our problem as follows: *Find a minimum cost acceptable trail.*

As we have stated, we can always write out the chosen trail so it starts and ends in a (single) vertex v_{i0} for some $i = 0, 1, \dots, m-1$. We stress here that we do not assume that a minimum cost acceptable trail includes any specified vertex fixed in advance, and thus we do not assume that a best approximating surface will include a specified span. By the Remark, the assumption that a best approximating surface "starts" with a span of the form $\overline{P_i Q_0}$ does not restrict the generality of the solution.

We reduce the problem of finding a minimum cost acceptable trail in our toroidal graph G to finding certain minimum cost paths in an appropriate planar graph G' (see Figure 10). (A *path* is a trail in which no vertex is repeated.) We define $G' = \langle V', A' \rangle$ by

$$V' = \{v_{ij} \mid i = 0, 1, \dots, 2m; j = 0, 1, \dots, n\},$$

$$A' = \{\langle v_{kl}, v_{st} \rangle \mid \text{either } s = k \text{ and } t = l + 1$$

$$\text{or } s = k + 1 \text{ and } t = l\}.$$

We also define for an arc $\langle v_{kl}, v_{st} \rangle$ of G'

$$C(\langle v_{kl}, v_{st} \rangle) = C(\langle v_{k(\bmod m), l(\bmod n)}, v_{s(\bmod m), t(\bmod n)} \rangle).$$

Thus G' is essentially constructed from G by cutting it open and gluing together two copies of the rectangle. Note that G' is acyclic (there are no closed nontrivial trails in G').

Let $i = 0, 1, \dots, m-1$. There exists a one-to-one correspondence between the set of the paths from v_{i0} to $v_{m+i,n}$ in G' and the set of the acceptable trails in G which start and end at v_{i0} . Indeed, such a correspondence is defined by associating with a path

$$v_{i0} = v_{i_1 j_1}, v_{i_2 j_2}, \dots, v_{i_{m+n+1} j_{m+n+1}} = v_{m+i,n}$$

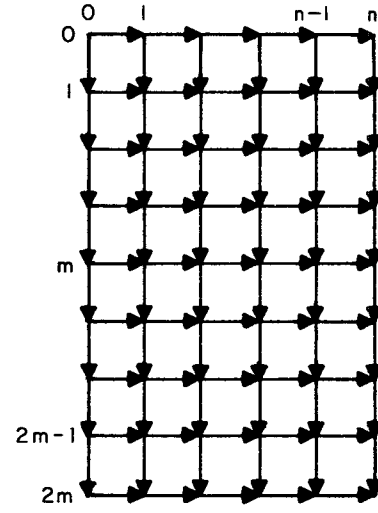
in G' the trail

$$v_{i0} = v_{i_1(\bmod m), j_1(\bmod n)}, v_{i_2(\bmod m), j_2(\bmod n)},$$

$$\dots, v_{i_{m+n+1}(\bmod m), j_{m+n+1}(\bmod n)} = v_{i0}$$

in G . We also note that this correspondence preserves the cost. Thus our problem can be succinctly stated as follows: *Find a minimum cost path π in G' from among all those paths which start at $v_{i,0}$ and end at $v_{m+i,n}$ for $i = 0, 1, \dots, m-1$. Define $\pi[i]$ for $i = 0,$*

Fig. 10. Planar graph G' obtained from toroidal graph G .



$1, \dots, m-1$ to be a minimum cost path from $v_{i,0}$ to $v_{m+i,n}$. Then the desired path π , corresponding to a best acceptable surface, can be found by first finding $\pi[0], \pi[1], \dots, \pi[m-1]$ and then picking a minimum cost path from among these m paths.

The problem of finding minimum cost paths in graphs has been studied extensively (for bibliography see [3] or [6]). Two main variants have been of interest:

- I. For a pair of vertices (u, w) in the graph, find a minimum cost path from u to w .
- II. For all pairs of vertices (u, w) in the graph, find a minimum cost path from u to w .

Well known methods exist for dealing with these variants.

If we find π by first finding m paths, $\pi[0], \pi[1], \dots, \pi[m-1]$, and then picking π from among them, then we are dealing with another possible variant, which falls between the two stated above:

- III. For several pairs of vertices (u, w) in the graph, find a minimum cost path from u to w .

Indeed, in our case we wish to find a minimum cost path for each of m pairs. Variant III can always be solved by solving m instances of variant I. (For our problem this is more advantageous than solving one instance of variant II and extracting the required information.) Thus if $T(m, n)$ is the number of operations required to find a single $\pi[i]$, all the required paths can be found in $mT(m, n)$ operations. In another paper a general method for more efficient solution of variant III is described [7]. In the sequel we present a possible implementation of that method for our problem. The implementation chosen is not the most elegant one, but rather one which lends itself to a simple presentation. As we shall see, variant III can then be solved in slightly more than $(\log_2 m)T(m, n)$ operations, resulting in a reduction of the number of opera-

tions to almost $(\log_2 m)/m$ of that required by the straightforward method of solving the problem by solving m instances of variant I.

We extend the problem by requiring that $\pi[m]$, a minimum cost path from $v_{m,0}$ to $v_{2m,n}$, be found also. (Since G' consists of two "copies" of G , $\pi[m]$ can be obtained by simply shifting a previously found $\pi[0]$ downwards.) It is simpler to describe the implementation by assuming that $\pi[0], \pi[1], \dots, \pi[m-1], \pi[m]$ are to be found.

THEOREM 2. *Let $\pi[i]$ be a minimum cost path from $v_{i,0}$ to $v_{m+i,n}$ for some $i \in \{0, 1, \dots, m\}$, and let $j \in \{0, 1, \dots, m\} - \{i\}$. Then there exists a minimum cost path from $v_{j,0}$ to $v_{m+j,n}$ which does not cross $\pi[i]$ (it may share vertices or arcs with $\pi[i]$).*

PROOF. Let $\pi[i] = (v_{i,0} = u_1, u_2, \dots, u_{m+n+1} = v_{m+i,n})$ and $\pi[j] = (v_{j,0} = w_1, w_2, \dots, w_{m+n+1} = v_{m+j,n})$ be minimum cost paths, and assume that they do cross. Without loss of generality assume that $i < j$. As $i < j$ and $m+i < m+j$, it follows that w_1 is below u_1 and w_{m+n+1} is below u_{m+n+1} . Let c be the smallest integer for which $\pi[i]$ and $\pi[j]$ cross at the vertex w_c , and thus $w_c = u_a$ for some a (see Figure 11). As w_1 is below u_1 , it follows that w_{c+1} is above $\pi[i]$. Since w_{m+n+1} is below u_{m+n+1} , $\pi[i]$ and $\pi[j]$ share some vertex w_d for $d > c+1$. Pick the minimal such d . Then, for some b , $b > a+1$, $u_b = w_d$, and $\pi_w \triangleq (w_c, w_{c+1}, \dots, w_d)$ lies above $\pi_u \triangleq (u_a, u_{a+1}, \dots, u_b)$. We show that π_w and π_u are of equal cost. Indeed if, say, π_w is of lower cost than π_u , then the path

$$\hat{\pi}[i] \triangleq (u_1, u_2, \dots, u_a = w_c, w_{c+1}, \dots, w_d = u_b, u_{b+1}, \dots, u_{m+n+1})$$

is a path from $v_{i,0}$ to $v_{m+i,n}$ which is of lower cost than $\pi[i]$, contradicting the assumption that $\pi[i]$ was a minimum cost path from $v_{i,0}$ to $v_{m+i,n}$. Assuming that π_u is of lower cost than π_w leads to a similar contradiction. Thus the costs of π_u and π_w must be equal. Define now the path

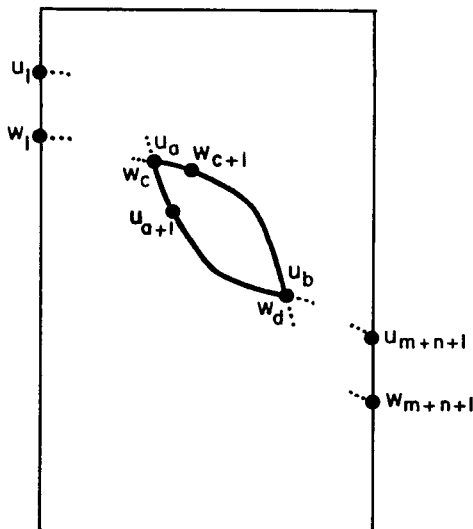
$$\hat{\pi}[j] \triangleq (w_1, w_2, \dots, w_{c-1}, w_c = u_a, u_{a+1}, \dots, u_b = w_d, w_{d+1}, \dots, w_{m+n+1}).$$

Then the cost of $\hat{\pi}[j]$ is equal to the cost of $\pi[j]$. Thus we have defined a minimum cost path $\hat{\pi}[j]$ from $v_{j,0}$ to $v_{m+j,n}$ which crosses $\pi[i]$ at fewer points than did $\pi[j]$. This process is repeated until a minimum cost path from $v_{j,0}$ to $v_{m+j,n}$ which does not cross $\pi[i]$ is obtained. \square

Let now $0 \leq i < j \leq m$ and let $\pi[i], \pi[j]$ be minimum cost paths which do not intersect. Let $V'(i, j)$ be the set of those vertices of G' which either belong to $\pi[i]$ or $\pi[j]$ or are between $\pi[i]$ and $\pi[j]$. Define $G'(i, j)$ to be the subgraph of G' spanned by $V'(i, j)$ (Figure 12).

COROLLARY. *Let $0 \leq i < j \leq m$ and let $\pi[i]$ and $\pi[j]$ be noncrossing minimum cost paths from $v_{i,0}$ to $v_{m+i,n}$ and from $v_{j,0}$ to $v_{m+j,n}$, respectively. Then for every k , $k = i+1, i+2, \dots, j-1$, there exists a minimum cost path $\pi[k]$ from $v_{k,0}$ to $v_{m+k,n}$ wholly contained in $G'(i, j)$.*

Fig. 11. Two paths of equal cost between $u_a = w_c$ and $u_b = w_d$.



PROOF. Indeed, $\pi[k]$ starts and ends below $\pi[i]$ and above $\pi[j]$. Thus by Theorem 2 it can be chosen so that it does not cross either $\pi[i]$ or $\pi[j]$ and thus is wholly contained in $G'(i, j)$. \square

Using Theorem 2, we can find all $\pi[0], \pi[1], \dots, \pi[m-1], \pi[m]$ by the following algorithm:

Algorithm ALLPATHS

SINGLEPATH(0, G');
SINGLEPATH(m , G');
PATHSBETWEEN(0, m);

where the procedure *PATHSBETWEEN*(i, j) is defined by

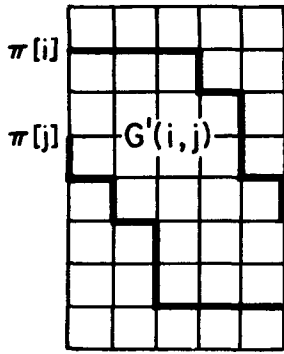
Procedure PATHSBETWEEN(i, j)
 $k := \lfloor (i + j)/2 \rfloor$;
if $i < k$ **then**
 begin
 SINGLEPATH(k , $G'(i, j)$);
 PATHSBETWEEN(i, k);
 PATHSBETWEEN(k, j);
 end

and the procedure *SINGLEPATH*(k, H'), not defined here, determines $\pi[k]$ by considering only the subgraph H' .

By the Corollary, a search for $\pi[k]$ can always be limited to $G'(i, j)$ whenever $i < k < j$ (and $\pi[i]$ and $\pi[j]$ have been previously found). We also note that for any invocation of *SINGLEPATH*($k, G'(i, j)$) in *PATHSBETWEEN*, the subgraph $G'(i, j)$ of G' is a subgraph of $G'(0, m)$.

We now describe an iterative version of this algorithm whose analysis is simpler to present. This version executes in $\lceil \log_2 m \rceil + 1$ stages. At the first stage $\pi[0]$ and $\pi[m]$ are found. At stage σ , $\sigma = 2, 3, \dots, \lceil \log_2 m \rceil$, the algorithm finds $2^{\sigma-2}$ paths, specifically $\pi[\lfloor (2k-1)m/2^{\sigma-1} \rfloor]$ is found in $G'(\lfloor (2k-2)m/2^{\sigma-1} \rfloor, \lfloor 2km/2^{\sigma-1} \rfloor)$ for $k = 1, 2, \dots, 2^{\sigma-2}$. At the stage $\lceil \log_2 m \rceil + 1$, the last stage, each of the paths of the form $\pi[\lfloor (2k-1)m/2^{\lceil \log_2 m \rceil} \rfloor]$ is found in $G'(\lfloor (2k-2)m/2^{\lceil \log_2 m \rceil} \rfloor, \lfloor 2km/2^{\lceil \log_2 m \rceil} \rfloor)$ for $k = 1, 2, \dots$

Fig. 12. Subgraph searched for paths $\pi(i + 1)$, $\pi(i + 2)$, \dots , $\pi(j - 1)$.



$2^{\lceil \log_2 m \rceil - 1}$, unless the path was found in the previous stages. One easily sees that $m - 2^{\lceil \log_2 m \rceil - 1}$ paths are found at this stage.

We derive now an upper bound on the number of operations required by this version of the algorithm to find all $\pi[0]$, $\pi[1]$, \dots , $\pi[m]$. For a subgraph H' of G' , we denote by $|H'|$ the number of arcs in H' . If u and w are two vertices in H' and if we know that there exists a minimum cost path (in G') from u to w which is wholly contained in H' , then the search for this path can be limited to H' , and the number of operations required to find such a path is a function of H' . An implementation of *SINGLEPATH*(k, H') using Dijkstra's algorithm [1] will require $O(|H'| \cdot \log_2 (|H'|/2))$ operations. Because of the uniform structure of G' it is possible to implement *SINGLEPATH*(k, H') so that the number of operations required is essentially equal to the number of arcs to be examined, and, since not every arc of H' has to be examined, the number of operations in an execution of *SINGLEPATH*(k, H') is bounded from above by $|H'|$.

We first observe that a search for $\pi[k]$, a minimum path from $v_{k,0}$ to $v_{m+k,n}$, can always be limited to those arcs of G' which are of the form $\langle v_{pq}, v_{st} \rangle$ for $k \leq p, s \leq m + k$, and there are $2mn + m + n$ such arcs. Thus the number of operations in the execution of *SINGLEPATH*($0, G'$) is bounded from above by $2mn + m + n$. Instead of actually executing *SINGLEPATH*(m, G'), $\pi[m]$ is obtained by "shifting" $\pi[0]$ downwards. Thus stage 1 requires $2mn + m + n$ operations. One easily sees that $|G'(0, m)| = 2mn + m + n + m = 2mn + 2m + n$.

Consider now a stage σ , $\sigma = 2, 3, \dots, \lceil \log_2 m \rceil$. The number of operations in this stage is bounded from above by the sum of the number of arcs in the subgraphs of $G'(0, m)$ considered in the stage. This bound is

$$\alpha = \sum_{k=1}^{2^{\sigma-2}} |G'(\lfloor (2k-2)m/2^{\sigma-1} \rfloor, \lfloor 2km/2^{\sigma-1} \rfloor)|.$$

To calculate this sum, we note that two subgraphs of the form

$$G'(\lfloor (2k-2)m/2^{\sigma-1} \rfloor, \lfloor 2km/2^{\sigma-1} \rfloor), \text{ and } G'(\lfloor 2km/2^{\sigma} \rfloor, \lfloor (2k+2)m/2^{\sigma} \rfloor),$$

which we call *consecutive*, both include the arcs in the path $\pi[2km/2^{\sigma-1}]$, and they do not have any other arcs in common. Thus every two consecutive subgraphs share $m + n$ arcs (the number of arcs in the path).

As there are $2^{\sigma-2} - 1$ pairs of consecutive graphs at this stage, σ , it follows that

$$\begin{aligned} \alpha &= |G'(0, m)| + (2^{\sigma-2} - 1)(m + n) \\ &= 2mn + 2m + n + (2^{\sigma-2} - 1)(m + n). \end{aligned}$$

Similarly, one can show that $|G'(0, m)| + (m - 2^{\lceil \log_2 m \rceil - 1} - 1)(m + n)$ is an upper bound on the number of operations at stage $\lceil \log_2 m \rceil + 1$, the last stage.

The upper bound $\tau(m, n)$ for the total number of operations will be:

$$\begin{aligned} \tau(m, n) &= 2mn + m + n + \sum_{\sigma=2}^{\lceil \log_2 m \rceil} [2mn + 2m + n \\ &\quad + (2^{\sigma-2} - 1)(m + n)] + 2mn + 2m + n \\ &\quad + (m - 2^{\lceil \log_2 m \rceil - 1} - 1)(m + n) \\ &= \lceil \log_2 m \rceil (2mn + m) + 3mn + m^2. \end{aligned}$$

Up to now we have not considered the relative magnitudes of m and n . To minimize $\tau(m, n)$, we assume that $m \leq n$. This assumption does not decrease the generality of the solution, since if $m > n$ one can simply interchange the roles of the two contours.

Thus we have

$$\tau(m, n) < (\lceil \log_2 m \rceil + 2)(2mn + m).$$

As $T(m, n)$, the number of operations required by *SINGLEPATH*(k, G'), is $2mn + m + n$, it follows that $\tau(m, n) < (\lceil \log_2 m \rceil + 2)T(m, n)$. Thus we have achieved a reduction in the number of operations required to solve our instance of variant III to less than $(\lceil \log_2 m \rceil + 2)/m$ of the number required by the straightforward method.

The iterative algorithm was executed for graphs of various sizes for which $m = n$. Ten graphs were chosen for each value of m , and the costs were uniformly distributed pseudo-random numbers between 0 and 1. In each graph the appropriate m paths were found. The number of arcs examined listed in Table I was the mean for the ten graphs for the value of m .

For the specific example described in the sequel, the performance did not vary significantly from the results summarized in Table I (e.g. for $m = 33, n = 52$, the number of arcs actually examined was 19,953 compared to 116,061 required by the straightforward method and the upper bound $\tau(33, 52) = 27,027$).

Example

We enclose an example of a surface constructed by our algorithm. For this application, the cost assigned to an arc was the area of the associated triangular surface tile. Thus the resulting overall surface is one with mini-

imum total surface area. Figure 13 illustrates the individual triangular tiles defined over the contours of Figure 1; Figure 14 contains continuous shaded images of the same surface from different orientations.

We note that the object definitions generated by this algorithm can be compressed, if desired, by combining those adjacent tiles which are most nearly coplanar.

Applications

We briefly sketch a few of the possible applications for this technique:

- Reconstruction of three-dimensional microscopic structures from two-dimensional images [10]
- Simulation of the likely results of reconstructive surgery from limited modifications of pre-surgery surface data
- Volume calculations of human body components from tomographs (reconstructed cross-sectional X-rays) for diagnosis and therapy [2]
- Construction of geographic terrain surfaces from topographic maps for realistic simulation of low-level flying in pilot training simulators
- Approximation of geographic surfaces from limited reconnaissance data
- Automatic construction of surfaces in interactive geometric design systems (e.g. realistic renderings of automobile bodies from early “skeleton” sketches).

Table I: Performance of New Minimum Cost Paths Algorithm.

Paths	Arcs	Old method	New method	New method upper bound	New/old
10	200	2,200	944	1,240	.429
20	800	16,800	4,622	5,700	.275
30	1,800	55,800	11,306	12,750	.203
40	3,200	131,200	21,700	25,840	.165
50	5,000	255,000	35,633	40,300	.140
60	7,200	439,200	52,819	57,960	.120
70	9,600	695,800	75,304	88,690	.108
80	12,800	1,036,800	98,802	115,760	.095
90	16,200	1,474,200	128,857	146,430	.087
100	20,000	2,020,000	162,406	180,700	.080
Paths	Number of minimum paths found: m .				
Arcs	Number of arcs in the graph G : $2mn$ ($m = n$).				
Old method	Number of arcs examined by straightforward method: $m(2mn + m + n)$.				
New method	Actual number of arcs examined by new method.				
New method upper bound	$\tau(m, n) = \lceil \log_2 m \rceil (2mn + m) + 3mn + m^2$.				
New/old	Actual ratio of arcs examined by new method to number of arcs examined by straightforward method.				

Fig. 13. Optimal tiled surface defined over the contours of Figure 1.

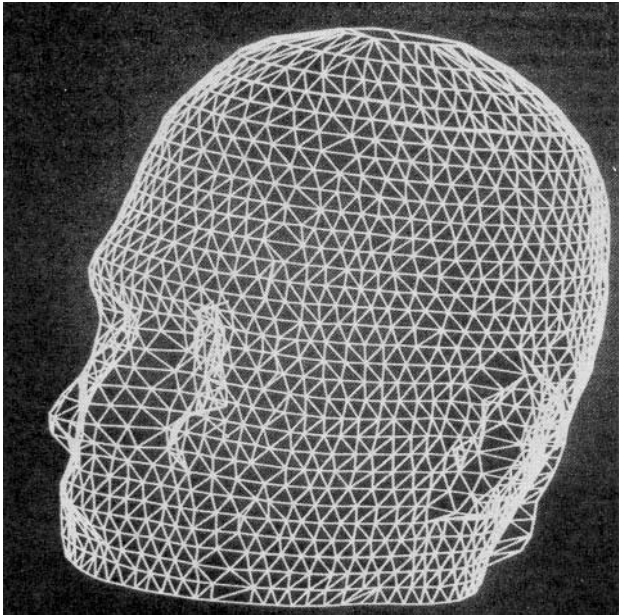
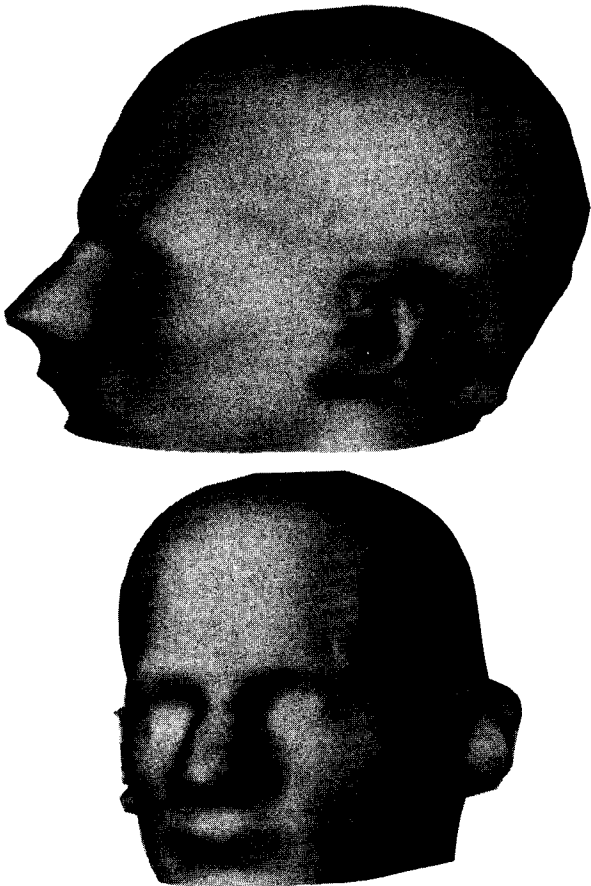


Fig. 14. Smooth shaded displays of surface of Figure 13.



Summary and Conclusions

We have shown that:

(1) The problem of defining a surface over contour points can be reduced to constructing a sequence of surfaces, one between each pair of adjacent contours.

(2) These surfaces can be constructed solely from elementary triangular tiles, each defined between two consecutive points on the same contour and a single point on an adjacent contour.

(3) All acceptable surfaces defined between two contours can be associated with certain cycles in a directed toroidal graph.

(4) Finding such cycles can be reduced to determining certain paths in an appropriate planar graph.

(5) Finding such paths can be performed efficiently by an algorithm which successively subdivides the graph to minimize the search space.

(6) The number of steps required to find a globally optimal path is bounded from above by $(\lceil \log_2 m \rceil (2mn + m) + 3mn + m^2)$, where m and n are the number of data points in each of the two contours.

One may wish to consider generalizations to this reconstruction algorithm. For example, one may be interested in applications in which the object contains multiple contours on a single plane. The reconstruction, for instance, of a standing human figure from contours on horizontal planes would involve perhaps three contours (two arms and the torso) at chest levels, merging into a single contour at the shoulder level. For these applications, the reconstruction algorithm would have to be extended to construct a single surface connecting more than two contours on a pair of adjacent planes.

Acknowledgments. We wish to thank R.E. Herron of Baylor University School of Medicine for providing the data for our example, Alain Fournier for his programming assistance, Michael Milochik for photographic assistance, the Medical Computing Resources Center at Dallas for the use of computing resources, and the Department of Computer Science of the University of Utah for the use of the graphic display equipment for generating Figures 13 and 14.

References

1. Aho, A.V., Hopcroft, J.E., and Ullman, J.O. *The Analysis and Design of Computer Algorithms*. Addison-Wesley, Reading, Mass., 1974.
2. Brooks, R.A., and DiChiro, G. Theory of image reconstruction in computed tomography. *Radiology* 117 (Dec. 1975), 561-572.
3. Christofides, N. *Graph Theory: An Algorithmic Approach*. Academic Press, New York, 1975.
4. Fuchs, H. The automatic sensing and analysis of three-dimensional surface points from visual scenes. UTECH-CSC-76-720, U. of Utah, Salt Lake City, Utah, Aug. 1975.
5. Harary, F. *Graph Theory*. Addison-Wesley, Reading, Mass., 1969.
6. Johnson, D.B. Efficient algorithms for shortest paths in sparse networks. *J. ACM* 24, 1 (Jan. 1977), 1-13.
7. Kedem, Z.M., and Fuchs, H. A fast method for finding several shortest paths in certain graphs. Submitted for publication.
8. Keppel, E. Approximating complex surfaces by triangulation of contour lines. *IBM J. Res. Develop.* 19 (Jan. 1975), 2-11.
9. Levinthal, C., and Ware, R. Three-dimensional reconstruction from serial sections. *Nature* 236 (March 1972), 207-210.
10. Shantz, M.J., and McGann, G. D. Computational morphology: three-dimensional computer graphics for electron microscopy. To appear in *IEEE Transactions on Biomedical Engineering*.
11. Weinstein, M., and Castleman, K.R. Reconstructing 3-D specimens from 2-D section images. *Proc. SPIE* 26 (May 1971), 131-138.