Surface interpolation
from sparse cross-sections
using region correspondence

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

CUED/F-INFENG/TR 342
March 1999

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

E-mail: gmt11,rwp,ahg@eng.cam.ac.uk, lb@radiol.cam.ac.uk

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 (3-D). 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 deal with 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. In this paper, an algorithm is presented which can interpolate a surface through sparse, complex cross-sections. This is an extension of *maximal disc guided interpolation* [25], which is itself based on *shape based interpolation* [8, 21]. 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.

Contents

1 Introduction

1.1 Motivation

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 the planes. The simplest interpolation technique is linear, using the original grey scale values, however this can generate artifacts in subsequent surface renderings. Hence, object based interpolation methods are often used instead, which interpolate the object cross-section rather than the grey scale values.

Segmentation of medical data is non-trivial, 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 [1, 11] is often used instead. 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 planes (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 non-parallel 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.

1.2 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 the minimum distance at 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 [13].

Direct triangulation of points on the cross-sections is a difficult 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 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 [2, 17]. In addition, there have been very few examples in the literature of triangulation of non-parallel cross-sections [6, 19].

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 [15] being the most widely used, although newer variants of this can give better quality meshes [24]. In some cases, extracting the iso-surface at some non-zero value (i.e. at a constant distance from the actual surface) can also provide useful information [18]. The functional approach also lends itself more naturally to data from non-parallel planes, and indeed scattered data, and there are several examples of this in the literature [10, 22].

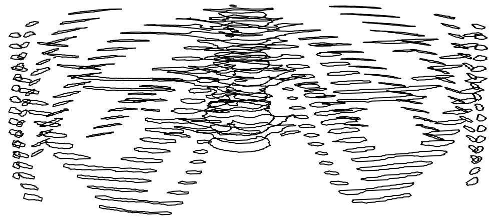
Shape-based interpolation is the simplest form of distance function creation, but it does not cope well when neighbouring cross-sections are significantly different. Various extensions to the original algorithm have been suggested to improve on this behaviour, but none can be relied upon to do so in all situations. One of the major problems faced by these algorithms is the need to determine the connectivity of the object cross-sections. This has previously only been determined for *whole objects*, which is not appropriate when neighbouring cross-sections are very different — a contour on one cross-section may connect to only a *part* of a contour on the other cross-section. A derivative of shape-based interpolation which determined connectivity from *region* correspondence, has been presented in [25]. This gives very good results for many objects, but unfortunately cannot be relied upon to improve the original algorithm in all circumstances. An extension of this algorithm is presented here, which improves on its performance, without generating additional artifacts.

It is important to recognise 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. The only exception to this would be if the connectivity of each part of each cross-section was determined manually by an expert. This is simply a consequence of under-sampling — if the inter-slice spacing is greater than the largest frequency present in the data, it will be reconstructed as a lower frequency surface. For example, the ribs in Figure 1 are incorrectly joined together when only a few cross-sections are used, since this is the simplest connection based on those cross-sections. 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 [7] for a discussion of connectedness).

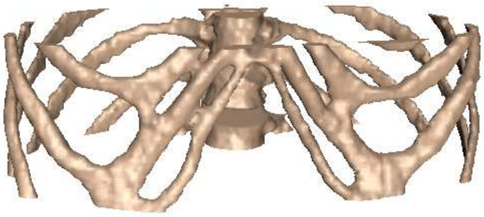
1.3 Original contribution

This report presents a major extension of *maximal disc guided interpolation*, first presented in [25]. This is a surface interpolation algorithm which is itself an extension of *shape based interpolation* [8, 21]. It has been improved in several important areas:

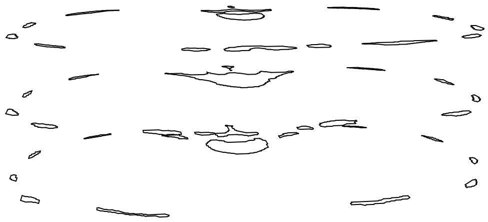
- The traditional 'branching' and 'correspondence' problems are combined by determining 'region correspondence'. This is achieved by representing the cross-sections as a set of maximal discs, derived from the distance transformation of the cross-section. The distance transformation is also used to determine the correspondence between discs on neighbouring cross-sections. This dramatically improves the estimated surface of



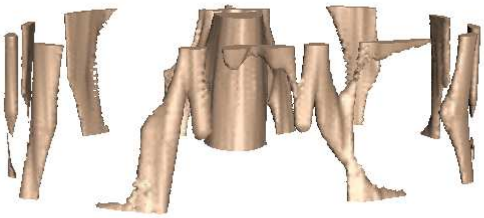
(a) Cross-sections



(b) Interpolated surface from (a)



(c) Fewer cross-sections



(d) Interpolated surface from (c)

Figure 1: **Section of human ribs.** The contours were automatically segmented by thresholding CT data.

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 in determining 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 report presents comparisons of surfaces from 3-D ultrasound, CT and MRI data.

2 Related work

2.1 Object based interpolation

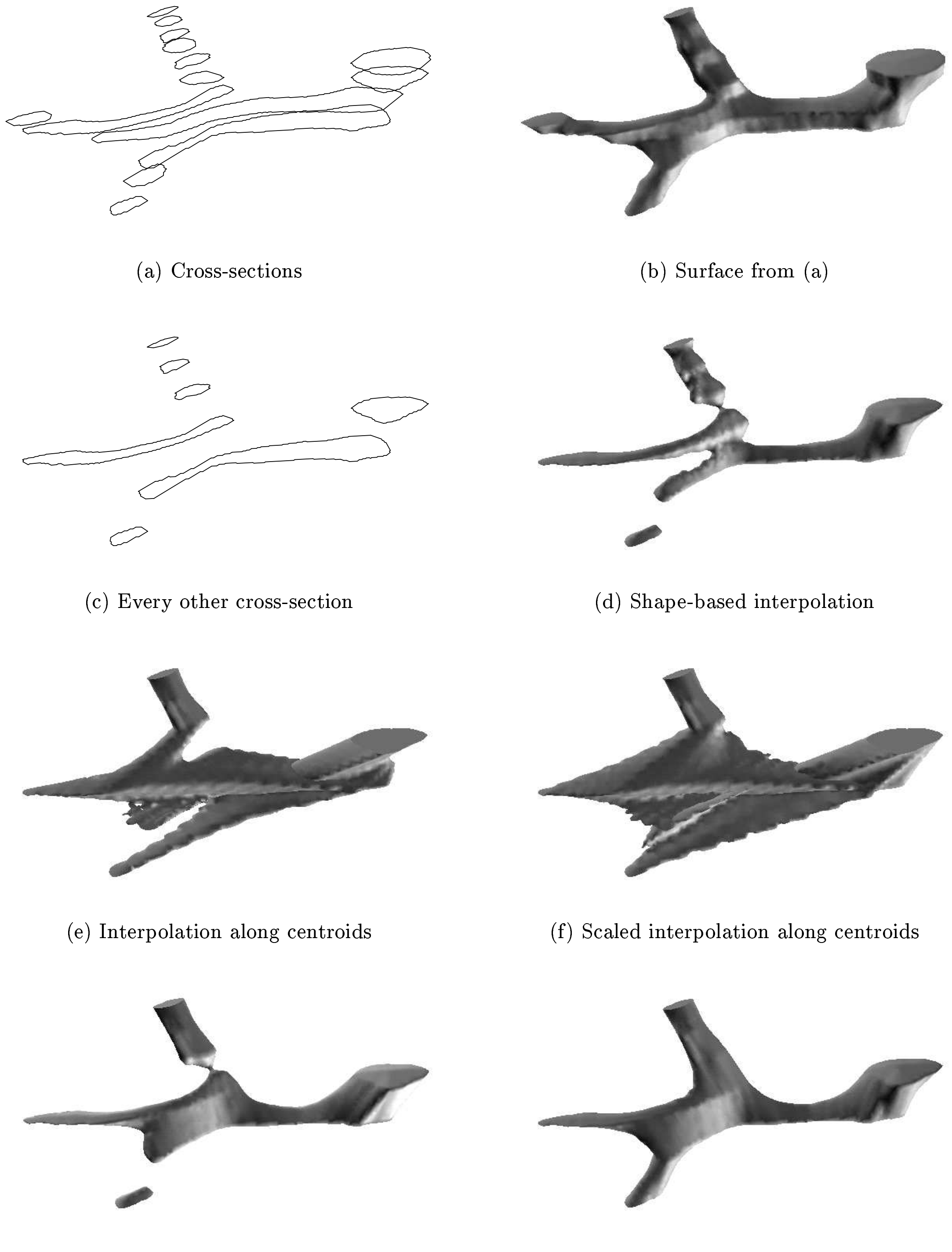
Object based interpolation methods use the shape of objects within the data as a guide for the interpolation. They can be used to interpolate grey-scale data, or applied to the cross-sections themselves, thus producing a binary interpolation of the object. In shape-based interpolation [8, 21], the binary cross-sections 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 [3]. Pixels inside the object are assigned positive distances, and those outside negative. The distance transformed image can then be interpolated using linear interpolation, to give new distance transformed images. Thresholding these images at zero then gives the new interpolated cross-sections, from which the surface can be rendered.

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 Figure 2, which shows some (near) parallel cross-sections from a freehand 3-D ultrasound scan of a part of the hepatic system. Figures 2(a) and (b) show the actual anatomy, whereas Figure 2(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) [9], as in Figure 2(e). This works well for tree-like shapes which have been scanned transverse to the main axis, like the top three cross-sections in Figure 2(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 — but makes the situation worse in the case of Figure 2(f). This algorithm has been implemented recently for modelling prostates for ultrasound guidance of biopsy protocols [16]. An implementationally more complicated, but practically very similar approach based on mathematical morphology, which aligns objects on each cross-section using an iterative minimisation of distance transform values, is presented in [14].

The connectivity in all of the above algorithms is based on whole contours, but it can be seen from Figures 2(e) and (f) that this is not always appropriate: the smaller contours only 'connect' with a small part of the longer contours. An alternative object-based interpolation approach, dynamic elastic interpolation, has been proposed, which also begins to address this problem [4]. Here, a force field is constructed which iteratively deforms one cross-section towards the other. The initial formulation produced similar results to centroid guided shape-based interpolation for complex objects. However, this behaviour 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. This region correspondence is determined by representing each cross-section by a set of discs which lie inside the object. The result of this algorithm, maximal disc guided interpolation, is shown in Figure 2(g). This is clearly an improvement on the previous shape-based techniques, however it is still not a faithful representation of the actual anatomy, Figure 2(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 Figure 2(h), which in this case correctly represents the actual anatomy.



(a) Cross-sections

(b) Surface from (a)

(c) Every other cross-section

(d) Shape-based interpolation

(e) Interpolation along centroids

(f) Scaled interpolation along centroids

(g) Previous disc-guided interpolation

(h) New disc-guided interpolation

Figure 2: **Part of the hepatic system.** The cross-sections were manually outlined on the original B-scans of a freehand 3-D ultrasound investigation. (a) and (b) show the original cross-sections, (d) to (h) show the surface interpolated from the selected cross-sections shown in (c).

2.2 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. In fact, distance field interpolation has recently been used for this purpose [5]. 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 surface which already exists. 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 [23]. Here, shape-based interpolation is used to determine intermediate cross-sections which are used to form the basis of the control points required by the field-morphing process.

2.3 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. Hoppe [10] developed an approach based on the idea of forming a distance function which can then be thresholded to generate the object surface. In this case, however, the function is formed by combining contributions from each point, having first carefully estimated the surface normal at each of these points. This represents a very general solution, but as a result it is not well constrained, and many points are required to ensure a correct reconstruction.

In freehand 3-D ultrasound, the cross-sections are already non-parallel, and therefore this approach becomes more attractive. It has recently been implemented for reconstructing the left ventricle of porcine hearts in a water bath [12]. 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.

It is also possible to use volume splines to reconstruct the surface from cross-sections [22]. In this method, 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 'distance' function which is then thresholded at zero to give the eventual surface. Since this involves the inversion of an $N \times N$ matrix, where N is the number of points on each cross-section, the approach is limited to relatively small data sets, although it is faster than using a volume spline on all the cross-sections simultaneously.

3 Maximal disc guided interpolation

3.1 Overall strategy

The overall strategy of the surface interpolation algorithm is displayed in Figure 3. Fundamentally, this is the same as shape-based interpolation, however the interpolation direction is allowed to vary at each point dependent on the object 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 the weighting is again derived from the original distance transforms. Thus, the distance function can be found at any point in space between neighbouring cross-sections and evaluated on a grid suitable for performing 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. Centres 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.

3.2 Processing steps

3.2.1 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, as those shown in Figure 3(a).

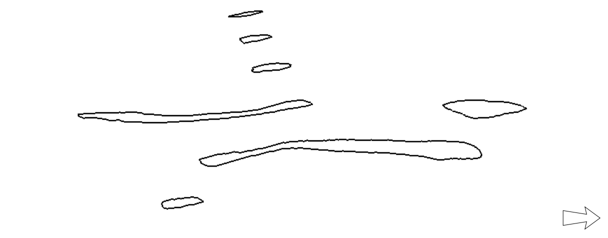
3.2.2 Distance field transformation

An efficient algorithm to calculate the distance transformation for a rectangular area is given in [3]. The distance transform is calculated for each cross section, as in Figure 3(b). In addition, a far point distance algorithm described in [25] is used to provide distance values outside of this rectangular region.

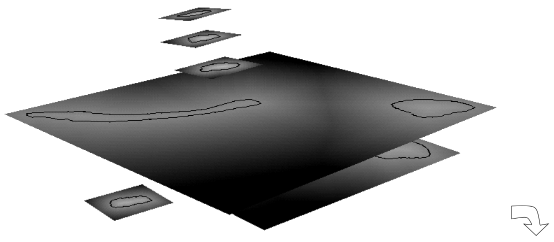
3.2.3 Maximal disc representation

Extraction of a set of maximal discs is also described in [25]. The disc centres are a sub-set 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 centre. In order to include external discs (those outside, rather than contained by, the contours), 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 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 Figure 3(a) is shown in Figure 3(c).

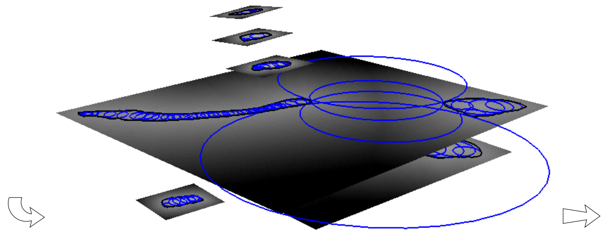
[1]In reality, there are situations where external discs exist outside this area, but this has little practical effect on the results.



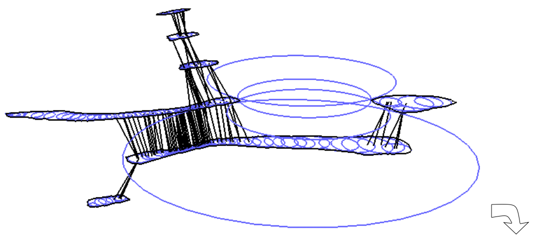
(a) Original cross-sections



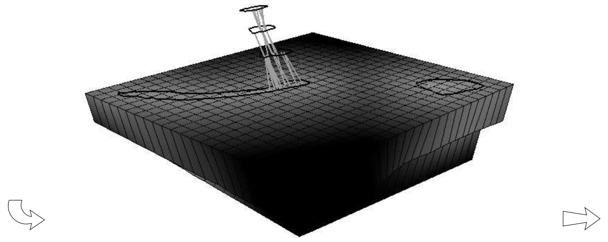
(b) Distance transformation



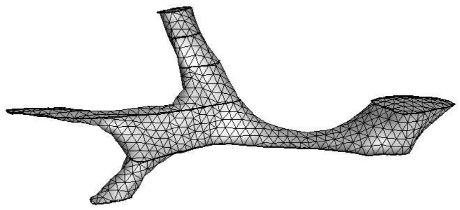
(c) Maximal disc extraction



(d) Disc correspondence

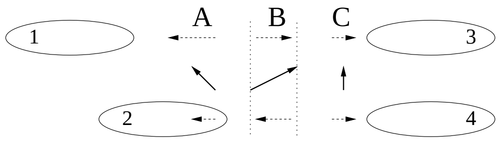


(e) Disc-guided interpolation

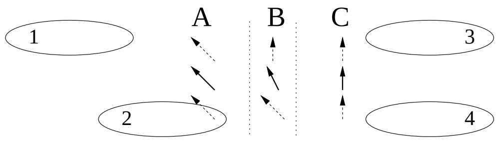


(f) Triangulation of zero iso-surface

Figure 3: **Surface interpolation algorithm.** Distance transformation is first performed on each cross-section (b), then a set of representative discs can be extracted from these transformations (c). Region correspondence is estimated for each disc (d), and this is combined to give a correspondence direction at any point. This provides the interpolation direction for the distance field interpolation (e), which can then be triangulated to reveal the object surface (f).



(a) Local centroid based



(b) Disc correspondence based

Figure 4: **Calculation of interpolation direction.** 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.

3.2.4 Disc correspondence

Figure 4 shows the calculation of interpolation direction for the previous and new algorithms. In (a), 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 neighbouring 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 Figure 4(b).

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

$$\vec{c}_a = \begin{cases} \frac{1}{\sum\limits_{b \in \text{discs}} \omega_b} \sum\limits_{b \in \text{discs}} \omega_b \vec{l}_{ab} & \text{if } \sum\limits_{b \in \text{discs}} \omega_b > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\vec{c}_a$ is the correspondence vector for disc a, $\vec{l}_{ab}$ is the vector from the centre of disc a to disc b, and ω_b is a weighting indicating the likelihood of correspondence of discs a and b, which are from cross-sections on neighbouring 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 discs is the most important step in this process. It is estimated by comparing, in each plane, the difference in distance transform values at the centre of each disc with the planar distance between the disc centres. This gives an error which, if it is small compared to 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:

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

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

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 centre 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 centre of disc a is affected by disc b. r_a and r_b are the radii of the discs, and d_a and d_b are the distance field values at the projection of each disc on the opposite plane. The function chamfer $\left(\vec{l}_{ab}\right)$ is an estimate of $\left|\vec{l}_{ab}\right|$, measured along a path formed entirely from lines parallel or at 45^o to the image edges. This is the same estimate as used in the distance transformation of the cross-sections [3].

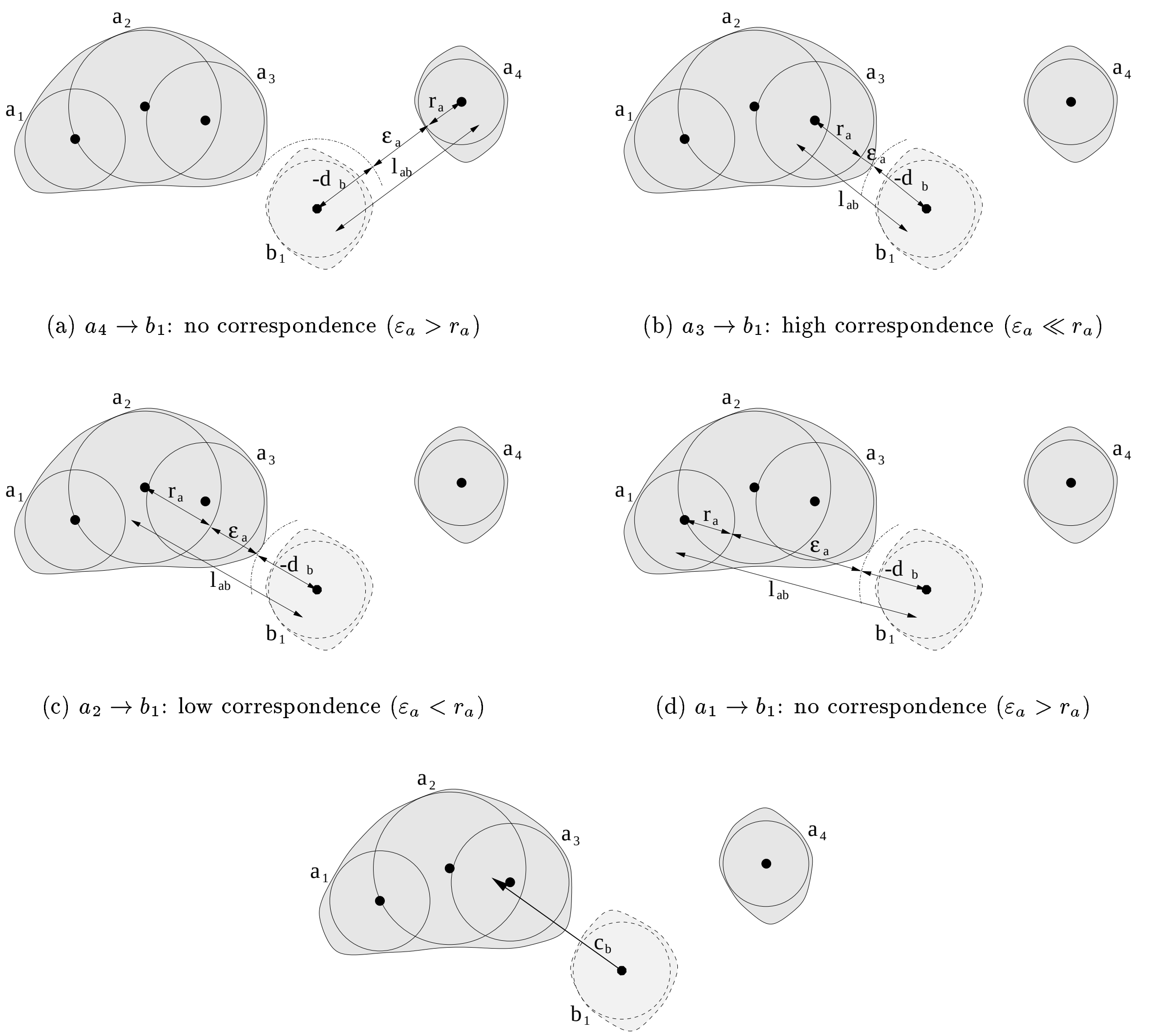
This way of weighting the contributions from each disc has a variety of features which make it attractive:

- Discs will only correspond with the nearest discs on the neighbouring 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 neighbouring plane, i.e. at least one object must span the cross-sections.

Figure 5 shows an example of the calculation of coverage error, given in equation (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 centre of the projected disc. This is equally the case for discs from other contours, as in Figure 5(a), or from the same contour, as in Figure 5(d). The criteria for there to be any correspondence at all that $\varepsilon_a < r_a$, 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 the disc in Figure 5(b) has a lower error than that in Figure 5(c). This leads to a correspondence vector which is closer to the disc with the lower error, as in Figure 5(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 Figure 6 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, as in Figure 6(a).

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 Figure 3(a) is shown in Figure 3(d) — note that in this case the external discs have no partners on neighbouring planes, and hence do not contribute to the correspondence estimate.



(a) $a_4 \to b_1$: no correspondence ($\varepsilon_a > r_a$)

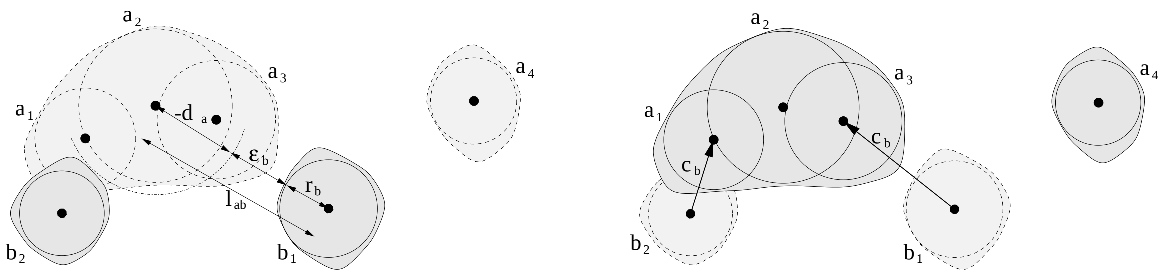
(b) $a_3 \to b_1$: high correspondence ($\varepsilon_a \ll r_a$)

(c) $a_2 \to b_1$: low correspondence ($\varepsilon_a < r_a$)

(d) $a_1 \to b_1$: no correspondence ($\varepsilon_a > r_a$)

(e) Resulting correspondence direction $\vec{c}_b$

Figure 5: **Effect of $\varepsilon_\mathbf{a}$ on correspondence.** Discs $a_1 \dots a_4$ (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) to (d) show 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. The resulting correspondence vector for disc b_1 is shown in (e).



(a) $b_1 \to a_2$: no correspondence ($\varepsilon_b < r_b$) (b) Resulting correspondence direction $\vec{c}_b$

Figure 6: **Additional effect of $\varepsilon_\mathbf{b}$ on correspondence.** An additional contour and representative disc b_2 has been added to those in Figure 5. This has no effect on the values of ε_a, but it changes those for ε_b, such that there is no longer any correspondence between disc a_2 and b_1, as shown in (a). The resulting correspondence vector for discs b_1 and b_2 is shown in (b).

3.2.5 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:

$$\vec{c}_p = \begin{cases} \frac{1}{\sum\limits_{a\in\mathrm{int}} \omega_a} \sum\limits_{a\in\mathrm{int}} \omega_a \vec{c}_a & \text{if } \sum\limits_{a\in\mathrm{ext}} \omega_a = 0 \\ \frac{1}{2(l_{int}+d_p)\left(\sum\limits_{a\in\mathrm{int}} \omega_a + \sum\limits_{a\in\mathrm{ext}} \omega_a\right)} \left[(l_{int} + 2d_p) \sum\limits_{a\in\mathrm{int}} \omega_a \vec{c}_a + l_{int} \sum\limits_{a\in\mathrm{ext}} \omega_a \vec{c}_a\right] & \text{else if } d_p > 0 \\ \frac{1}{2(l_{ext}-d_p)\left(\sum\limits_{a\in\mathrm{int}} \omega_a + \sum\limits_{a\in\mathrm{ext}} \omega_a\right)} \left[(l_{ext} - 2d_p) \sum\limits_{a\in\mathrm{int}} \omega_a \vec{c}_a + l_{ext} \sum\limits_{a\in\mathrm{ext}} \omega_a \vec{c}_a\right] & \text{otherwise} \end{cases} \tag{4}$$

where *int* implies internal discs, *ext* implies external discs, and d_p is the distance field value at the point p.

There are three separate cases in equation (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 equation (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 equation (4):

$$l_{int} = \frac{1}{\sum\limits_{a \in \mathrm{int}} \omega_a} \sum_{a \in \mathrm{int}} \omega_a l_a, \quad l_{ext} = \frac{1}{\sum\limits_{a \in \mathrm{ext}} \omega_a} \sum_{a \in \mathrm{ext}} \omega_a l_a \tag{5}$$

The effect of these values in equation (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 correspondence is evenly weighted.

The weighting ω_a and coverage error ε_a in equations (4) and (5) are similar to those in equations (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 centre of the projected disc:

$$\omega_a = \frac{1}{{\varepsilon_a}^2 + \mu} \tag{6}$$

$$\varepsilon_a = \mathrm{chamfer}\left(\vec{l}_{ap}\right) - |r_a - d_p| \tag{7}$$

The final correspondence for a point *between* neighbouring 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.

3.2.6 Interpolation and iso-surface extraction

The point correspondence $\vec{c}_p$ is a vector lying within the plane of one of the neighbouring 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 non-parallel planes, although care is needed to constrain the eventual interpolation direction [25].

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 neighbouring distance transforms in this direction. We use a body-centred 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 [24]. The resolution of the interpolated surface can be varied by varying the resolution of the sampling lattice.

4 Results

Surfaces were estimated by four variants of shape-based interpolation:

Shape-based interpolation The original algorithm [8] was implemented, save that for non-parallel planes the interpolation direction was taken to be the average planar normal to those planes.

Centroid-guided This was a combination of techniques described in [9] 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 each were the same size. Correspondence of contours was determined manually (rather than simply by detecting object overlap). Each of the generated surfaces were then binary or'ed to create the final object surface. The method of edge-shrinking interpolation [14] was not implemented since it is much more complicated, and in practice gives very similar results to the centroid-guided technique. The results will be *slightly* better, due to the more sophisticated alignment of corresponding contours, but the same gross artifacts are present.

Previous disc-guided This is the technique described in our previous paper [25].

New disc-guided The technique described here, which includes external discs, and a better correspondence and interpolation method than the previous algorithm.

All the surfaces were generated by triangulating the zero iso-surface of the interpolated distance fields, using regularised marching tetrahedra [24]. These triangulations were displayed with Gouraud shading using the Geomview[2] visualisation package. Processing was done on a Silicon Graphics Indigo 2 workstation[3] with 256Mb RAM, and a 166MHz Pentium MMX with 32Mb RAM running Linux, with similar performance on each platform.

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.

4.1 Freehand 3-D ultrasound

In vivo ultrasound data was recorded using a Toshiba Powervision 7000[4] ultrasound machine with a 3.75MHz convex curvilinear array probe. A Polhemus FASTRAK[5] magnetic field position sensor was mounted on this probe. The position signal from this, in addition to the video output of the ultrasound machine, was fed to a Silicon Graphics Indy workstation with a digital acquisition card. Acquisition of the ultrasound images and position readings, calibration of the system, and manual segmentation of the data were all performed using Stradx v5.3[6] software [20].

The bladder and prostate in Figure 7 demonstrate the effect of interpolating two close bodies. Although both are simple, the contours overlap, and hence shape-based interpolation and the previous disc-guided interpolation make a connection between the bodies. Centroid-guided interpolation is not affected, since each of the bodies are treated individually (although they would still be connected if object correspondence was based on contour overlap). The new disc-guided interpolation correctly constructs the surface, since the external discs ensure the algorithm also 'connects' the gap between each of the objects.

The foetus in Figure 8 is a much more complicated shape than the bladder or prostate. 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

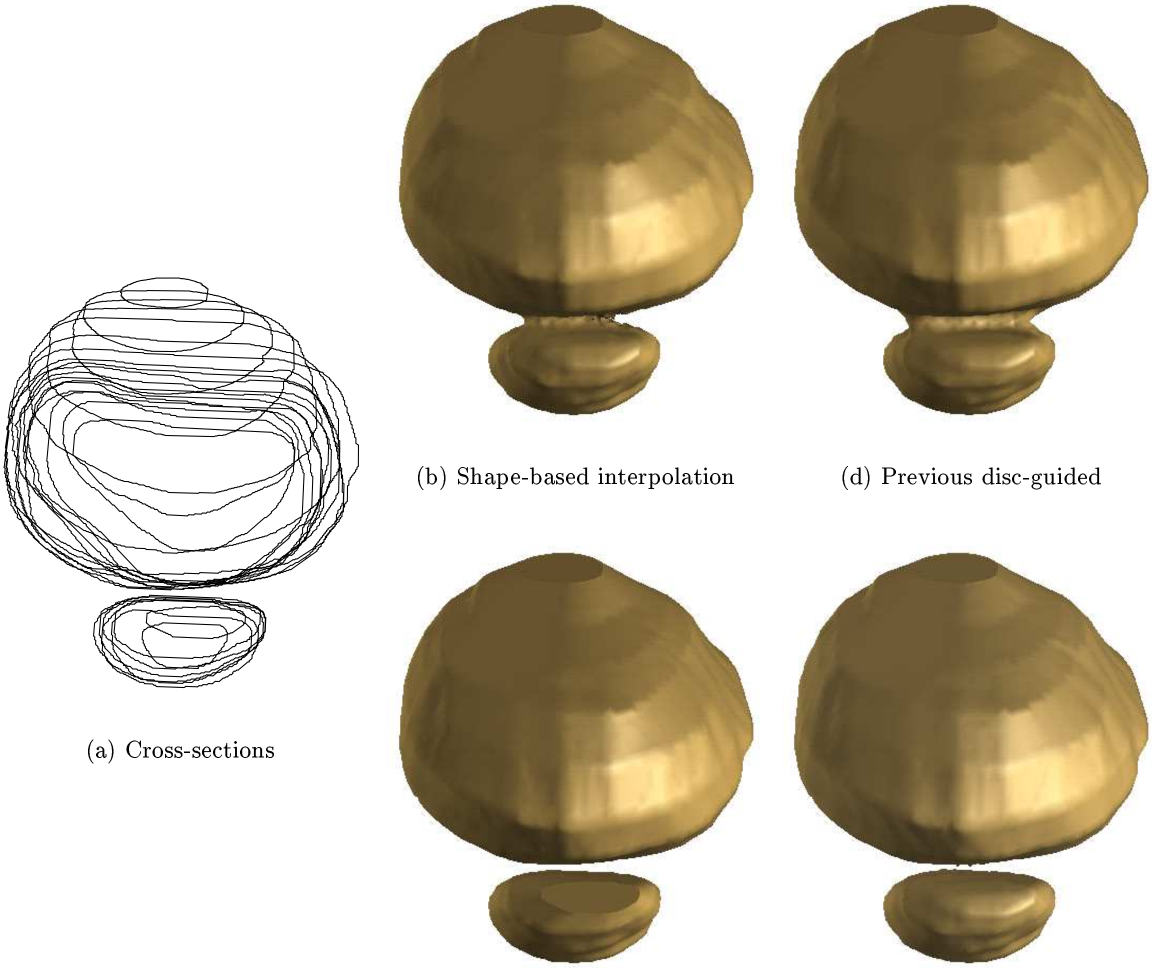
[2] `http://www.geom.umn.edu/software/geomview/.`

[3] Silicon Graphics Incorporated, Mountain View, California

[4] Toshiba America Medical Systems, Tustin, California

[5] Polhemus Incorporated, Colchester, Vermont

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



(a) Cross-sections

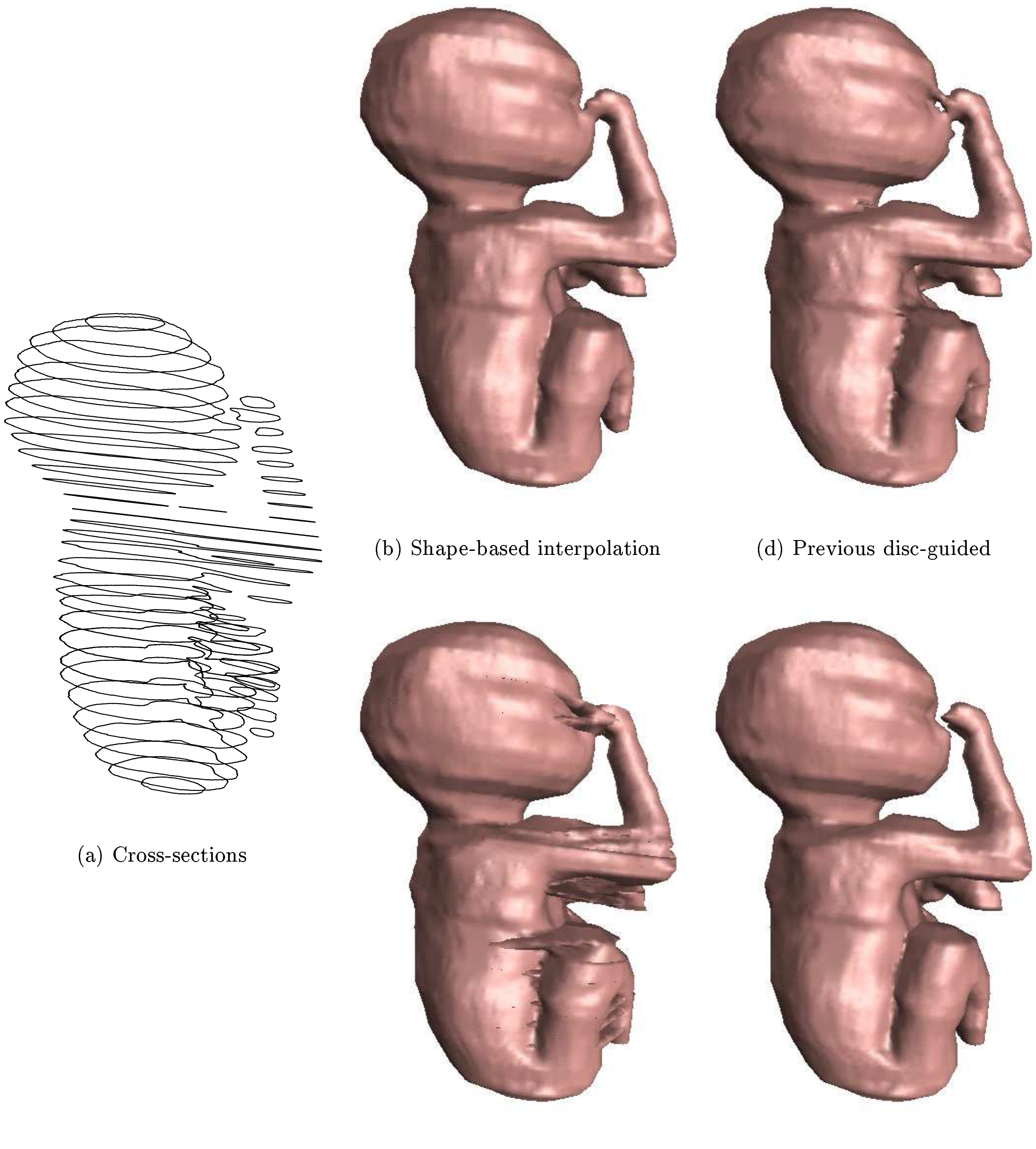
(b) Shape-based interpolation

(d) Previous disc-guided

(c) Centroid-guided

(e) New disc-guided

Figure 7: **Human bladder and prostate.** The cross-sections were manually outlined in the original B-scans of a freehand 3-D ultrasound investigation.



(a) Cross-sections

(b) Shape-based interpolation

(d) Previous disc-guided

(c) Centroid-guided

(e) New disc-guided

Figure 8: **Foetus at week 22.** The cross-sections were manually outlined in the original B-scans of a freehand 3-D ultrasound investigation.

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.

4.2 X-ray Computed Tomography

The child's skull in Figure 9 is from data provided with the 3DViewnix[7] visualisation package [26]. Here, a high resolution ($0.41\text{mm} \times 0.41\text{mm} \times 1mm$) CT scan has been thresholded with the appropriate coefficient for bone, but only every sixth cross-section has been used in the reconstruction. The female pelvis in Figure 10 is from CT data provided by the Visible Human Project[8]. In this case the original CT resolution was $0.9375\text{mm} \times 0.9375\text{mm} \times 1\text{mm}$, however only every tenth cross-section has been used in reconstructing the surface.

The child's skull in Figure 9 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 has not been applied in this case, since there are multiple contours which are neither simple nor tree-like. Liu's method [14] 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 as a result 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.

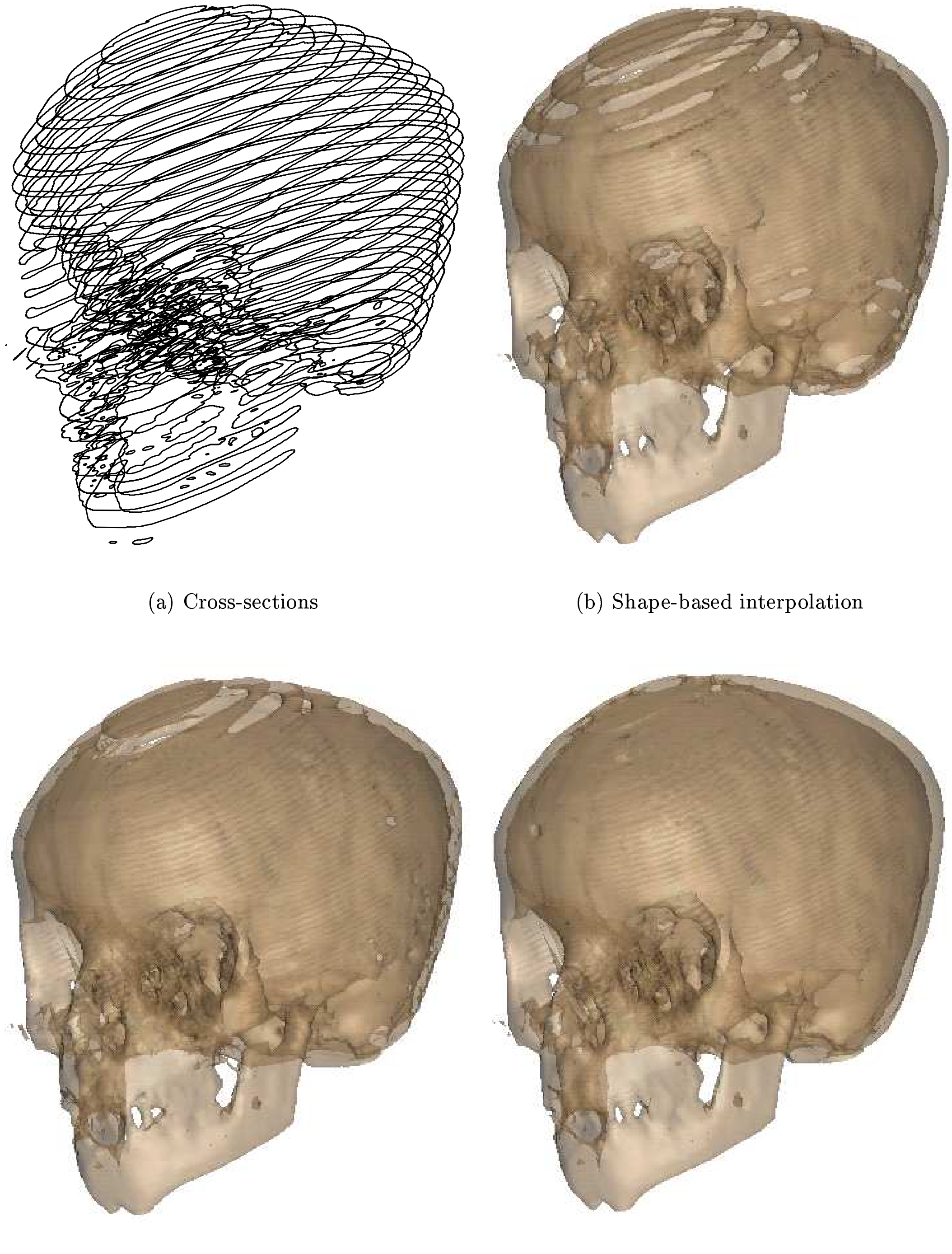
The previous disc-guided method improves on this result, but it is still limited by the condition that the discs representing each cross-section must overlap in order for them to correspond. There is no such constraint in the new disc-guided interpolation, and hence the skull can still be correctly reconstructed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

The female pelvis in Figure 10 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 Figure 10(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. Figure 10(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.

4.3 Magnetic Resonance Imaging

The human liver in Figure 11 is from MRI data provided by the Visible Human Project, in this case from the male data set, the original resolution being $1.875\text{mm} \times 1.875\text{mm} \times 4\text{mm}$. Five cross-sections were manually segmented from this data. This is enough information to give a reasonable idea of the shape of the liver and estimate of its volume. However, it is not enough information for shape-based interpolation to estimate the surface — hence the large invagination in Figure 11(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

[7]From `CHILD.IMO`, 3DViewnix v1.1.1 (c) 1993-1996 M I P G University of Pennsylvania, All Rights Reserved.
[8]The Visible Human Project is an initiative from the National Library of Medicine in Bethesda, Maryland.



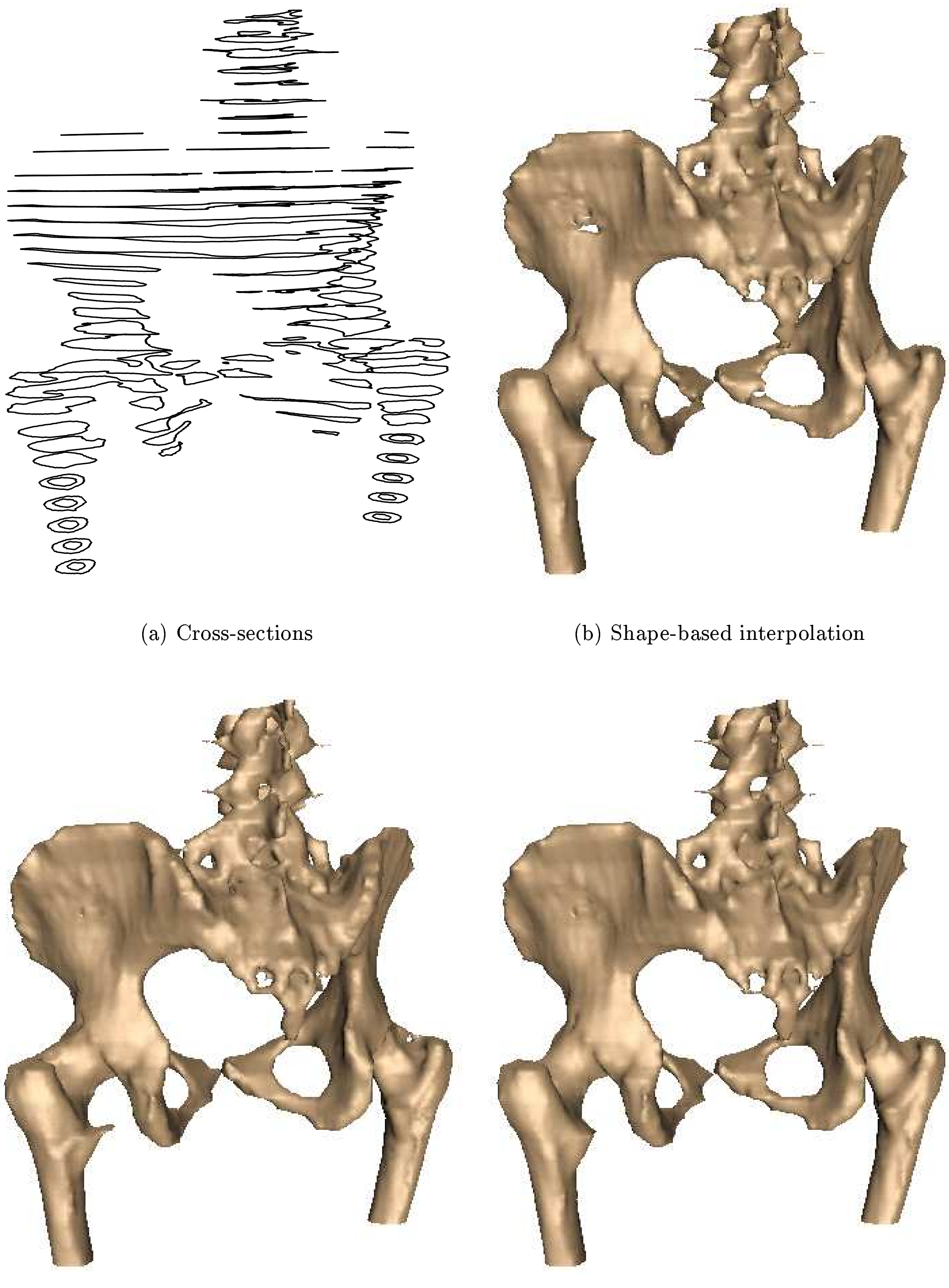
(a) Cross-sections

(b) Shape-based interpolation

(c) Previous disc-guided interpolation

(d) New disc-guided interpolation

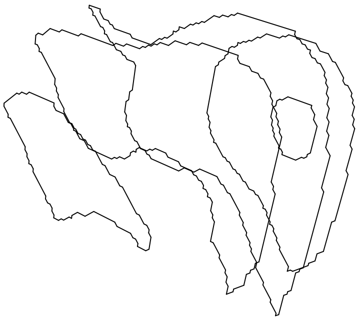
Figure 9: **Child's skull.** The cross-sections were automatically segmented by thresholding CT data. The surface is rendered with a slight transparency, so that both the inner and outer surfaces can be seen.



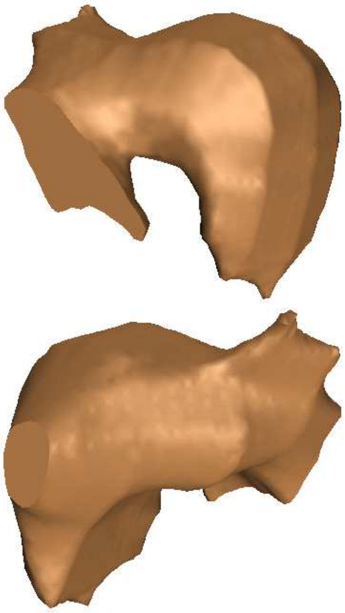
(a) Cross-sections (b) Shape-based interpolation

(c) Previous disc-guided interpolation (d) New disc-guided interpolation

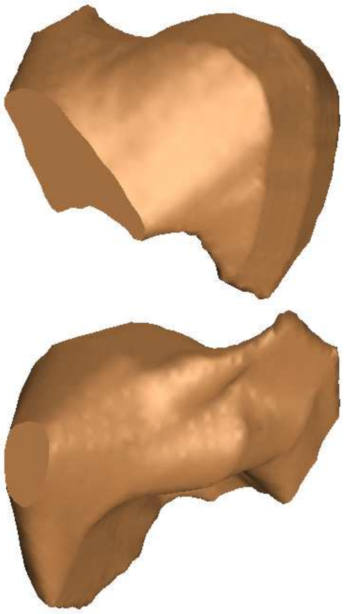
Figure 10: **Female pelvis.** The cross-sections were semi-automatically segmented by thresholding CT data, then manually editing the results of this threshold operation.



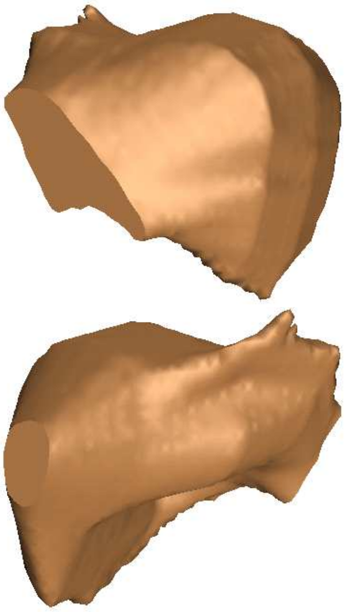
(a) Cross-sections



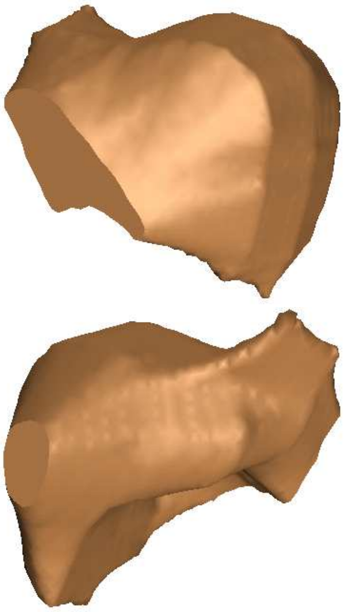
(b) Shape-based interpolation



(d) Previous disc-guided



(c) Centroid-guided



(e) New disc-guided

Figure 11: **Human liver.** The cross-sections were manually outlined from coronal MRI slices of the abdomen.

Object	Modality	Triangles	Processing Time /secs			
			Shape-based	Centroids	Prev. disc	New disc
Bladder/Prostate	Ultrasound	13,000	2.3	2.3	3.0	4.0
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

Table 1: **Processing time.** These include all operations after creation of the cross-sections, up to and including display of the triangulated surface. Times were recorded on a Silicon Graphics Indigo 2 workstation, however they were comparable to a 166MHz Pentium MMX running Linux.

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 re-adjust the interpolation direction in the region of the feature.

4.4 Processing time

Processing times for each of the surfaces in Figures 7 to 11 are shown in Table 1. The software was designed for non-parallel 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 Figure 9) 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.

5 Conclusions

The method of surface interpolation presented in this report 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 does introduce 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 to 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.

Acknowledgements

Graham Treece is supported by an EPSRC studentship, and a Newton Trust award from the University of Cambridge Department of Engineering.

References

[1] W. A. Barrett and E. N. Mortensen. Fast, accurate, and reproducible live-wire boundary extraction. *Lecture Notes in Computer Science*, 1131:183–192, 1996.

[2] J.-D. Boissonnat. Shape reconstruction from planar cross sections. *Computer Vision, Graphics, and Image Processing*, 44:1–29, 1988.

[3] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.

[4] 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 Transactions on Medical Imaging*, 9(1):71–83, 1990.

[5] D. Cohen-Or, D. Levin, and A. Solomovici. Three-dimensional distance field metamorphosis. *ACM Transactions on Graphics*, 17(2), April 1998.

[6] C. R. Dance. *Computing Models from 3D Ultrasound*. PhD thesis, University of Cambridge, 1996.

[7] T. D. Haig, Y. Attikiouzel, and M. Alder. Border marriage: matching of contours of serial sections. *IEE Proceedings*, 138(5):371–376, October 1991.

[8] G. T. Herman, J. Zheng, and C. A. Bucholtz. Shape-based interpolation. *IEEE Computer Graphics and Applications*, pages 69–79, May 1992.

[9] W. E. Higgins, C. Morice, and E. L. Ritman. Shape-based interpolation of tree-like structures in three-dimensional images. *IEEE Transactions on Medical Imaging*, 12(3):439–450, 1993.

[10] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, July 1992.

[11] M. Kass, A. Witkin, and D. Terzopoulos. Snakes — active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[12] 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 echocariography of the left ventricle based on a magnetic-field position and orientation sensing system. *IEEE Transactions on Biomedical Engineering*, 45(4):494–504, April 1998.

[13] D. Levin. Multidimensional reconstruction by set-valued approximations. *IMA Journal of Numerical Analysis*, 6:173–184, 1986.

[14] Y.-H. Liu, Y.-N. Sun, C.-W. Mao, and C.-J. Lin. Edge-shrinking interpolation for medical images. *Computerized Medical Imaging and Graphics*, 21(2):91–101, 1997.

[15] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–168, July 1987.

[16] M. Loughlin, I. Carlbom, C. Busch, T. Douglas, L. Egevad, H. Frimmel, M. Norberg, I. Sesterhenn, and J. M. Frogge. Three-dimensional modelling of biopsy protocols for localized prostate cancer. *Computerized Medical Imaging and Graphics*, 22:229–238, 1998.

[17] J-M Oliva, M. Perrin, and S. Coquillart. 3D reconstruction of complex polyhedral shapes from contours using a simplified generalized Voronöi diagram. *Computer Graphics Forum*, 15(3):C–397–C–408, 1996.

[18] B. A. Payne and A. W. Toga. Distance field manipulation of surface models. *IEEE Computer Graphics and Applications*, pages 65–71, January 1992.

[19] B. A. Payne and A. W. Toga. Surface reconstruction by multiaxial triangulation. *IEEE Computer Graphics and Applications*, 14(6):28–35, November 1994.

[20] R. W. Prager, A. H. Gee, and L. Berman. Stradx: real-time acquisition and visualisation of freehand 3D ultrasound. *Medical Image Analysis*, 1999. In press.

[21] S. P. Raya and J. K. Udupa. Shape-based interpolation of multidimensional objects. *IEEE Transactions on Medical Imaging*, 9(1):32–42, 1990.

[22] V. V. Savchenko, A. A. Pasko, O. G. Okunev, and T. L. Kunii. Function representation of solids reconstructed from scattered surface points and contours. *Computer Graphics Forum*, 14(4):181–188, 1995.

[23] W.-S. V. Shih, W.-C. Lin, and C.-T. Chen. Morphologic field morphing: Contour model-guided image interpolation. *International Journal of Imaging Systems and Technology*, 8:480–490, 1997.

[24] G. M. Treece, R. W. Prager, and A. H. Gee. Regularised marching tetrahedra: improved iso-surface extraction. Technical Report CUED/F-INFENG/TR 333, Cambridge University Engineering Dept, September 1998.

[25] G. M. Treece, R. W. Prager, A. H. Gee, and L. Berman. Fast surface and volume estimation from non-parallel cross-sections, for freehand 3-D ultrasound. *Medical Image Analysis*, 1999. In press.

[26] 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. In *SPIE Proceedings*, volume 2164, pages 58–73, 1994.