Surface Reconstruction of 3D Objects in Computerized Tomography

S. B. Xu AND W. X. Lu

Department of Scientific Instrument, Zhejiang University, Hangzhou, People's Republic of China

Received January 6, 1988; revised May 24, 1988

This paper deals with the problem of surface reconstruction of 3D objects from their boundaries in a family of slice images in computerized tomography (CT). Its mathematical formulation is first given, in which it is considered as a problem of functional minimization. Next, the corresponding Euler partial differential equation is derived and it is then solved by the finite difference method. Numerical solution can be found by using the iterative method.


1. INTRODUCTION

Since the advanced imaging technology of CT has been widely used in medicine, the demand of CT image processing has been increasing. In clinics, there is a need for displaying 3D images of human organs of interest. Usually, a group of slice images from the CT scanner are available. Our aim is to reconstruct the surfaces of the objects of interest from their boundaries that have been detected in those slice images. Therefore, the mathematical description of the surface reconstruction problem may be given: knowing a set of closed curves in N slice planes \( \{ B_k, k = 1, 2, \ldots, N \} \), where \( B_k = \{(x, y)|S(x, y) = z_k\} \), reconstruct the surface \( S(x, y) \).

Similar problems arise in other scientific and industrial fields. For instance, interpolation of one family of parametric curves such as \( \{ P(t, u_i), i = 1, 2, \ldots, n \} \) has been used in aircraft- and ship-building industries [2]. This process is called lofting, which in fact is the simplest linear interpolation. In CAD/CAM, surfaces are constructed piecewisely by methods of Coons' surface patch or Bezier surface control or interpolation of the tensor products of splines [1]. In these situations, surfaces are approximated by interpolation across the slices.

Some work has been done on the problem of surface reconstruction of 3D objects in CT images. One popular approach is the polygon tiling technique. Keppel [3] proposed a triangular tile technique for reconstructing a surface between contours. He associated the set of all possible triangular patch arrangements as a directed graph, then searched for an optimal triangular tile arrangement by finding the maximum cost path of the graph, which is equivalent to maximizing the volume enclosed by the tiled surface. Fuchs et al. [4] presented an algorithm for finding the minimum path of the graph which corresponds to a tile arrangement minimizing the surface area of the tiled surface.

Christiansen and Sederberg [5] proposed an algorithm that produces the triangular tiles by mapping the adjacent contours onto a unit square. Then the triangulation was selected on the basis of the shortest diagonal, with the requirement that the first selected points of each contour must approximate one another.

To produce a "nice-looking" surface, the upper and lower contours should be of similar shape and orientation. Based on this principle, Cook et al. [6] pursued a
method to reconstruct a surface from contours by connecting the sample points on two adjacent contours. The centroid of the region bounded by each contour is first calculated. The triangulation is then selected by joining the sample points from one contour with those from the other, with the condition that the orientation of the line segments connecting the sample points should be similar, as closely as possible, to the orientation of the line joining the centroids.

For diagnostic purposes the triangulation approach is not very suitable; only simple object surfaces can be produced automatically. Another drawback is that a tiled surface may not approximate the true surface if the distance between contours becomes larger.

Another approach proposed by Herman and Udupa et al. [7, 8] is the cuberille representation, in which an object is represented as a set of cubes (or voxels). The surface produced by this method is a set of faces of voxels. If necessary, linear interpolation is used to achieve inter-slice contours. Udupa [9] gave an efficient algorithm for finding the faces of voxels of the object. The voxel method does not have the shortcomings of the triangular approach. However, the niceness of the reconstructed surface depends heavily on the slice distance. Artifacts arise in the case of large slice distance.

In this paper, partially inspired by the work of Grimson [12] and of Terzopoulos [13], we think of the problem of surface reconstruction as that of functional minimization (i.e., the variational problem), which is solved by the partial differential equation (PDE) method; that is, the Euler equation corresponding to the variational problem is first derived, then this PDE is solved by the finite difference method. Finally a numerical solution is obtained. The following sections give the discussion in detail.

2. FORMULATION OF THE VARIATIONAL PROBLEM

As mentioned above, our purpose is to reconstruct the surface of a 3D object from its boundaries in a family of slice planes. Since there are infinite solutions of 2D surfaces passing through the given closed curves, it is impossible to determine the solution \( S(x, y) \) only based on the set of boundaries. Thus additional constraints should be imposed. The constraint we choose here is: the sum of the curvatures of the surface \( S(x, y) \) at all points should be a minimum. Intuitively, the constraint requires that the surface should be as smooth as possible. We may conclude that such a surface exists uniquely in the viewpoint of geometry.

The curvature of surface \( S(x, y) \) at a point \( (x, y, S(x, y)) \) is represented as \( S_{xx}^2 + 2S_{xy}^2 + S_{yy}^2 \), where \( S_{xx}, S_{xy}, S_{yy} \) are the second-order partial derivatives. Assuming \( S(x, y) \) has continuous 2nd-order partial derivatives, then we have the functional

\[
J[S(x, y)] = \iiint_D \left(S_{xx}^2 + 2S_{xy}^2 + S_{yy}^2\right) dx \, dy
\]

in which \( D \) is the orthogonal projection region of \( S(x, y) \) on the x-y plane, i.e., the domain of \( S(x, y) \). Now the surface reconstruction problem becomes one to find the minimum of the functional \( J[S(x, y)] \) in the region \( D \),

\[
J[S(x, y)] = \min_{(x, y) \in D} \left\{(x, y) \bigg| S(x, y) = z_k \right\} = B_k.
\]
According to the variational principle [10], the corresponding Euler equation is

$$\frac{\partial^2}{\partial x^2} F_t + \frac{\partial^2}{\partial x \partial y} F_s + \frac{\partial^2}{\partial y^2} F_t = 0,$$

where $F = S_{xx}^2 + 2S_{xy}^2 + S_{yy}^2$, $r = S_{xx}$, $s = S_{xy}$, $t = S_{yy}$. That is,

$$\frac{\partial^4 S}{\partial x^4} + 2\frac{\partial^4 S}{\partial x^2 \partial y^2} + \frac{\partial^4 S}{\partial y^4} = 0, \quad (x, y) \in D,$$

$$S(x, y) \bigg|_{(x, y) \in B_k} = z_k, \quad k = 1, 2, \ldots, N. \tag{3}$$

Here we assume the fourth-order partial derivatives of $S(x, y)$ exist.

We will solve this biharmonic PDE by using the finite difference method. The PDE is converted to a difference equation, which is solved by the iterative method to obtain a numerical solution.

3. ESTABLISHMENT OF DIFFERENCE EQUATION

Let us cover the $x$-$y$ plane by two families of straight lines parallel to the coordinate axes,

$$x_i = x_0 + i \cdot h, \quad y_j = y_0 + j \cdot h, \quad i, j = 0, \pm 1, \pm 2, \ldots$$

in which $(x_0, y_0)$ is an arbitrary point, $h$ is known as the step size. As such, the $x$-$y$ plane is segmented as a square lattice. $(x_i, y_j)$ is a lattice knot $(i, j)$. The region $D$ is discretized as $D_h = \{(x_i, y_j) \in D\}$. The original contour $B_k$ is digitized as $C_k$.

Substituting the difference operator for the differential operator and using the Taylor's series expansion near a point $(x, y)$, we get

\[
\frac{1}{h^4} \left[ (x + 2h, y) - 4S(x + h, y) + 6S(x, y) - 4S(x - h, y) + S(x - 2h, y) \right] \\
= \frac{\partial^4 S}{\partial x^4} + O(h^2)
\]

\[
\frac{1}{h^4} \left[ S(x + h, y + h) - 2S(x + h, y) + S(x + h, y - h) \\
- 2S(x, y + h) + 4S(x, y) \\
- 2S(x, y - h) + S(x - h, y + h) - 2S(x - h, y) + S(x - h, y - h) \right] \\
= \frac{\partial^4 S}{\partial x^2 \partial y^2} + O(h^2)
\]

\[
\frac{1}{h^4} \left[ S(x, y + 2h) - 4S(x, y + h) + 6S(x, y) - 4S(x, y - h) + S(x, y - 2h) \right] \\
= \frac{\partial^4 S}{\partial y^4} + O(h^2).
\tag{4}
\]
Neglecting the intercepting error $O(h^2)$ and substituting (4) for (3) results in the equation

$$\frac{1}{h^4} [S(x + 2h, y) + 2S(x + h, y + h) - 8S(x + h, y)$$
$$+ 2S(x + h, y - h) + S(x, y + 2h)$$
$$- 8S(x, y + h) + 20S(x, y)$$
$$- 8S(x, y - h) + S(x, y - 2h) + 2S(x - h, y + h)$$
$$- 8S(x - h, y) + 2S(x - h, y - h) + S(x - 2h, y)] = 0.$$  (5)

which reduces to

$$\frac{1}{h^4} [S_{i+2,j} + 2S_{i+1,j+1} - 8S_{i+1,j} + 2S_{i+1,j-1} + S_{i,j+2} - 8S_{i,j+1} + 20S_{i,j}$$
$$- 8S_{i,j-1} + S_{i,j-2} + 2S_{i-1,j+1} - 8S_{i-1,j} + 2S_{i-1,j-1} + S_{i-2,j}] = 0.$$  (6)

Let an operator $\Diamond_h$ represent the operation of the left-hand side in Eq. (6), then the difference equation is reduced to $\Diamond_h S_{i,j} = 0$, satisfying the boundary condition

$$S_{i,j} \big|_{(i,j) \in C} = z_k, \quad k = 1, 2, \ldots, N.$$  (7)

When a continuous contour $B$ is digitized, the boundary condition on the digitized contour $C$ need be, generally, transformed by the nearest point method or linear interpolation method. Fortunately, our closed curves that come from CT images are given in discrete form itself. Thus no transformation of boundary condition should be made, they may be directly used to solve the difference equation.

4. SOLVABILITY AND CONVERGENCE OF THE DIFFERENCE EQUATION

First we prove the existence of a solution of the difference equation (7). Changing variables, let

$$u_{i,j} = \frac{1}{h^2} [S_{i+1,j} - 2S_{i,j} + S_{i-1,j}] + \frac{1}{h^2} [S_{i,j+1} - 2S_{i,j} + S_{i,j-1}]$$
$$= \Delta_h S_{i,j},$$  (8)

where $\Delta_h$ is the Laplace difference operator. Then Eq. (7) becomes

$$\Delta_h u_{i,j} = 0$$  (9)

which is nothing more than the difference equation corresponding to the Laplace equation $\partial^2 u / \partial x^2 + \partial^2 u / \partial y^2 = 0$. It has been proved that solution of Eq. (9) exists uniquely [11].

If $U_{i,j}$ is the solution of Eq. (9), we have from the variable change formula (8),

$$\Diamond_h S_{i,j} = \Delta_h U_{i,j} = \Delta_h U_{i,j} = 0,$$

namely,

$$\Delta_h S_{i,j} = U_{i,j}.$$
which is nothing other than the difference equation corresponding to the Poisson equation $\frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2} = u(x, y)$: its solution exists uniquely has also been proved [11], hence the solution of Eq. (7).

Next we discuss the error convergence of solution of Eq. (7). From Eq. (4) we derive

$$\nabla_S(x, y) = \left( \frac{\partial^4 S}{\partial x^4} + 2 \frac{\partial^2 S}{\partial x^2 \partial y^2} + \frac{\partial^4 S}{\partial y^4} \right)_{i, j} + R_h$$

in which the estimated error $|R_h| \leq k \cdot M \cdot h^2$, $k$ is a constant, $M = \max_{i=0, \ldots, 6} |\partial^6 S / \partial x^6 \partial y^6 - i|$. Assuming $S(x, y)$ is the solution of the PDE (3), let $f_{i, j} = S(x_i, y_j) - S_{i, j}$, then $\nabla_h f_{i, j} = R_h$, and $|\nabla_h f_{i, j}| = |R_h| \leq k \cdot M \cdot h^2$.

In order to estimate the error of solution of the difference equation from the solution of the PDE, we represent the operator $\nabla_h$ in another form. Letting

$$g_{i, j} = \begin{bmatrix} 1 & 2 & -8 & 2 \\ -8 & 20 & -8 & 1 \\ 2 & -8 & 2 & 1 \end{bmatrix}$$

The difference operation may be represented as the convolution $\nabla_h S_{i, j} = g_{i, j} * * S_{i, j}$, where the notation $* *$ denotes the two-dimensional convolution. In terms of the theory of two-dimensional signal processing [14], it is easy to show the inverse operator $g_{i, j}^{-1}$ of operator $g_{i, j}$ exists. So

$$f_{i, j} = R_h * * g_{i, j}^{-1}$$

Thus the error estimate

$$|S(x_i, y_j) - S_{i, j}| = |f_{i, j}| = |R_h * * g_{i, j}^{-1}| \leq M_1 |R_h| \leq k \cdot M \cdot M_1 \cdot h^2,$$

where $M_1 = \max |g_{i, j}^{-1}|$. Equation (10) brings about $|S(x_i, y_j) - S_{i, j}| \to 0$ for $h \to 0$.

Based on the above analysis, we conclude that solution of Eq. (7) exists uniquely and converges to the true solution $S(x, y)$ of the PDE.

5. SOLUTION OF THE DIFFERENCE EQUATION

As a matter of fact, Eq. (7) or (6) corresponds to a simultaneous system of linear algebraic equations $A \cdot S = B$. If the number of the knots in the region is $P$, then $A$ is a $P \times P$ order square matrix, $S$ and $B$ are $P$-dimensional vectors. Thus solution of the PDE resorts to solving the linear equation system. Since the dimension $P$ is usually very large, and $A$ is a sparse matrix, it is appropriate to solve the system by an iterative method. The main advantage of an iterative method lies in computation easiness and storage saving. We discuss three kinds of iterative methods.

(1) **Simple iterative method.** First, an initial value of the solution $S^{(0)}_{i, j}$ is arbitrarily given; then approximating solutions $S^{(1)}_{i, j}, S^{(2)}_{i, j}, \ldots S^{(k)}_{i, j}$ are iteratively acquired.
by the equation
\[
S_{i,j}^{(k+1)} = \frac{1}{20} \left[ 8 \left( S_{i+1,j}^{(k)} + S_{i,j+1}^{(k)} + S_{i-1,j}^{(k)} + S_{i,j-1}^{(k)} \right) \\
- 2 \left( S_{i+1,j+1}^{(k)} + S_{i,j-1}^{(k)} + S_{i-1,j+1}^{(k)} \right) \\
\times S_{i-1,j-1}^{(k)} \right] - \left( S_{i+2,j}^{(k)} + S_{i,j+2}^{(k)} + S_{i-2,j}^{(k)} + S_{i,j-2}^{(k)} \right) 
\] (11)

When the knots on the boundaries come across in computing the right-hand side of Eq. (11), the corresponding boundary condition is applied. If the error of two successive iteratives \( |S_{i,j}^{(k+1)} - S_{i,j}^{(k)}| \leq \epsilon \) (the tolerable error given in advance), the iterative process stops. The lastest \( S_{i,j}^{(k)} \) is used as the approximating value of \( S(x, y) \) at point \((x_i, y_j)\). The features of the simple iterative method are: its iterative computation is independent of the order of knots and is thus appropriate for parallel processing, but it needs to store two successive groups of iterative values.

(2) **Gauss–Seidel iterative method.** Since in the simple iterative method, the next iterative is not performed until all knots have been computed once, the rate of iterative convergence is rather slow. One speed method is in the iterative process: if the \((k+1)\)th iterative value of some point near point \((i, j)\) currently being computed is known, it is at once applied in the computation of the point \((i, j)\). Therefore we have another iterative form,

\[
S_{i,j}^{(k+1)} = \frac{1}{20} \left[ 8 \left( S_{i+1,j}^{(k)} + S_{i,j+1}^{(k)} + S_{i-1,j}^{(k+1)} + S_{i,j-1}^{(k+1)} \right) \\
- 2 \left( S_{i+1,j+1}^{(k)} + S_{i,j-1}^{(k+1)} + S_{i-1,j+1}^{(k+1)} \right) \\
\times S_{i-1,j-1}^{(k+1)} \right] - \left( S_{i+2,j}^{(k+1)} + S_{i,j+2}^{(k+1)} + S_{i-2,j}^{(k+1)} + S_{i,j-2}^{(k+1)} \right) 
\] (12)

G–S iterative computation depends on the order of knots; for example, we may compute knots in order from left to right and from bottom to top. In addition, G–S iterative does not store the previous group of iterative values, thus reducing the storage requirement.

(3) **Successive over-relaxation (SOR) iterative method.** SOR iterative method is another accelerating convergence method. The weighted average of the \((k+1)\)th iterative value \( S_{i,j}^{(k+1)} \) of the G–S iterative method and the previous iterative value \( S_{i,j}^{(k)} \) is used as the current iterative value,

\[
S_{i,j}^{(k+1)} = \frac{1}{20} \left[ 8 \left( S_{i+1,j}^{(k)} + S_{i,j+1}^{(k)} + S_{i-1,j}^{(k+1)} + S_{i,j-1}^{(k+1)} \right) \\
- 2 \left( S_{i+1,j+1}^{(k)} + S_{i,j-1}^{(k+1)} + S_{i-1,j+1}^{(k+1)} \right) \\
\times S_{i-1,j-1}^{(k+1)} \right] - \left( S_{i+2,j}^{(k+1)} + S_{i,j+2}^{(k+1)} + S_{i-2,j}^{(k+1)} + S_{i,j-2}^{(k+1)} \right) ] + (1 - w)S_{i,j}^{(k)},
\] (13)

where \( 2 > w > 0 \) is a relaxation factor. If \( w = 1 \), the SOR iterative method reduce to the G–S iterative method.
6. RECONSTRUCTION ALGORITHM

From the foregoing discussion, we now present a surface reconstruction algorithm.

Step 1. Project the slice boundaries onto x-y plane, forming a family of contours $C_k$.

Step 2. Find the definition region $D_h$ of the surface $S(x, y)$, if $\bar{C}_k$ denotes the area enclosed by $C_k$, then usually

$$D_h = \bigcup_{k} \bar{C}_k \setminus \bigcap_{k} \bar{C}_k.$$  

Step 3. Set initial values of $S_{i,j}^{(0)} = 0$, for every internal knot $(i, j)$ in $D_h$, compute $S_{i,j}^{(k)}$ using the G–S (or SOR) iterative equation (12) or (13). For knots near the boundary, the coefficients in the iterative equation (12) should be modified. What follows is the modification from [13],

$$
\begin{array}{cccc}
1 & 1 & 1 & 1 \\
2 & -6 & 1 & -8 & -8 & 1 \\
1 & -6 & 1 & 2 & -6 & 2 \\
\end{array}
$$

where the circled figure stands for the current iterative point, the line segment for the boundary.

Step 4. Compare two successive iterative values $S_{i,j}^{(k)}$ and $S_{i,j}^{(k-1)}$; if their absolute difference $e$ is smaller than a constant $\epsilon$, stop the iteration. Otherwise go to Step 3 for continuing iteration. The successive iterative error $e$ may be taken as

$$e = \max_{(i, j) \in D_h} |S_{i,j}^{(k)} - S_{i,j}^{(k-1)}|,$$

or

$$e' = \frac{1}{P} \sum_{(i, j) \in D_h} |S_{i,j}^{(k)} - S_{i,j}^{(k-1)}|.$$  

(14')

Step 5. Represent the reconstruction surface $S(x, y)$ in the contour form for later processing.

The tolerable error in the algorithm is selected as follows: since we are interested in the coordinate values of $S(x, y)$, the tolerable error $\epsilon$ can be taken as $\epsilon = a/10$, where $a$ is the required coordinate accuracy of $S(x, y)$.

7. DISCUSSION

(1) Other constraints may be imposed on the surface reconstruction problem, which results in different PDEs, therefore, different iterative equations (the finite difference method is still used). For example, another constraint is that the area of the reconstructed surface be minimum. That is,

$$J_1[S(x, y)] = \iint_D \left(1 + S_x^2 + S_y^2\right) dx dy.$$
This constraint brings about the Laplace equation
\[ S_{xx}^2 + S_{yy}^2 = 0, \]
whose solution is a minimal rotation-like surface, which we think is not very appropriate to medical organ surface reconstruction for diagnostic purposes.

(2) Another approach to solve the variational problem (2) is the direct method, that is, using finite element method to get approximating solution of \( S(x, y) \). The finite element method avoids PDE; it also resorts to solving a system of linear algebraic equations. But the composition of its total stiffness matrix is arduous.

(3) At last we briefly explain the convergence rate of an iterative method. It is known that linear iteration \( S^{(k+1)} = G \cdot S^{(k)} + f \) converges if and only if \( \lim_{k \to \infty} G^k = 0 \), or \( \rho(G) < 1 \), where \( G \) is the iterative matrix, \( \rho(G) = \max \lambda \) is the spectral radius of \( G \), \( \lambda \) is the eigenvalue of \( G \). Let the iterative error \( e^{(k)} = S - S^{(k)} \), then \( e^{(k)} = G \cdot e^{(k-1)} = G \cdot e^{(0)} \). If \( G \) is symmetric, then \( ||G|| = \rho(G) \). It follows \( ||e^{(k)}|| = ||G^k \cdot e^{(0)}|| \leq ||G^k|| \cdot ||e^{(0)}|| = \rho^k(G) \cdot ||e^{(0)}|| \). If the error contraction factor is \( a < 1 \), then we have \( ||e^{(k)}||/||e^{(0)}|| = \rho^k(G) < a \). Thus the required iterative number is \( k \geq -\ln \alpha/\ln \rho(G) \), where the denominator \(-\ln \rho(G)\) is known as the convergence rate of the iterative method.

For the iterative algorithm of Eq. (7), the iterative matrix is

\[
G = \begin{bmatrix}
C & D & E \\
D & C & D & E \\
E & D & C & D & E \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \cdots & \cdots & C & D \\
0 & \cdots & \cdots & E & D & C \\
\end{bmatrix},
\]

where

\[
C = \begin{bmatrix}
1 & -8 & 2 \\
-8 & 20 & -8 & 2 \\
1 & -8 & 20 & -8 & 1 \\
0 & \cdots & \cdots & 1 \\
0 & \cdots & \cdots & -8 \\
1 & -8 & 20 & \end{bmatrix},
\]

\[
D = \begin{bmatrix}
2 & -8 & 2 & 0 \\
2 & -8 & 2 & \vdots \\
0 & \cdots & \cdots & 2 \\
0 & \cdots & \cdots & 2 \\
1 & 1 & 0 & \end{bmatrix},
\]

and

\[
E = \begin{bmatrix}
1 \\
0 & 1 \\
\end{bmatrix}.
\]
We can decompose $G$ as $G = A + U + L$, where $A$ is a diagonal matrix, $U$ and $L$ are upper and lower triangular matrix respectively, and $U = L^T$. The iterative matrices of the above three iterative methods are $G_1 = -A^{-1}(U + L)$, $G_2 = -(A + L)^{-1}U$, and $G_3 = (A + wL)^{-1} + [(1 - w)A - wU]$, respectively. Since it is very difficult to calculate the eigenvalues of the above matrices, the convergence rate of iteration of Eq. (7) is suspending.

8. CONCLUSION

We have proposed a new approach for reconstructing 3D object surfaces from their CT slice images. Starting from the variational principle, the corresponding PDE is derived and solved by the finite difference method, which can result in numerical solution of the surface $S(x, y)$. We have also proved the existence and convergence of the difference equation, discussed three iterative methods to solve the difference equation, and presented an iterative reconstruction algorithm. Future work will give experimental results.

An important feature of the proposed method lies in the fact that there are no rigorous requirements on the slice distance; it can handle cases of relatively large slice distance and non-equidistant slices. It may also be used in other fields coping with surface reconstruction, such as machine vision, geographic mapping, etc.

ACKNOWLEDGMENT

This research is in part supported by the grant from the National Education Committee of P. R. China.

REFERENCES