Surface Interpolation from Sparse Cross Sections Using Region Correspondence

G. M. Treece*, R. W. Prager, A. H. Gee, and L. Berman

***Abstract*—The ability to estimate a surface from a set of cross sections allows calculation of the enclosed volume and the display of the surface in three-dimensions. This process has increasingly been used to derive useful information from medical data. However, extracting the cross sections (segmenting) can be very difficult, and automatic segmentation methods are not sufficiently robust to handle all situations. Hence, it is an advantage if the surface reconstruction algorithm can work effectively on a small number of cross sections. In addition, cross sections of medical data are often quite complex. *Shape-based interpolation* is a simple and elegant solution to this problem, although it has known limitations when handling complex shapes. In this paper, the shape-based interpolation paradigm is extended to interpolate a surface through sparse, complex cross sections, providing a significant improvement over our previously published *maximal disc-guided interpolation*. The performance of this algorithm is demonstrated on various types of medical data (X-ray computed tomography, magnetic resonance imaging and three-dimensional ultrasound). Although the correspondence problem in general remains unsolved, it is demonstrated that correct surfaces can be estimated from a limited amount of real data, through the use of *region* rather than *object* correspondence.**

***Index Terms*—Object correspondence, shape-based interpolation, surface from cross sections.**

I. Introduction

THREE-DIMENSIONAL (3-D) data is now available in many medical imaging modalities. One of the ways of displaying such data is by rendering a surface within it. In order to do this, the position of the surface has to be determined by some means. This is often achieved by segmenting many planes through the data, resulting in a set of cross sections from which the surface can be estimated. If the data is sufficiently dense, and the segmentation is performed over all of it, a surface can be rendered directly from these cross sections.

In practice, there are many situations where this is not the case. In X-ray computed tomography (CT) data, for instance, the resolution of the pixels in each plane is typically greater than the spacing of these planes, and it is desirable to interpolate new data between them. The simplest interpolation technique is linear, using the original grey scale values, however, this can generate artifacts in subsequent surface renderings [1]. Hence, object-based interpolation methods are often used, which interpolate the object cross section rather than the grey scale values.

Segmentation of medical data is nontrivial, and although there are many techniques which exist for automatically segmenting specific data, they are not sufficiently robust for generic use. Consequently, manual segmentation, or computer-assisted manual segmentation [2], [3] is often necessary. In typical applications, the resolution of the data is $256\times 256 \times 256$ or so—which implies manual segmentation is required in several hundred planes. Hence, even where the data is sufficiently dense that surface interpolation is not strictly necessary, it can be helpful in reducing the number of planes which have to be segmented.

In freehand 3-D ultrasound, the original data is acquired on planes which are not parallel. Although it is possible to interpolate this data to a uniform array and segment this array, it is easier to segment the original images (B-scans). This leads to cross sections which are themselves not parallel. In addition, ultrasound B-scans are particularly difficult to segment automatically, due to the various artifacts present in these images. It is, therefore, important to have a robust surface interpolation algorithm which can handle nonparallel cross sections. It is also important for this algorithm to be fast, so that the surface estimation can be used iteratively to refine the manual segmentation.

II. Related Work

A. Surface from Cross Sections

There are two distinct approaches for constructing a surface from cross sections of an object. Both approaches can be used to generate a triangular mesh of the surface, which is useful for rendering using standard graphics hardware. The first approach directly triangulates the set of points making up each of these cross sections, such that they become the vertices of the triangular mesh. The alternative is to use the cross sections to estimate a 3-D function which represents some measure of distance from any point to the surface. Once this function has been created, the zero iso-surface can be triangulated to reveal the object surface. This approach was originally suggested by Levin for the interpolation of CT data [4].

Direct triangulation of points on the cross sections is a difficult (though well studied) problem in cases where the cross-sectional shape varies between planes. Additional vertices must be created if the topology changes, for instance a saddle point for

Manuscript received May 17, 1999; revised August 11, 2000. The work of G. Treece was supported by an EPSRC studentship and by the University of Cambridge Department of Engineering under a Newton Trust award. The Associate Editor responsible for coordinating the review of this paper and recommending its publication was S. Pizer. *Asterisk indicates corresponding author.*

*G. M. Treece is with the Department of Engineering, University of Cambridge, Trumpington Street, CB2 1PZ Cambridge, U.K., (e-mail: gmt11@eng.cam.ac.uk).

R. W. Prager is with the Department of Engineering, University of Cambridge, CB2 1PZ Cambridge, U.K.

A. H. Gee is with the Department of Engineering, University of Cambridge, CB2 1PZ Cambridge, U.K.

L. Berman is with the Department of Radiology, University of Cambridge, Addenbrooke's Hospital, CB2 2QQ Cambridge, U.K.

Publisher Item Identifier S 0278-0062(00)09745-7.

a branching structure. It is also impossible to constrain the aspect ratios of the generated triangles, since the vertices are defined by the positions of the cross sections. This can lead to poor quality surface displays if Gouraud shading (which is the fastest effective shading technique) is used. Much effort is required to detect and correct special cases where the triangulation of complex shapes might otherwise fail [5]–[8]. In addition, there have been very few examples in the literature of triangulation of nonparallel cross sections [9].

In contrast, the functional approach applies the same criteria to simple and complex shapes. Once the function has been created, the iso-surface can be triangulated simply and quickly by a variety of algorithms, marching cubes [10] being the most widely used, although other algorithms can give better quality meshes [11]. In some cases, extracting the iso-surface at some nonzero value (i.e., at a constant distance from the actual surface) can also provide useful information [12].

Shape-based interpolation [1], [13] is probably the simplest form of distance function creation. In this method, binary images defining the object cross section are first transformed, such that each pixel in the image is replaced by the minimum distance to the cross section. The distance transform can be estimated most efficiently by an algorithm introduced by Borgefors [14]. Pixels inside the object are assigned positive distances, and those outside negative. These two-dimensional (2-D) distance functions can then be interpolated using linear interpolation, to give a C_0 continuous 3-D distance function, where the zero iso-surface represents the estimated surface.

Shape-based interpolation gives good results for cross sections which do not differ significantly between slices, but performs poorly when the cross sections do not have significant overlap, or have very different shapes. This is demonstrated in Fig. 1, which shows some (near) parallel cross sections from a freehand 3-D ultrasound scan of a part of the hepatic system. Fig. 1(a) and (b) shows the actual anatomy, whereas Fig. 1(d) shows the result of applying shape-based interpolation to every other cross section. Since this is a linear interpolation, normal to the scan planes, the surface can only exist within the projection of the original cross sections—where these do not overlap, the cross sections are not connected.

This problem has been addressed to some extent by moving the cross sections to align the centroids of each object *before* interpolation (then using the inverse transform on the interpolated cross sections) [15], as in Fig. 1(e). This works well for tree-like shapes which have been scanned transverse to the main axis, like the top three cross sections in Fig. 1(e). However, disastrous results can ensue when the shape of each cross section changes dramatically, even if the original shape is still a tree-like structure, as can be seen from the other cross sections in the figure. Scaling each cross section such that the bounding rectangles are the same size, in addition to aligning the object centroids, can help in some cases [16]—but makes the situation worse in the case of Fig. 1(f). An implementationally more complicated, but practically very similar approach based on mathematical morphology, which aligns objects on each cross section using an iterative minimization of distance transform values, is presented in [17].

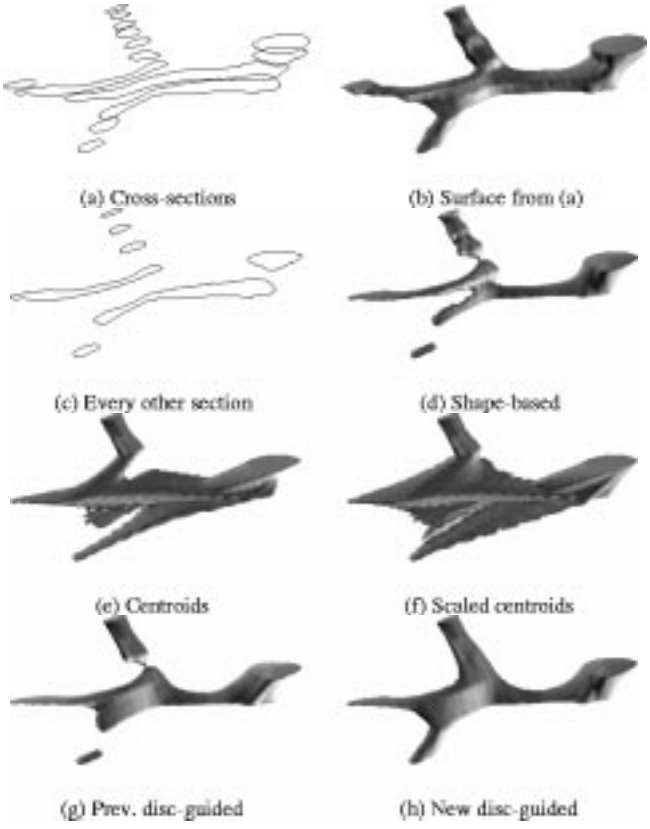

Fig. 1. Part of the hepatic system. The cross sections were manually outlined on the original images (B-scans) of a freehand 3-D ultrasound investigation. (a) and (b) shows the original cross sections, (d)-(h) shows the surface interpolated from the selected cross sections shown in (c).

Where there is some definition of connectivity in all of the above algorithms, it is based on *whole contours*, but it can be seen from Fig. 1(e) and (f) that this is not always appropriate: the small contours only correspond to a small part of the longer contours. An alternative object-based interpolation approach, dynamic elastic interpolation, begins to address this problem [18]. Here, a force field is constructed which iteratively deforms one cross section toward the other. The initial formulation produced similar results to centroid-guided shape-based interpolation for complex objects. However, this behavior was improved by imposing an upper threshold on the forces at each point on the contour. This prevents far portions of the cross section having a detrimental effect on the deformation, but also adds further complexity, and an additional parameter to be estimated.

We have previously presented an algorithm which guides the interpolation of a distance transform by using correspondence of *regions* of cross sections [19]. This region correspondence is determined by representing each cross section by a set of discs which are contained by the object. The result of this algorithm, maximal disc-guided interpolation, is shown in Fig. 1(g). This is clearly an improvement on the previous shape-based techniques, however, it is still not a faithful representation of the actual anatomy [Fig. 1(b)]. In addition, in common with the centroid based techniques, this technique can introduce artifacts into complicated surfaces which would not be introduced by shape-based interpolation alone. In contrast, the extension

of this algorithm presented in this report gives the surface of Fig. 1(h) which in this case correctly represents the actual anatomy.

It is important to recognize that no technique can guarantee to reconstruct the actual anatomy from *any* set of cross sections. It will always be possible to create a set of cross sections whose connectedness will be misjudged by a given algorithm. This is simply a consequence of under-sampling: if the inter-slice spacing is greater than the shortest wavelength present in the data, fine detail will inevitably be missed in the surface reconstruction. However, it is fair to expect a reconstruction algorithm to produce a "reasonable" surface, in the sense that similar contours should be connected and the surface should have the simplest topology given the cross sections (see [20] for further discussion of this issue)—this is the goal of the algorithm presented in this paper.

B. Surface from Scattered Points

Algorithms which can create surfaces from scattered point sets can also be applied to more regular planar cross sections, by using the vertices of each cross section—although in general this increases the complexity of the problem. There are several such algorithms which generate a 3-D distance function from these points [21]–[23]. These represent very general solutions, but as a result are not always well constrained, and many points are required to ensure a correct reconstruction.

In freehand 3-D ultrasound, the cross sections are already nonparallel and, therefore, this approach becomes more attractive: indeed it has recently been implemented for reconstructing the left ventricle of porcine hearts in a water bath [24]. However, it was noted that the ultrasound data was not sufficiently dense to constrain the surface, and the method had to be adapted to include a prior model of the surface, which was then deformed to represent the data, rather than constructed from the data.

To allow interpolation of more sparse data, the problem can also be restated in terms of minimizing the curvature of the resulting surface. This can be achieved by using volume splines to reconstruct the surface from cross sections. In one method [25], each cross section is first represented by a 2-D carrier function, whose values are calculated at the vertices of the cross section. Then a spline function of two variables is used to approximate the function, which can then be linearly interpolated between each cross section, once again giving a 3-D distance function which is then thresholded at zero. Alternatively, a volume spline can be fitted to all the cross sections simultaneously, generating the 3-D distance function directly [26]. Since these methods involve the inversion of an $N \times N$ matrix, where N is the number of points on each cross section for the former method, or on all cross sections for the latter, the processing time increases significantly with data complexity.

C. Morphing

The idea of constructing a surface between two differing cross sections is very closely linked with that of "morphing" one 2-D shape to another, much used in computer graphics [27]. In fact, distance field interpolation has recently been used for this purpose [28]. Here, distance transformation is combined with warping of the original binary images to give a smoother transition (the technique is also applied to morphing 3-D data). There is, however, a difference between the relative philosophies in computer graphics and medical imaging, in that user-definition of the way in which the image is transformed is considered an advantage in computer graphics, whereas it is desired that the cross sections should be created automatically from medical data. In computer graphics, the task is to create a new surface, whereas the clinician wants to visualise a pre-existing surface. Nevertheless, the idea of warping a distance transform is close to that of interpolating it based on region correspondence, since the warping function is itself an estimate of local correspondence.

Distance transformation has also been used in conjunction with field-morphing in order to interpolate the intensity information in medical images [29]. Here, shape-based interpolation is used to determine intermediate cross sections which form the basis of the control points required by the field-morphing process.

D. Original Contribution

We present a major extension of *maximal disc-guided interpolation*, first presented in [19]. The algorithm has been improved in several important areas.

- The traditional "branching" and "correspondence" problems are combined by determining "region correspondence." This is achieved by representing the cross sections as a set of maximal discs, derived from the distance transformation of the cross section. The distance transformation is also used to determine the correspondence between discs on neighboring cross sections. This dramatically improves the estimated surface of complex objects.
- External discs are used in addition to internal ones. This means that holes in the object, or external concave features, can also be used to determine correspondence.
- An improved combination of the disc correspondences gives a better point correspondence, resulting in better treatment of cross sections of multiple objects.

In addition, to the authors' knowledge, there has previously been no attempt to compare the surfaces generated by the derivatives of shape-based interpolation across a broad range of data. This paper contains comparisons of surfaces from 3-D ultrasound, CT, and MRI data.

III. Maximal Disc Guided Interpolation

A. Overall Strategy

The overall strategy of the surface interpolation algorithm is displayed in Fig. 2. This differs from shape-based interpolation in that the interpolation direction is allowed to vary at each point dependent on the region correspondence. Since shape-based interpolation already requires that a distance transform has been calculated for each cross section, it is straightforward to extract a set of maximal discs from these transforms, which loosely represent the shape of each object. Correspondence is then calculated for these discs, assisted by the original distance transforms. The interpolation direction at any point is then calculated as a weighted sum of contributions from each disc, where

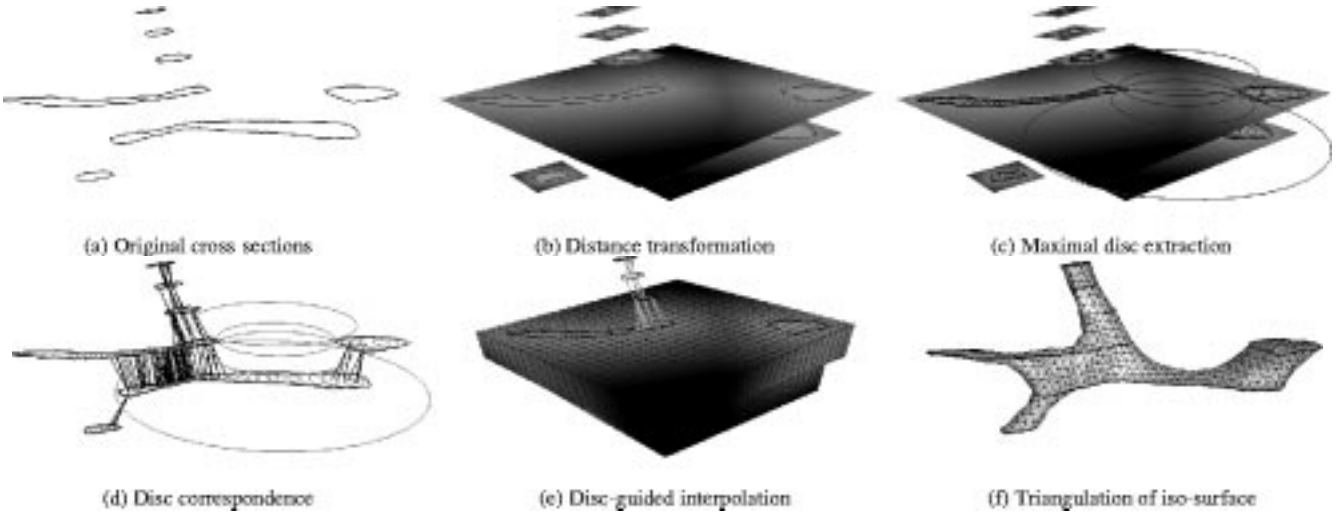

Fig. 2. Surface interpolation algorithm. (a) Original cross sections. (b) Distance transformation is first performed on each cross section, then (c) a set of representative discs can be extracted from these transformations. (d) Region correspondence is estimated for each disc, and this is combined to give a correspondence direction at any point. (e) This provides the interpolation direction for the distance field interpolation, which can then be triangulated (f) to reveal the object surface.

the weighting is again derived from the original distance transforms. Thus, the distance function can be found at any point in space between neighboring cross sections and evaluated on a grid suitable for zero iso-surface triangulation.

The use of the distance transform to weight both the correspondence and interpolation direction calculations makes up for the loose representation of the object by only a few discs. Centers of discs are a useful representation to guide the shape-based interpolation, since they are independent of the disc radii. Shape-based interpolation is itself already very good at handling changes of scale in similar shapes.

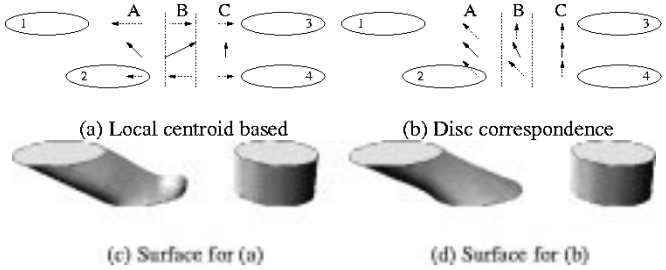

Fig. 3. Calculation of interpolation direction. Discs 1 and 3 are from one cross section, and 2 and 4 from another. The regions A, B and C are those nearest to discs 1 and 2, 2 and 3, and 3 and 4, respectively. Dashed arrows show the information from each of the top and bottom planes, bold arrows show the interpolation direction.

B. Processing Steps

1) Segmentation: Binary segmentation of each cross section, by any 2-D method, must be performed prior to surface interpolation. This information is required in the form of a binary array, but can easily be converted from sets of ordered contour points, like those shown in Fig. 2(a).

2) Distance Field Transformation: An efficient algorithm to calculate the distance transformation for a rectangular area is given in [14]. The distance transform is calculated for each cross section, as in Fig. 2(b). In addition, a far point distance algorithm described in [19] is used to provide distance values outside this rectangular region.

3) Maximal Disc Representation: Extraction of a set of maximal discs is also described in [19]. The disc centers are a subset of the peaks and ridges of the distance transform, the radius in each case being determined by the value of the distance transform at the center. In this case, we also want to include external discs (those outside, rather than contained by, the contours), so the troughs and valleys are also considered, using the same criteria as for the internal discs. This allows correspondence of holes, concavities and gaps between objects, in addition to the objects themselves. The region over which external discs are gathered is limited by the size of the distance transformed area[1]. The set of internal and external discs for the cross sections in Fig. 2(a) is shown in Fig. 2(c).

4) Disc Correspondence: Fig. 3 shows the calculation of interpolation direction for the previous and new algorithms. In Fig. 3(a) and Fig. 3(c), the discs were used to calculate a "local centroid" vector in each plane, which indicated the direction to the nearest object contour. Local centroids from each neighboring plane were combined at each point to give the interpolation direction. This results in correspondence between discs 2 and 3 in region B, potentially generating a "spur" in that direction on the surface, which does not seem appropriate from the cross sections.

The new algorithm calculates the correspondence direction for each disc first, then interpolates this, rather than the local centroid, at each point. This results in a more consistent correspondence (and, hence, interpolation direction) across the plane, as shown in Fig. 3(b) and (d).

[1]In reality, there are situations where external discs exist outside this area, but this has little practical effect on the results.

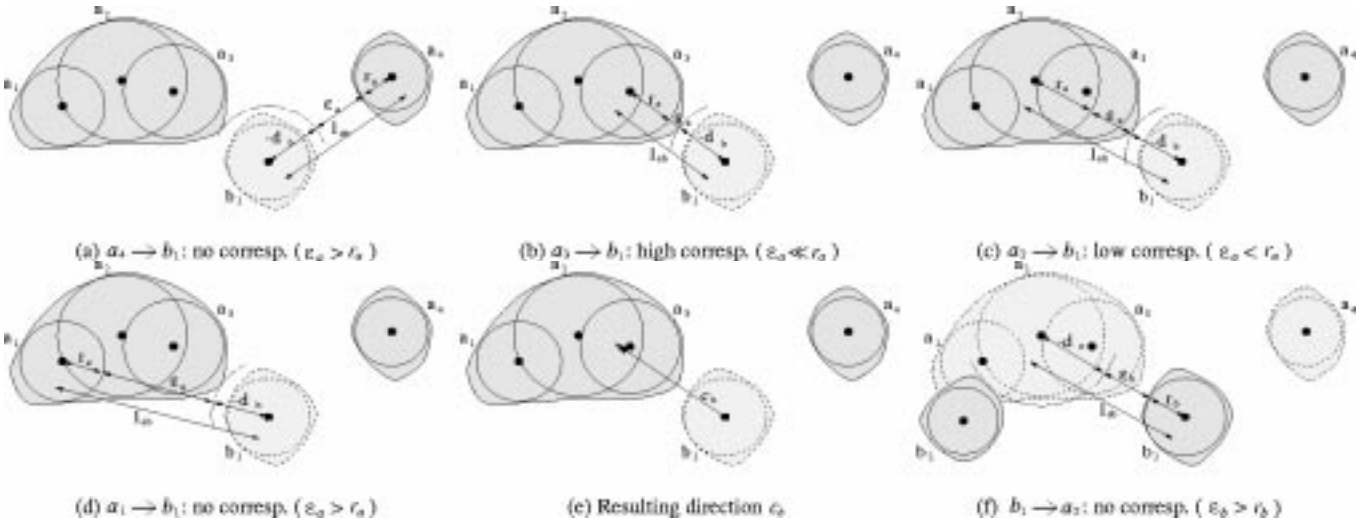

Fig. 4. Effect of ε_a and ε_b on correspondence. Discs $a_1 \ldots a_4$ (shown in dark grey) are from one plane, and disc b_1 (shown light grey and dashed) is projected onto this plane from another. (a)-(d) The calculation of "coverage error" ε_a for each pair of discs—the converse error ε_b is not shown, since in this case it is nearly the same for all disc pairs. (e) The resulting correspondence vector for disc b_1. (f) demonstrates the effect on ε_b of adding disc b_2.

The correspondence for each disc on each cross section is calculated as the weighted sum of contributions from each of the discs on the neighboring cross section:

$$\vec{c}_a = \begin{cases} \dfrac{1}{\sum_{b \in \text{discs}} \omega_b} \sum_{b \in \text{discs}} \omega_b \vec{l}_{ab}, & \text{if } \sum_{b \in \text{discs}} \omega_b > 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $\vec{c}_a$ is the correspondence vector for disc a, $\vec{l}_{ab}$ is the vector from the center of disc a to disc b, and $\omega_b$is a weighting indicating the likelihood of correspondence of discs a and b, which are from cross sections on neighboring planes A and B. All distances and vectors are calculated in either of these planes (it does not matter which), by projection of the discs from the alternate plane along the average normal to these planes.

The calculation of correspondence likelihood for each pair of discc is the most important step in this process. It is estimated by comparing, in each plane, the difference in distance transform values at the center of each disc with the planar distance between the disc centers. This gives an error which, if it is small compared with the radii of each of the discs, is used as the likelihood estimate for this pair of discs. If the error is larger than either of the disc radii, no correspondence is made between these discs, and the likelihood estimate is set to zero

$$\omega_b = \begin{cases} \frac{1}{(\varepsilon_a^2+\mu)} + \frac{1}{(\varepsilon_b^2+\mu)}, & \text{if } \varepsilon_a < r_a \text{ and } \varepsilon_b < r_b \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$\varepsilon_a = \text{chamfer}(\vec{l}_{ab}) - |r_a - d_b|$$
$$\varepsilon_b = \text{chamfer}(\vec{l}_{ab}) - |r_b - d_a| \quad (3)$$

ω_b is itself simply a combination of the "coverage errors" for each disc, ε_a and ε_b, with a small value μ chosen to be the square of the chamfer distance represented by one image pixel, which prevents an infinite weighting. ε_a is an estimate of the extent to which the distance field of plane A at the projected center of disc b is affected by disc a. Conversely, ε_b is an estimate of the extent to which the distance field of plane B at the projected center of disc a is affected by disc b. r_a and r_b are the distance field values at the disc centers, and d_a and d_b are the distance field values at the projection of each disc on the opposite plane. The function chamfer $(\vec{l}_{ab})$ is an estimate of $|\vec{l}_{ab}|$, measured along a path formed entirely from lines parallel or at 45° to the image edges. This is the same estimate as used in the distance transformation of the cross sections [14].

This way of weighting the contributions from each disc has a variety of features which make it attractive.

- Discs will only correspond with the nearest discs on the neighboring cross section if the nearness relationship is reciprocal. This allows regions to be left unconnected, if appropriate.
- It is *not* necessary for regions (or discs) to overlap in order to correspond.
- Small discs will only tend to have a local effect on correspondence, unless the contour they represent is itself small. Larger discs on the same contour will take priority in the far field.
- The only limiting assumption is that at least one contour must be connected to one other contour on a neighboring plane, i.e., at least one object must span the cross sections.

Fig. 4 shows an example of the calculation of coverage error, given in (3), for one disc from a pair of simple cross sections. Essentially, ε is large if there are other discs on the cross section which are nearer to the center of the projected disc. This is equally the case for discs from other contours, as in Fig. 4(a), or from the same contour, as in Fig. 4(d). The criterion $\varepsilon_a < r_a$ for there to be any correspondence at all, rather than $\varepsilon_a = 0$, allows for the loose representation of the contour by a small number of discs—in practice ε will rarely be equal to zero. If there *is* correspondence, then ε indicates how strong the correspondence is, so for instance disc a_3 in Fig. 4(b) has a lower error than disc a_2 in Fig. 4(c). This leads to a correspondence vector which is closer to the disc with the lower error, as in Fig. 4(e).

In the previous case, the converse error ε_b is nearly the same for each pair of discs, and as a result has little bearing on the

correspondence. In general, both errors must be considered before correspondence between a pair of discs is determined. The addition of another contour to plane B in Fig. 4(f) demonstrates this. ε_b for discs b_1 and a_2 is now greater than r_b, so even though ε_a has not changed, there is no longer any correspondence between these discs. The effect on the correspondence vector for b_1 is to move it further away from b_2, thus "sharing out" the larger contour on plane A more equally between b_1 and b_2.

The correspondence $\vec{c}$ is calculated for each interior disc, based on all the other interior discs, and for each exterior disc, based on all the other exterior discs, such that interior discs do not correspond to exterior discs, and *vice versa*. Any discs which have no correspondence at all are ignored in later processing. The result of this operation on the cross sections of Fig. 2(a) is shown in Fig. 2(d)—note that in this case the external discs have no partners on neighboring planes and, hence, do not contribute to the correspondence estimate.

5) Point Correspondence: Once the correspondence for each *disc* has been estimated, the correspondence (and, therefore, the interpolation direction) at any *point* on each plane can also be estimated from a weighted sum of the disc correspondences on that plane. This is performed in much the same way as for disc correspondence, save that now the information from both internal and external discs must be combined. The point correspondence $\vec{c}_p$ at any point p on the plane, is as follows:

$$\vec{c}_p = \begin{cases} \dfrac{1}{\sum_{a\in\text{int}}\omega_a}\displaystyle\sum_{a\in\text{int}}\omega_a\vec{c}_a & \displaystyle\sum_{a\in\text{ext}}\omega_a = 0 \\ \dfrac{\left[(l_{\text{int}}+2d_p)\sum_{a\in\text{int}}\omega_a\vec{c}_a + l_{\text{int}}\sum_{a\in\text{ext}}\omega_a\vec{c}_a\right]}{2(l_{\text{int}}+d_p)\left(\sum_{a\in\text{int}}\omega_a+\sum_{a\in\text{ext}}\omega_a\right)} & d_p > 0 \\ \dfrac{\left[(l_{\text{ext}}-2d_p)\sum_{a\in\text{int}}\omega_a\vec{c}_a + l_{\text{ext}}\sum_{a\in\text{ext}}\omega_a\vec{c}_a\right]}{2(l_{\text{ext}}-d_p)\left(\sum_{a\in\text{int}}\omega_a+\sum_{a\in\text{ext}}\omega_a\right)} & d_p \le 0 \end{cases} \tag{4}$$

where

int implies internal discs;
ext implies external discs;
d_p distance field value at the point p.

There are three separate cases in (4). In the first and simplest case, if the geometry is such that there are no external discs, or no external discs have any correspondence, then the point correspondence depends solely on the internal discs much as in (1). Otherwise, the point correspondence is calculated by a combined weighting of internal and external discs, using a slightly different equation for points inside ($d_p > 0$) and outside ($d_p < 0$) the object cross section. This combined weighting is further affected by the estimate of the distance to the internal and external disc sets, l_{int} and l_{ext}. These estimates are calculated using a similar weighting to the first case of (4)

$$l_{\text{int}} = \frac{1}{\sum_{a\in\text{int}}\omega_a}\sum_{a\in\text{int}}\omega_a l_a$$
$$l_{\text{ext}} = \frac{1}{\sum_{a\in\text{ext}}\omega_a}\sum_{a\in\text{ext}}\omega_a l_a \tag{5}$$

The effect of these values in (4) is to ensure that the use of internal or external discs is governed by the distance transform value d_p, i.e., how far the current point is inside or outside the object cross section. Inside the cross section, the weighting of internal discs increases as the point moves closer to the nearest internal disc. Outside the cross section, the weighting of external discs increases as the point moves closer to the nearest external disc. At the edge of the cross section ($d_p = 0$), internal and external disc correspondences are evenly weighted.

The weighting ω_a and coverage error ε_a in (4) and (5) are similar to those in (2) and (3), save that the error is now based on the distance field d_p at the point position p, rather than at the center of the projected disc

$$\omega_a = \frac{1}{{\varepsilon_a}^2 + \mu} \tag{6}$$
$$\varepsilon_a = \text{chamfer}(\vec{l}_{ap}) - |r_a - d_p|. \tag{7}$$

The final correspondence for a point *between* neighboring planes is found from the average of the values of $\vec{c}_p$ from each plane, at the intersection of the average plane normal with these planes.

6) Interpolation and Iso-Surface Extraction: The point correspondence $\vec{c}_p$ is a vector lying within the plane of one of the neighboring cross sections. If the cross sections are parallel, this can easily be converted to an interpolation direction by adding it to the vector distance between planes. This operation is essentially the same for nonparallel planes, although care is needed to constrain the eventual interpolation direction [19].

The distance field values at each point in a regular lattice are found by first establishing the interpolation direction for that point, then linearly interpolating the neighboring distance transforms in this direction. We use a body-centered cubic lattice, which has better sampling efficiency than a simple cubic lattice, and is appropriate for using regularised marching tetrahedra to extract the zero iso-surface of the interpolated data [11]. The resolution of the interpolated surface can be varied by varying the resolution of the sampling lattice.

IV. Results

Surfaces rather than numerical results are shown since they convey much more information, and provide an understanding of the underlying cause of the differences between the algorithms. Volume measures [13] or binary difference measures [15] can hide gross local changes in topology. Surfaces were estimated by four variants of shape-based interpolation.

1) *Shape-Based:* The original algorithm [1] was implemented (for nonparallel planes the interpolation direction was taken to be the average planar normal to those planes).
2) *Centroid-Guided:* This was a combination of techniques described in [15] and [16], whereby the interpolation was performed for each pair of corresponding contours along the direction of linked centroids. The bounding rectangles for each contour were also scaled independently in the x and y directions such that they were the same size. Correspondence of contours was determined manually. The union of the generated surfaces was used to create the final object surface. The method of edge-shrinking interpolation [17] was not implemented since it is much more complicated, and in practice gives very similar results to the centroid-guided technique. The results would

be *slightly* better, due to the more sophisticated alignment of corresponding contours, but the same gross artifacts would be present.

3) *Previous Disc-Guided:* The technique described in our previous paper [19].
4) *New Disc-Guided:* The technique described here.

All the surfaces were generated by triangulating the zero iso-surface of the interpolated distance fields, using regularised marching tetrahedra [11]. These triangulations were rendered with Gouraud shading using the Geomview[2] visualization package. Processing was done on a Silicon Graphics Indigo 2 R10000 workstation[3] with 256-Mb RAM.

It should be noted that there are many examples of simple surfaces where the difference between the four techniques described above is negligible. The following data have been selected to highlight the properties of each algorithm over a broad range of surface types, particularly concentrating on surfaces which produce different results with each algorithm.

In vivo ultrasound data was recorded using a Toshiba Powervision 7000[4] ultrasound machine with a 3.75-MHz convex curvilinear array probe. A magnetic field position sensor was mounted on this and the position signal, in addition to the video output of the ultrasound machine, was fed to a Silicon Graphics Indy workstation. The child's skull in Fig. 6 is from data provided with the 3DViewnix[5] visualization package [30]. This is a high resolution (0.41 mm$\times$ 0.41 mm $\times$ 1 mm) CT scan. The female pelvis in Fig. 7 is from CT data provided by the Visible Human Project[6]. In this case, the original CT resolution was 0.9375 mm $\times$ 0.9375 mm $\times$ 1 mm. The human liver in Fig. 8 is from MRI data provided by the Visible Human Project, in this case from the male data set, the original resolution being 1.875 mm $\times$ 1.875 mm $\times$ 4 mm.

Segmentation of the data, and acquisition of the ultrasound images, was performed using Stradx[7] software [31]. The ultrasound and MRI data was segmented manually, using 31 and 5 cross sections, respectively. The CT data was segmented by first thresholding, then manually "tidying up" the cross sections to remove some of the incorrectly thresholded regions. Every sixth cross section was segmented for the skull, and every tenth for the pelvis.

The foetus in Fig. 5 is quite a complex shape—many cross sections contain only one contour which nevertheless includes the body and limbs. Centroid-guided interpolation performs very badly when connecting such contours with simple cross sections of the torso—thin "webs" are formed between regions which should not be connected. The previous disc-guided interpolation improves on this, but still suffers from similar drawbacks, and in this case shape-based interpolation performs better than either of them. However, shape-based interpolation incorrectly connects the upper hand with the head, a connection which is not made by the new disc-guided interpolation, even though the contours overlap, due to the use of external discs.

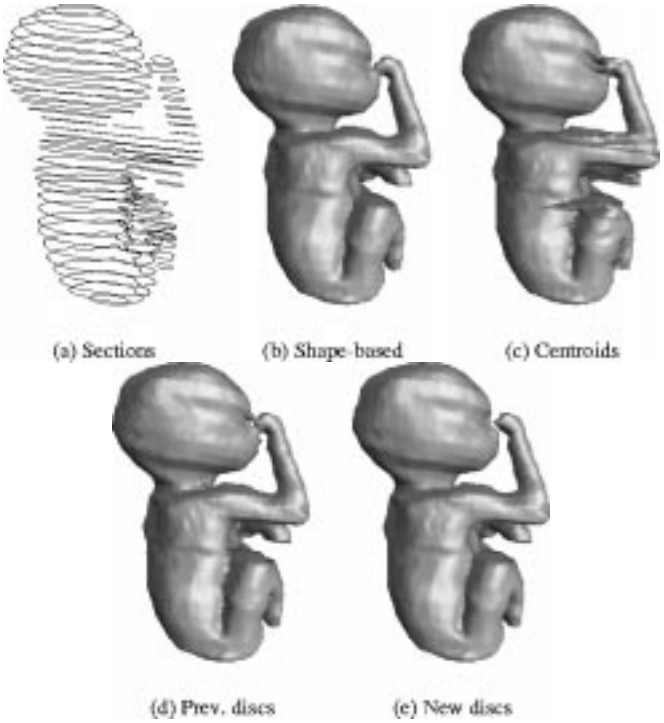

Fig. 5. Foetus at week 22. The cross sections were manually outlined in the original images (B-scans) of a freehand 3-D ultrasound investigation.

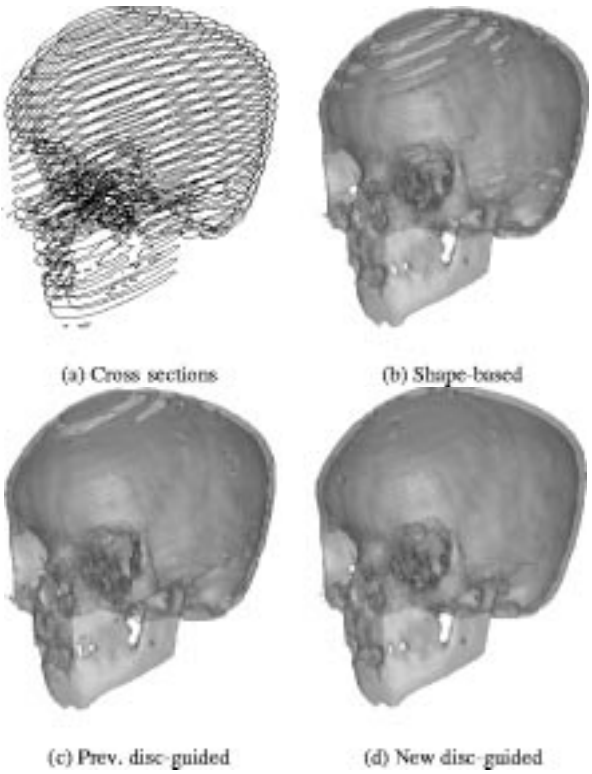

Fig. 6. Child's skull. The cross sections were automatically segmented by thresholding CT data. The surface is rendered with a slight transparency, so that both the inner and outer surfaces can be seen.

[2]http://www.geom.umn.edu/software/geomview/.

[3]Silicon Graphics Incorporated, Mountain View, CA.

[4]Toshiba America Medical Systems, Tustin, CA.

[5]From `CHILD.IM0`, 3DViewnix v1.1.1 (c) 1993–1996 M I P G University of Pennsylvania, Philadelphia, PA. All Rights Reserved.

[6]The Visible Human Project is an initiative from the National Library of Medicine in Bethesda, MD.

[7]http://svr-www.eng.cam.ac.uk/~rwp/stradx/

TABLE I
PROCESSING TIME. THESE INCLUDE ALL OPERATIONS AFTER CREATION OF THE CROSS SECTIONS, UP TO AND INCLUDING DISPLAY OF THE TRIANGULATED SURFACE

Object	Modality	Triangles	Processing Time /secs			
			Shape-based	Centroids	Prev. disc	New disc
Hepatic ducts	Ultrasound	1,800	0.7	0.8	0.9	1.1
Foetus	Ultrasound	39,000	8.1	12.6	10.3	12.9
Child's skull	CT	93,000	13.9	-	100.5	147.1
Pelvis	CT	103,000	31.6	-	60.1	71.1
Liver	MRI	7,000	1.5	1.6	1.7	1.7

Times were recorded on a Silicon Graphics INDIGO 2 R10000 Workstation.

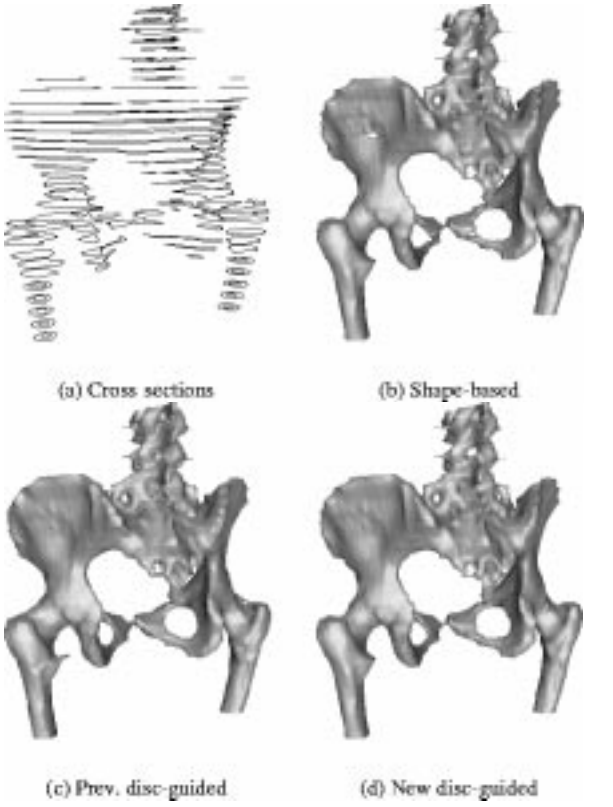

Fig. 7. Female pelvis. The cross sections were semi-automatically segmented by thresholding CT data, then manually editing the results of this threshold operation.

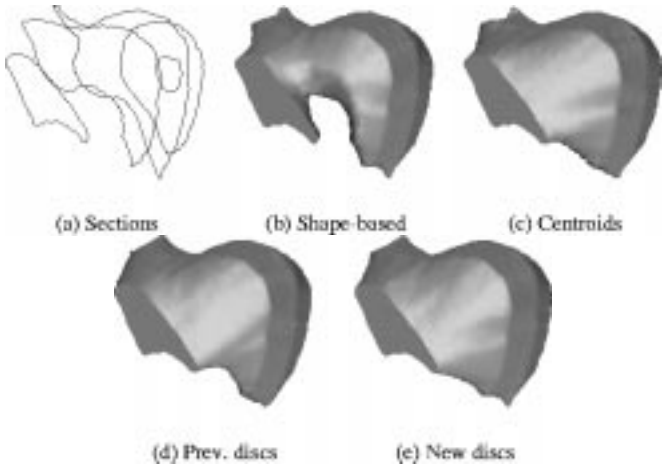

Fig. 8. Human liver. The cross sections were manually outlined from coronal MRI slices of the abdomen.

The child's skull in Fig. 6 is a good example of the sort of complicated cross sections which result from automatically thresholding medical images. Most cross sections have many contours, and each contour is complex. The centroid-guided technique is not appropriate in this case, since there are multiple contours which are neither simple nor tree-like. Liu's method [17] would have given sensible results if the inner skull contour was separate from the outer. However, this is not the case on some of the cross sections and, hence, there is no topological hole in these cross sections. The cross sections in the upper part of the skull in particular are far from overlapping, which results in the separate rings constructed by shape-based interpolation.

The previous disc-guided method improves on this result, but it is still limited by the condition that the discs representing each cross section must overlap in order for them to correspond. There is no such constraint in the new disc-guided interpolation and, hence, the skull can still be correctly reconstructed despite the lack of overlap.

The female pelvis in Fig. 7 shows the interplay between the interpolation process and the resolution of the interpolated data from which the triangulation is created. The hole in the left-hand side of the pelvis in Fig. 7(b) is the result of under-sampling of data—the pelvis is very thin in this region. Even though the other figures are generated at the same resolution, this effect has been reduced by a better interpolation direction. Fig. 7(c) also demonstrates the "spurs" which can be generated by the previous disc-guided interpolation, for instance from the left-hand femur to the lower part of the pelvis.

The few cross sections segmented from the liver data in Fig. 8 contain enough information to give a reasonable idea of the shape of the liver and estimate of its volume. However, this is not enough information for shape-based interpolation to estimate the surface; hence, the large invagination in Fig. 8(b). This sort of shape should be ideally suited to centroid-guided interpolation, however, the disc-guided interpolation performs slightly better (particularly noticeable on the sharp upper edge of the liver). This is because the calculation of centroid position is dominated by the bulk of the object, hence, small features on large objects do not contribute and may not be reconstructed correctly. Disc-guided interpolation uses local information and can readjust the interpolation direction in the region of the feature.

A. Processing Time

Processing times for each of the surfaces in this paper are shown in Table I. The software was designed for nonparallel

cross sections, and the processing time for the parallel cases (CT and MRI) would have been significantly less had this not been the case. However, since this applies equally to all the surface interpolation algorithms, the times still give a good relative measure of performance.

Unsurprisingly, the more complex disc-based algorithms take longer than the centroid-based or standard shape-based algorithms. However, the increase in time is often quite small, since all algorithms require calculation of the distance transform and triangulation of the interpolated data. The additional processing time for the disc-based algorithms scales with the number of discs involved in the correspondence calculation, hence, long thin objects (for instance the skull in Fig. 6) take more time to process than more spherical objects. The new disc-based algorithm takes longer than the previous algorithm, since external discs are also considered.

V. Conclusion

The method of surface interpolation presented in this paper is capable of improving on the results of shape-based interpolation in many cases where there are few cross sections or the cross sections vary significantly between planes. In addition, and unlike previous methods, it can be relied upon not to generate additional artifacts in cases where shape-based interpolation is already a good surface estimator. This is an important result, since it indicates that the technique can be safely used on many different types of data.

The technique introduces a processing overhead, which increases with the number of maximal discs required to represent any cross section, but this overhead is small for many practical cases. Even where the overhead is large, the reconstruction processing time is still small compared with the time required to segment the data. In addition, the ability of this algorithm to interpolate sparse cross sections reduces the number which need to be segmented, thus potentially *reducing* the total time from scanning to 3-D display.

Acknowledgment

The authors would like to thank D. Galletly and P. Gosling for permission to publish the data in Fig. 5. G. Treece would his wife S. Treece for her support.

References

[1] G. T. Herman, J. Zheng, and C. A. Bucholtz, "Shape-based interpolation," in *IEEE Comput. Graphics Applicat. Mag.*, May 1992, pp. 69–79.
[2] W. A. Barrett and E. N. Mortensen, "Fast, accurate, and reproducible live-wire boundary extraction," *Lecture Notes Comput. Sci.*, vol. 1131, pp. 183–192, 1996.
[3] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes—Active contour models," *Int. J. Comput. Vision*, vol. 1, no. 4, pp. 321–331, 1987.
[4] D. Levin, "Multidimensional reconstruction by set-valued approximations," *IMA J. Numer. Anal.*, vol. 6, pp. 173–184, 1986.
[5] C. L. Bajaj, E. J. Coyle, and K.-N. Lin, "Arbitrary topology shape reconstruction from planar cross sections," *Graphical Models Image Processing*, vol. 58, no. 6, pp. 524–543, Nov. 1996.
[6] G. Barequet and M. Sharir, "Piecewise-linear interpolation between polygonal slices," *Comput. Vision Image Understanding*, vol. 63, no. 2, pp. 251–272, Mar. 1996.
[7] J.-D. Boissonnat, "Shape reconstruction from planar cross sections," *Comput. Vision Graphics, Image Processing*, vol. 44, pp. 1–29, 1988.
[8] D. Meyers, S. Skinner, and K. Sloan, "Surfaces from contours," *ACM Trans. Graphics*, vol. 11, no. 3, pp. 228–258, July 1992.
[9] B. A. Payne and A. W. Toga, "Surface reconstruction by multiaxial triangulation," *IEEE Comput. Graphics Applicat. Mag.*, vol. 14, no. 6, pp. 28–35, Nov. 1994.
[10] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Comput. Graphics*, vol. 21, no. 4, pp. 163–168, July 1987.
[11] G. M. Treece, R. W. Prager, and A. H. Gee, "Regularised marching tetrahedra: Improved iso-surface extraction," *Comput. Graphics*, vol. 23, no. 4, pp. 583–598, 1999.
[12] B. A. Payne and A. W. Toga, "Distance field manipulation of surface models," in *IEEE Comput. Graphics Applicat. Mag.*, Jan. 1992, pp. 65–71.
[13] S. P. Raya and J. K. Udupa, "Shape-based interpolation of multidimensional objects," *IEEE Trans. Med. Imag.*, vol. 9, pp. 32–42, Feb. 1990.
[14] G. Borgefors, "Distance transformations in digital images," *Comput. Vision Graphics Image Processing*, vol. 34, pp. 344–371, 1986.
[15] W. E. Higgins, C. Morice, and E. L. Ritman, "Shape-based interpolation of tree-like structures in three-dimensional images," *IEEE Trans. Med. Imag.*, vol. 12, pp. 439–450, June 1993.
[16] M. Loughlin, I. Carlbom, C. Busch, T. Douglas, L. Egevad, H. Frimmel, M. Norberg, I. Sesterhenn, and J. M. Frogge, "Three-dimensional modeling of biopsy protocols for localized prostate cancer," *Computerized Med. Imag. Graphics*, vol. 22, pp. 229–238, 1998.
[17] Y.-H. Liu, Y.-N. Sun, C.-W. Mao, and C.-J. Lin, "Edge-shrinking interpolation for medical images," *Computerized Med. Imag. Graphics*, vol. 21, no. 2, pp. 91–101, 1997.
[18] S.-Y. Chen, W.-C. Lin, C.-C. Liang, and C.-T. Chen, "Improvement on dynamic elastic interpolation technique for reconstructing 3-D objects from serial cross sections," *IEEE Trans.Med. Imag.*, vol. 9, no. 1, pp. 71–83, 1990.
[19] G. M. Treece, R. W. Prager, A. H. Gee, and L. Berman, "Fast surface and volume estimation from nonparallel cross-sections, for freehand 3-D ultrasound," *Med. Image Anal.*, vol. 3, no. 2, pp. 141–173, 1999.
[20] T. D. Haig, Y. Attikiouzel, and M. Alder, "Border marriage: Matching of contours of serial sections," *Inst. Electr. Eng. Proc.*, vol. 138, no. 5, pp. 371–376, Oct. 1991.
[21] C. L. Bajaj, F. Bernardini, and G. Xu, "Reconstructing surfaces and functions on surfaces from unorganized three-dimensional data," *Algorithmica*, vol. 19, pp. 243–261, 1997.
[22] E. Bittar, N. Tsingos, and M.-P. Gascuel, "Automatic reconstruction of unstructured 3D data: Combining a medial axis and implicit surfaces," *Comput. Graphics Forum*, vol. 14, no. 3, pp. C-457–C-468, Aug. 1995.
[23] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *Comput. Graphics*, vol. 26, no. 2, pp. 71–78, July 1992.
[24] M. E. Legget, D. F. Leotta, E. L. Bolson, J. A. McDonald, R. W. Martin, X.-N. Li, C. M. Otto, and F. H. Sheehan, "System for quantitative three-dimensional echocardiography of the left ventricle based on a magnetic-field position and orientation sensing system," *IEEE Trans. Biomed. Eng.*, vol. 45, pp. 494–504, Apr. 1998.
[25] V. V. Savchenko, A. A. Pasko, O. G. Okunev, and T. L. Kunii, "Function representation of solids reconstructed from scattered surface points and contours," *Comput. Graphics Forum*, vol. 14, no. 4, pp. 181–188, 1995.
[26] G. Turk and J. F. O'Brien, "Shape transformation using variational implicit functions," in *Proc. Siggraph 99: Computer Graphics*, Los Angeles, CA, Aug. 1999, pp. 335–342.
[27] F. Lazarus and A. Verroust, "Three-dimensional metamorphosis: A survey," *Visual Comput.*, vol. 14, pp. 373–389, 1998.
[28] D. Cohen-Or, D. Levin, and A. Solomovici, "Three-dimensional distance field metamorphosis," *ACM Trans. Graphics*, vol. 17, no. 2, pp. 116–141, Apr. 1998.
[29] W.-S. V. Shih, W.-C. Lin, and C.-T. Chen, "Morphologic field morphing: Contour model-guided image interpolation," *Int. J. Imag. Syst. Technol.*, vol. 8, pp. 480–490, 1997.
[30] J. K. Udupa, D. Odhner, S. Samarasekera, R. Goncalves, K. Iyer, K. Venugopal, and S. Furuie, "3DVIEWNIX: An open, transportable, multidimensional, multi-modality, multiparametric imaging software system," *SPIE Proc.*, vol. 2164, pp. 58–73, 1994.
[31] R. W. Prager, A. H. Gee, and L. Berman, "Stradx: Real-time acquisition and visualization of freehand 3D ultrasound," *Med. Image Anal.*, vol. 3, no. 2, pp. 129–140, 1999.