Fast surface and volume estimation from non-parallel cross-sections, for freehand 3-D ultrasound

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

CUED/F-INFENG/TR 326

July 1998

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

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

Abstract

Volume measurements from ultrasound B-scans are useful in many clinical areas. It has previously been demonstrated that using 3-D ultrasound can greatly increase the accuracy of these measurements. *Freehand* 3-D ultrasound allows freedom of movement in scanning, but the processing is complicated by having non-parallel scan planes. Two techniques are proposed for volume measurement from such data, which also improve surface and volume estimation from data acquired on parallel planes. *Cubic planimetry* is a more accurate extension of a volume measurement technique involving vector areas and centroids of cross-sections. *Maximal disc shape based interpolation* is an extension of shape based interpolation which uses maximal disc representations to locally adjust the interpolation direction and hence improve the quality of the generated surface. Both methods are tested in simulation and on *in-vivo* data. Volumes estimated using cubic planimetry are more accurate than step-section planimetry, and require fewer cross-sections, even for complex objects. Maximal disc shape based interpolation is ideally suited for reconstructing surfaces from a handful of cross-sections, and can therefore be used to give confidence in the segmentation and hence also the cubic planimetry volume.

Contents

1 Motivation

In the last two decades, a great many researchers have attempted to produce systems which will allow the construction and visualisation of three dimensional (3-D) images from ultrasound data. There is general agreement that this development represents a positive step forward in medical imaging, and clinical applications have been suggested in many different areas. However, it is clear that 3-D ultrasound has not yet gained widespread clinical acceptance, and that there are still important problems to solve before this becomes a common tool.

1.1 Advantages of 3-D ultrasound over other imaging modalities

In order to gain acceptance, 3-D ultrasound must have clear advantages over the other imaging modalities, e.g. computed x-ray tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET) or conventional B-mode and Doppler 2-D ultrasound. The main advantages which have been suggested for 3-D ultrasound can be grouped into five categories :

- Ultrasound is a real time imaging modality, and 3-D ultrasound has the potential for displaying information in near real time too. This is currently limited more by the processing than the acoustics, which is the ultimate limit of speed. High rates of acquisition (between 10-60 frames per second) [67] can allow imaging within a single breath hold, greatly increasing modelling accuracy in organs which move with inspiration and expiration. Shorter examination times are also of benefit to both the clinician and the patient [21, 23].

- Extension of ultrasound to 3-D provides new images which would be impossible to visualise otherwise, or at least could only be visualised as the clinician builds up a mental picture from 2-D information [54]. In effect, the computer can assist the clinician by performing the reconstruction for them. This may make the modality more accessible to those less experienced in analysing ultrasound images. Perhaps more importantly, a 3-D image may provide a better point of reference for discussing diagnosis than a conventional 2-D hard copy [67]. It would also provide a better means of documentation of the examination for clinical record, or in the case of a foetus for the parents' information too [74]. Ultrasound is in some ways inherently suited to 3-D reconstruction, since the images are tomographic [23, 67]. The acquisition of an entire volume also allows the clinician to view 2-D ultrasound images in planes which would not normally be possible due to the physical restrictions of the scanning process, for instance parallel to the skin surface.

- The reconstruction of 3-D ultrasound by computer potentially brings greater standardisation and repeatability to conventional examinations [67], which are otherwise quite subjective [23]. It also allows for more accurate assessments of volumes due to the use of non-geometric methods rather than the conventional ellipsoid-based formulae [1]. 3-D ultrasound also offers the potential for improving the quality of even a 2-D image, by compounding images acquired from different perspectives [6, 69].

- There are some practical reasons for using ultrasound rather than CT, MRI or PET. Existing ultrasound machines are less expensive, and require less specialised facilities [54].

They can also be upgraded to 3-D surprisingly cheaply [67]. Also, ultrasound scanning does not involve the use of ionising radiation.

- Finally, in some cases 3-D ultrasound may be able to replace other, more invasive, procedures, e.g. in the monitoring of ventricular function in critically ill patients during surgery [48].

1.2 Clinical uses of 3-D ultrasound

3-D ultrasound has been applied to several clinical areas, for example:

Heart Much of the work involving 3-D (and 4-D) ultrasound has been concerned with echocardiography of the heart. For reviews of this wealth of literature see, for example, Moritz [51] or Salustri [71]. Much of this work has been driven by the desire to understand and be able to visualise the structure of the heart and how it changes throughout the cardiac cycle. Most clinical applications relate to the measurement of ventricular volume, from which ejection fraction can be derived, for instance in ischemic and congenital heart disease [23, 58, 71]. Volumetry of the ventricles by 3-D ultrasound has been shown to be accurate even (and perhaps especially) for hearts of abnormal size and geometry [29]. Increased left ventricular mass has also been established as a predictor of increased cardiac morbidity and mortality independent of age and blood pressure [29]. Accurate measurements of atrial diameter can also be made [46]. In addition, 3-D visualisation may reduce invasive monitoring for managing critically ill patients during surgery [48].

Carotid Artery Many researchers have suggested that 3-D visualisation of the carotid artery can help in the accurate definition of the size and features of atherosclerotic plaques [23, 70]. For instance, it can generate simultaneous displays of both longitudinal and circumferential distribution of the plaque, which is impossible with 2-D images. However, Rankin [67] suggests that clinical applications for the popular 3-D reconstruction of the carotid bifurcation have yet to be defined.

Vascular Anatomy Serial surveillance of lower extremity saphenous vein bypass grafts is of benefit for the detection of haemodynamically significant myointimal hyperplasia lesions [33]. Geometric measurements of luminal change with 3-D ultrasound may allow detection prior to the development of haemodynamic disturbances.

Brain 3-D ultrasound has been suggested for the localisation of tumours and arteriovenous malformations in the brain, in order to accurately guide surgeons [67].

Eye Most of the work on the eye has concentrated on the evaluation of choroidal melanomas. The 3-D display shows the tumour in relation to the globe and optic nerve, which can be significant for surgical planning, and allows volume measurement, which is important as tumour size is significantly correlated to survival rate [37]. Accurate volume measurement of melanomas has also been used for follow up of therapeutic response [67]. 3-D ultrasound display has been suggested to be better than conventional 2-D for the evaluation of vitreous haemorrhages and retinal detachment [21].

Breast Some early work by Lalouche [43] indicates that the hypoechoic areas, which are difficult to distinguish in 2-D, organise into a visible ductular system in 3-D. There is also a higher accuracy of diagnosis of benign cystic disease with 3-D ultrasound than with X-ray mammography [23, 24].

Foetus Like the heart, the foetus has had much attention in this area. Reviews of foetal applications of 3-D ultrasound are given by Steiner [74] and to a lesser extent Merz [49]. Much of this work has been concerned with the detection of foetal abnormalities, for instance cleft lip or other malformations of the face [44]. The visualisation of the foetal spine and thorax has also been achieved, important in evaluating skeletal dysplasia, abnormalities leading to a small thorax or subsequent pulmonary hypoplasia, and neural tube defects [55]. Foetal volume and weight have been estimated and used to evaluate intrauterine growth retardation [67]. Attempts at foetal echocardiography using 3-D ultrasound have also been reported [20].

Placenta Placental volume around mid-pregnancy has been measured using 3-D ultrasound, to investigate how placental volume compared to birth weight affects susceptibility to heart disease in later life [34].

Uterus Reconstruction of transvaginal images of the uterine cavity has been performed for accurate visualisation prior to surgical treatment of fibroids and endometrial polyps [5]. 3-D transvaginal ultrasound has also been used to control intrauterine device insertion [10]. In this case, abnormal IUD insertions were identified as accurately and precisely as would have been done by hysteroscopy.

Kidney The size and discrete anatomy of the kidney make 3-D imaging of this organ attractive. Possible clinical uses include intrarenal neoplasia and renal transplants, where accurate assessment of renal volume and its change over time may be of help in defining rejection [67]. Volume assessment by 3-D ultrasound has been compared favourably to that performed by MRI [28].

Liver Reconstruction of the liver with 3-D ultrasound has been severely limited, as it is large and is sheltered behind the lower ribs. It also moves considerably with respiration and the left lobe moves with cardiac motion [23]. As a result, clinical application has yet to be defined, but 3-D ultrasound should have an advantage over other modalities since it is inherently fast [67].

Prostate There have been a great many successful attempts to measure the volume of the prostate using 3-D ultrasound [7, 53]. Accurate assessment of the size of the prostate gland at regular intervals is essential to clinical studies of drug therapy for reducing prostate volume [2]. Tumour volume is of great prognostic significance in prostatic cancer, and its measurement may be more accurate with this technique [24]. A precise estimate of the enlargement due to benign prostatic hyperplasia helps to determine the appropriate therapy [75].

Bladder Incomplete voiding is an important sign of many urologic disorders. This is conventionally measured by catheterization, but ultrasound estimates of the volume are potentially more accurate and reliable [47].

Urethra 3D images of the urethra generated from ultrasound scans provide more information than conventional voiding urethrograms [56].

Baboon prostates 3-D ultrasound has also been used for the measurement of baboon prostates in order to better understand the pathogenic mechanisms of prostate cancer. Rather than using "invasive surgery or terminal experiments to excise the prostate", this method is minimally traumatic [38].

1.3 3-D ultrasound systems

A good review of the various types of 3-D ultrasound systems is given by Fenster and Downey [23], or for echocardiography by Salustri [71]. They suggest that systems vary in three main areas — acquisition, reconstruction and display. In fact, reconstruction can be performed as an integral part of the display process, rather than a separate step prior to display, and it is easier to deal with these two together.

1.3.1 Acquisition techniques

Acquisition techniques can be classified either by scan pattern or positioning equipment. As can be seen from Figure 1, a variety of scan patterns can be achieved with most techniques. Freehand scanning is classified separately since, although it can be used to give approximate linear, fan or rotational scans, the relative plane orientation will not be as precise. A brief summary of the reported systems follows :

Position Estimation by Eye Some earlier systems (and some more recent ones) relied on estimation of the position of the scan planes by eye. This requires extreme care when scanning, as it is imperative that the movement of the transducer is smooth and constant. Any jitter introduced at this stage is carried through directly to the reconstruction. Examples of linear sweeps include [5] and [70], where in both cases the slice separation was assumed to be uniform. Rotational scans have also been performed using this technique [34], or by estimating the angle at each frame recorded individually [25]. While these methods are simple, they clearly leave much to be desired if accurate reconstructions are required.

Manual Position Measurement A more reliable technique is manual position measurement. This is only possible if each frame is acquired individually (therefore giving time to measure the position of said frame). Examples include a catheter mounted on a ratcheted device, giving a measurement of drawback [53, 75] and rotational scanning with the angle measured from a scale attached to a fixed mechanical arm and the transducer [18]. Clearly this method precludes fast acquisition, one of the potential benefits of 3-D ultrasound.

Automatic Position Sensing An alternative to measuring the position is to sense it remotely by some mechanism. This potentially allows complete freedom of movement of the transducer, and hence results in arbitrarily orientated scan planes. There are four reported ways of achieving this, namely acoustical, optical, magnetic and electrical (mechanical arm).

The acoustical method works by attaching an emitter to the transducer and picking up these signals with three remote microphones, positioned in different orientations (not necessarily orthogonal). An acoustic spark gap is generally used for the emitter as it can produce high frequencies (hence giving good resolution) and is also a much

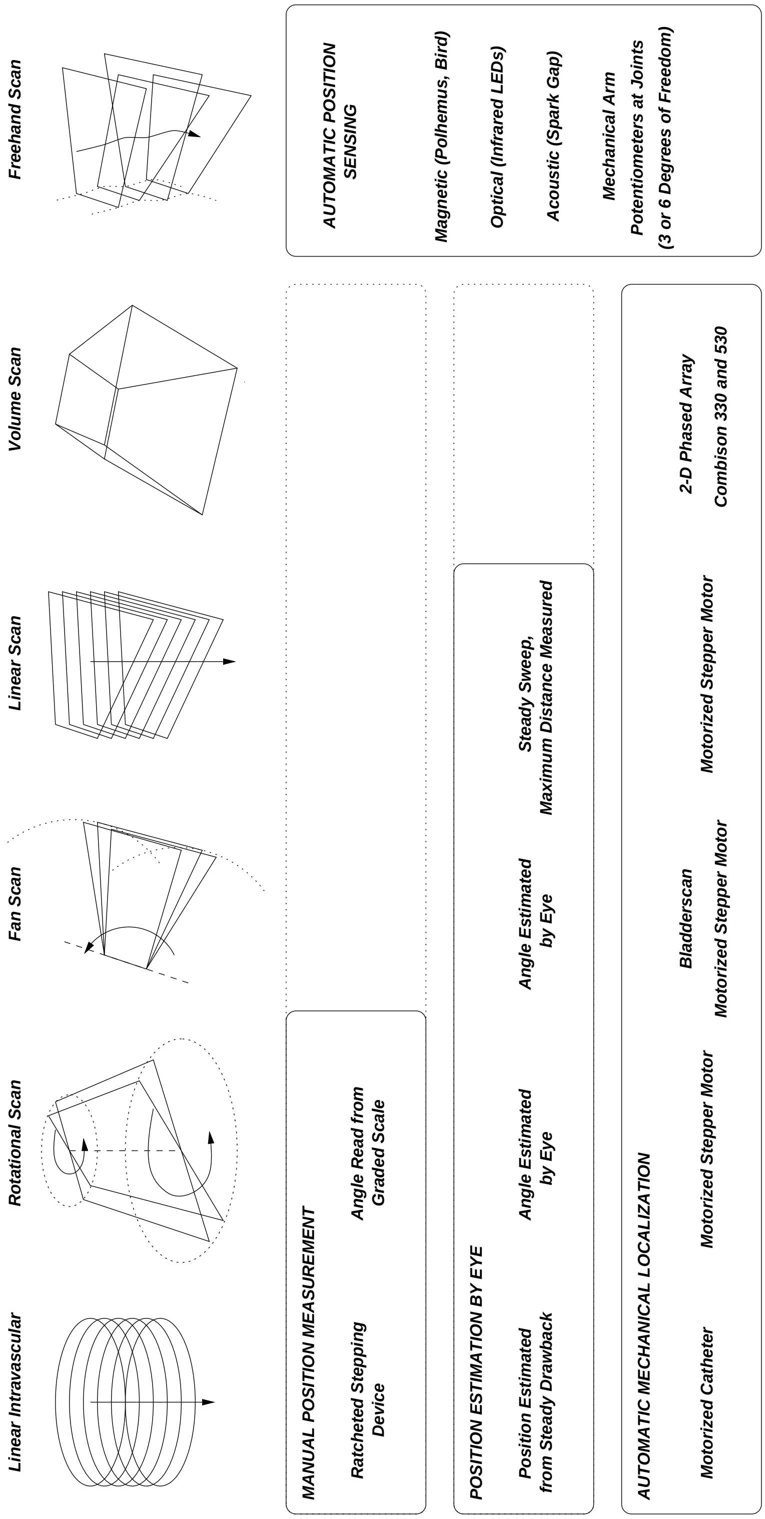


Figure 1: **3-D Ultrasound Systems: Scan pattern and position measurement.**

smaller source than, for instance, a loudspeaker. The system was originally devised by Moritz [51] and is still used in, for example, [3, 29, 41].

The optical ('stereotactic') method works in a similar manner, except the emitter is replaced by at least three infrared LEDs [76]. A restriction of this technique is that there must be a line of sight at all times between the LEDs and the optical sensor.

Remote sensing can also be achieved using a magnetic field. In this case the receiver is mounted on the transducer and the transmitter, which generates the magnetic field, is remote [6, 20, 33, 35, 36, 54, 55, 56]. The magnetic field is affected by the presence of ferrous materials in the vicinity.

In the final approach, the transducer is attached to a mechanical arm with either one [48], three [60] or six [23] degrees of freedom. Potentiometers are mounted at the arm joints, and the electrical signals from these can be used to calculate the position. In the one degree of freedom case (used for intravascular scanning of the heart) the scanning pattern is similar to a fan, but the angle between each scan need not be constant.

Automatic Mechanical Localisation A completely different approach to measuring position is to determine it in advance by mechanical localisation. This is generally achieved by attaching the transducer to a stepper motor, for either linear [38, 43, 46, 72], rotational [8, 21, 37, 58], fan [8, 28] or intravascular linear [42] scanning. This kind of approach has also been used in integrated transducers, for instance Kretz Technik's Combison 330 [1, 2] and 530 [10, 44, 74, 49], which scans an entire pyramidal volume. An integrated scanner and volume calculation device, Diagnostic Ultrasound Corporation's Bladderscan, also exists for the scanning of bladders [47]. All these devices impose limitations on the maximum swept volume, since this is constrained by the mechanics.

In a similar category, two-dimensional phased arrays are also being investigated [67, 71]. These may represent the future of 3-D ultrasound, however before they become practical a number of problems must be overcome, for instance low yields due to the large number of small elements [23], and they still constrain the maximum swept volume.

Both the freehand and mechanical acquisition techniques have clear advantages. The mechanical systems produce a regular data set which simplifies later processing and ensures uniform coverage of the scanned volume. On the other hand, it is impossible to scan volumes which are larger than the mechanics allow for (the coordinate reference frame is based on the stepper motor housing, which must not be moved during acquisition). This limitation can not simply be overcome by increasing the size of the system — larger systems are difficult to use in practice, since the transducer must keep a good contact with the surface at all times, but too much pressure moves the underlying anatomy, generating mis-registered data.

The freehand systems have certain advantages over the mechanical systems in that they do not constrain the acquisition volume, and they allow the clinician the freedom to scan in planes which are suitable for the area of interest. However, the lack of regularity in the data set complicates the processing, and hence potentially increases the time from scanning to display. The processing techniques for this situation are a current research area, outlined in the next section.

1.3.2 Interpolation to voxel-based representations

An overview of processing and display techniques, although primarily voxel-based, is given by Nelson [54]. Processing can be helpfully seen as a path from the representation (constrained by the acquisition technique) to the display (desired by the clinician). This is shown in diagrammatic form in Figure 2 (note that for simplicity, other processing steps such as filtering have been omitted). It is clear from this figure that there is often more than one route between these two stages. In addition, each transition generally represents a whole area of research, with many underlying problems and solutions. The selection of processing 'path' has implications for the speed, accuracy, reliability and memory requirements of the eventual system.

The majority of freehand 3-D ultrasound systems use a voxel-based representation: in fact Salustri does not mention any other approaches at all in his survey of echocardiography [71]. The first processing step therefore becomes interpolation of scan-plane based data to a regular 3-D grid. This grid is generally of equal resolution in all three dimensions, since this simplifies the display processing. For this reason, interpolation is required even if the original scans are already parallel, since the scan spacing is usually not the same as the in-plane pixel resolution.

There are many, many ways to perform this interpolation. The techniques employed are often very simple, due to the time required to process typical volumes, which is generally large. In fact, many researchers do not specify the technique they used at all [21, 46, 54, 55]. Generally, some linear method involving a limited number of data points is used [8, 70, 72]. Gilja used trilinear interpolation of the eight neighbouring values from a regular rotational scan [28]. Lalouche used cubic interpolation, but from parallel scans [43].

Interpolation methods for freehand scans include nearest-neighbour bin-filling [76], which is probably the simplest technique — pixel coordinates are transformed to 3-D space, and rounded to the nearest voxel, which is then assigned the original pixel value. Larger 'bins' (i.e. more than one voxel) can be assigned from each pixel in order to reduce any empty space between scans. More sophisticated is closest-points interpolation [76], where each voxel bounded by two scan planes is assigned a distance weighted value based on the closest pixel in each of the two scan planes. An inverse-distance weighting scheme can also be used, which assigns to a voxel the inverse-distance weighted value of all the surrounding pixels in a bounding sphere of some chosen dimension [6]. More sophisticated still, a 3-D convolution of the 2-D slice with a 3-D reconstruction kernel (which was Gaussian in the cited case) can create the 3-D volume incrementally [60].

Voxel based representations are relatively easy to manipulate, but they also have disadvantages. As already mentioned, the time needed to create them is large (typically several minutes with today's computing technology). They also require a lot of memory space, for example a $256 \times 256 \times 256$ voxel representation with 8-bit grey-scale would require 16Mbytes. This is in addition to the memory required to store the original scans, unless these are thrown away after interpolation. Also, the interpolation, however it is done, creates new data values for *all* the voxels, which can introduce artifacts, especially if (as is usually the case with 3-D ultrasound) the interpolation is a simple linear one [31]. These artifacts can then affect the accuracy of segmentation, or lead to misrepresentation of images.

Traditionally, the voxel based representation has been preferred because tools already exist for displaying this data in various ways. Volume-based and re-slice displays in particular are usually only considered from such a representation. However, it has recently been demonstrated that re-slice displays can be constructed in a few seconds directly from scan-

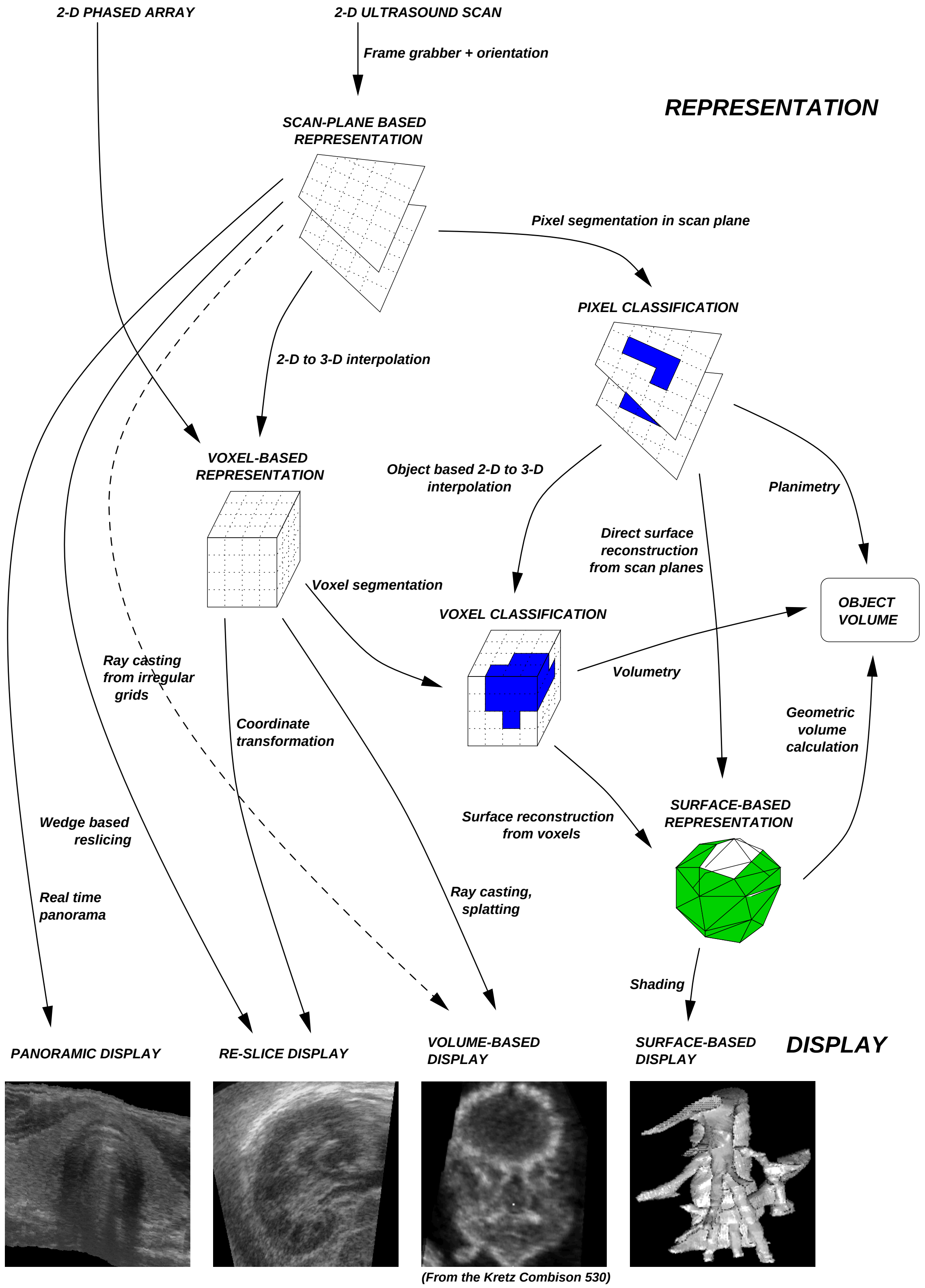

2-D PHASED ARRAY
2-D ULTRASOUND SCAN
Frame grabber + orientation
REPRESENTATION
SCAN-PLANE BASED REPRESENTATION
Pixel segmentation in scan plane
PIXEL CLASSIFICATION
2-D to 3-D interpolation
VOXEL-BASED REPRESENTATION
Object based 2-D to 3-D interpolation
Planimetry
Direct surface reconstruction from scan planes
Voxel segmentation
VOXEL CLASSIFICATION
OBJECT VOLUME
Volumetry
Ray casting from irregular grids
Coordinate transformation
Geometric volume calculation
SURFACE-BASED REPRESENTATION
Surface reconstruction from voxels
Wedge based reslicing
Real time panorama
Ray casting, splatting
Shading
PANORAMIC DISPLAY
RE-SLICE DISPLAY
VOLUME-BASED DISPLAY
SURFACE-BASED DISPLAY
DISPLAY
(From the Kretz Combison 530)

Figure 2: **3-D Ultrasound Systems: Data processing and display.**

plane based data [62]. Even volume-based displays can be calculated from this representation, although in this case the processing time is still large.

1.3.3 Segmentation

In order to extract the surface of an anatomical feature, the data must first be classified such that the location of the feature is known. The automation of this task is probably the most challenging problem in 3-D ultrasound (and many other related research areas). This is equally the case for voxel-based and scan-plane based data. The problem is accentuated with ultrasound because of the poor quality typical in the images when compared with those of other imaging modalities. Many researchers recognise this difficulty and use manual segmentation (usually by outlining the feature on a screen with a mouse). For instance, in a recent paper Gopal [29] states that

> . . . manual boundary tracing is the most accurate method for identifying the surfaces of the ventricle for *all* imaging modalities,

and that

> automated methods . . . have not yet reached a stage suitable for adaptation to 3-D echocardiography.

Semi-automated segmentation techniques are suggested by some authors for particular applications (for instance intravascular scans). However, it is not the intention of this report to approach this problem — manual border tracing is used throughout this work.

1.3.4 Surface representation from segmented images

Surface reconstruction achieves two goals — the ability to display the object in 3-D distinct from the rest of the data, and the estimation of object volume. The former can only be achieved by surface reconstruction, whereas the latter can also be achieved without it, by planar measurements made in each scan plane (discussed later). Most surface reconstruction techniques have been developed for parallel contours, which is the case if the data is voxel-based and the segmentation was performed after interpolation to this representation. Even then, estimating the surface position in a robust and feasible manner is difficult, as can be seen from the many techniques (particularly within the literature on triangulation) which have been suggested for doing this.

In order to reconstruct the surface directly from contours segmented in the original scan planes of a freehand scan, and thus avoid a voxel-based intermediate step, we require a method which does not assume parallel planes. Such reconstruction methods are few and far between: an overview of the reported schemes (as applied to ultrasound data in particular) follows.

Surfaces from arbitrarily orientated cross-sections Most of the following techniques will not in fact handle arbitrarily orientated cross-sections. Usually the scan plane orientation is restricted such that they are nearly parallel (or nearly regular for fan or rotational scanning patterns), or at least do not cross each other. In addition, there are often restrictions on the shape and number of contours in each plane.

In [33], a CAD package was used to triangulate between cross-sections of phantoms of the saphenous vein. They were nearly parallel (a 'translational sweep' was used) and all very nearly circular, such that the resulting surface was conical or cylindrical. A similar application, again with nearly circular cross-sections, was investigated in [42], however the surface construction technique was not explained.

Several researchers [3, 29, 41] make use of an algorithm initially developed by Cook [17]. Two contours are sampled as a set of lines joining boundary points, then the volume calculated from these points by forming tetrahedra with a common central point on each plane. This volume represents that included by the surface formed by triangulating between the contours each boundary point in turn. This will be accurate for fairly similar contours, but not so for larger changes of shape. In addition to this restriction on contour similarity, the contours are not allowed to cross each other.

A totally different approach to surface reconstruction by using deformable models is presented in [18], which also contains a review of the application of similar techniques to the heart. Rather than starting with the cross-sections and attempting to extract a surface, deformable models start with a model and attempt to fit this to the cross-sections. The fitting process is controlled by appropriate forces applied to the model from the data, and usually also internal forces (e.g. bending stress) to prevent the model from moving too far from its original shape. The technique has several advantages — notably that the cross-sections need not be parallel, nor even closed. In fact, the data used to drive the model deformation can consist of arbitrarily scattered points or line segments. Also, the adoption of a model allows some *a priori* knowledge to be included in the process, which is almost always appropriate in medical imaging. However, typical processing times can be large (although the cited method only takes in the order of one minute for a coarse grid), and perhaps more disappointingly, there are frequently many parameters which have to be carefully chosen, or 'tweaked' for each case.

Another method which can cope with overlapping planes is given in [51]. This relies on finding a suitable longitudinal axis, which passes near the centre of each of the cross-sections. Having selected the axis, the original cross-sections are effectively rescanned by a rotational method, where the new scan planes rotate about the longitudinal axis. This creates points on the new scan planes where the contours crossed them, which can then be joined up to produce new contours. Importantly, the new contours no longer overlap (the points are joined up in the order of distance along the axis, *not* in the original scan plane order) so that further processing is greatly simplified. The technique works well for simple shapes, but does not handle complex ones (e.g. with bifurcations). A very similar approach is adopted in [48] for fan scanning of the heart, save that the scan planes do not overlap in this case.

The only technique applied to ultrasound data which will handle arbitrarily orientated cross-sections of any shape is given in [19]. This is similar to the triangulation approach, but here tetrahedra are gradually fitted to the 3-D data set until the entire object volume has been reconstructed. If cross-sections from different planes overlap, they must be properly aligned (registered) before triangulation. Unfortunately, both registration and triangulation can be very time consuming operations.

1.4 Volume measurement using ultrasound

It is apparent from examining the clinical areas where 3-D ultrasound has already been used that volume measurement is a recurrent theme. There is a need for accurate, reliable and preferably fast techniques for measuring volumes using this modality. Volume measurement can be achieved by constructing the surface of the object to be measured, as discussed in the previous section, or it can be calculated directly from measurements made in the original scan planes.

Volumes from scan-plane data Volume measurements using conventional 2-D ultrasound are in frequent clinical use today. These can be achieved by approximating the organ of interest as a simple mathematical shape (generally an ellipsoid), then estimating the major and minor axes by measurement from appropriately selected 2-D ultrasound images. A correction factor is then applied to the result, dependent on the organ, the age and sex of the patient and possibly other factors. There are many formulations for the resulting equations [34, 75].

Ellipsoid formulae are easy to use, but they have an inherent disadvantage in that they make geometrical assumptions about the shape of a given organ. This leads to errors in the volume measurement which can be greater than 20%. An alternative approach which is possible with 3-D ultrasound is planimetry, where the volume is calculated from the cross-sectional areas and plane positions. The most common implementation of this is step-section planimetry, which assumes that the cross-sections are on parallel planes.

There are numerous reports which indicate that step-section planimetry is much more accurate than ellipsoid or other geometrical formulae, for example [1, 53, 66]. Terris [75] compares step-section planimetry with sixteen geometrical equations for measuring prostatic volume and finds that using $\frac{\pi}{6}(transverse\ diameter)^2(anteroposterior\ diameter)$ is marginally more accurate than planimetry, but does not apply this result to an independent data set.

Watanabe developed a planimetric method for calculating volumes from cross-sections which were not parallel [77] and indeed could be overlapping. This has rarely been used in practice, but has been implemented for the prostate [7].

Volumes from surfaces Once the surface of an object has been estimated, the volume can be calculated from this surface representation. Several possible ways of doing this are:

Volumetry If the object of interest has been sampled to a regular array of points, the volume can be calculated simply by summing the number of points inside the object and multiplying this by the volume contained by one point. This technique is often employed even when freehand scanning has been used, so long as the scans are first interpolated to a uniform grid.

Tetrahedral Volume If the surface has been estimated by forming tetrahedra, the volume can be calculated from the sum of the volumes of these tetrahedra, the volume of a tetrahedron being $\frac{1}{6}\vec{pa} \times \vec{pb} \times \vec{pc}$, where the points p, a, b and c are the vertices of the tetrahedron. Alternatively, the polyhedral approximation formula developed by Cook [17] can be used. This is based on the above equation, but formulated in terms of the points making up the object cross-section on each plane. Although this appears to allow volume calculation from cross-sections without triangulation, in fact a simple triangulation is assumed in the algorithm which will only be correct for simple shapes. This technique is used for instance in [3, 29, 41].

Cylindrical/Pyramidal Volume Tetrahedra are not the only simple mathematical shapes which can be used to estimate the volume. If the scanning pattern was rotational, parts of cross-sections can be connected with the mid-point of the rotation to form pyramidal or cylindrical part sections, from which the volume can be calculated. This technique has been used for the eye [37]. Moritz also applied this technique to freehand scans by re-sampling these scans in a rotational scanning pattern and then calculating the volume from the new cross-sections [51].

Volume from Surface Hughes has suggested two ways of measuring the volume directly from a triangulated surface, without forming tetrahedra. 'Ray Tracing' involves projecting rays from a 2-D grid through the object, and calculating the intersection points of these rays with the object. The volume can then be deduced from the length of each ray inside the object and the granularity of the 2-D grid [36]. Alternatively, a discrete version of Gauss' theorem can be adapted to calculate the volume component for each individual triangle such that the sum of these components is equivalent to the object volume [35].

Comparisons of some of the various volume measurement techniques have been performed [35]. It is clear from these that any of the non-geometrical methods are to be preferred over the ellipsoid or similar equations [29].

1.5 Overview of proposed methods

Two methods for calculating the volume of an organ are presented in this report, one with and one without reconstruction of the surface. They have several factors in common :

- Both methods calculate the volume from segmented *cross-sections*. This eliminates the need for a prior geometrical model of the organ of interest. The segmentation in both cases is performed manually using a mouse.

- The cross-sections are segmented in the original scan planes, which must have known position and orientation. There is *no restriction on the orientation* of these scan planes, they can be non-parallel or overlapping, have widely varying spacing, or even have a changing sweep direction (i.e. go backwards then forwards). The only restrictions on the *scanning pattern* are that the scan planes must enclose the entire object, and the side of the plane which first enters the object must also be the first side to leave it. This means that the methods are applicable to 3-D freehand ultrasound, without the need for interpolation to a voxel-based representation.

- Both methods are *sequential* in that they calculate the volume from the scan planes in time order of scanning, using a pair of planes at a time.

- Both methods are equally appropriate for *simple or complex shapes*. They will handle bifurcations, invagination, large changes of scale, shapes with multiple contours and internal holes, etc. The processing for complex shapes is identical to that for simple shapes — there is no exception handling, which results in straightforward code. In simple cases (e.g. simple ellipsoids or cylinders), the methods reduce to their simpler counterparts, and in some mathematical cases (e.g. cylinders) reduce to the actual equations for the volume.

- Volume calculation or surface construction methods which properly deal with overlapping cross-sections must be able to cope with the contradictory information that these situations can present. Both the methods described handle this type of registration error, in that the algorithms will complete and the volume calculation will degrade gracefully with the same magnitude of error as the input registration errors. *Pre-processing of the cross-sections to remove registration errors is not required.*

- Both methods are fast, completing in less than one second (planimetry based) or about 20 seconds (surface reconstruction) for typical ultrasound applications. These times do not include the manual segmentation, but do (in the case of surface reconstruction) include the preparation of the surface for 3-D display.

Cubic planimetry Cubic planimetry is an extension of Watanabe's original volume estimation by vector areas and centroids of serial cross-sections. The original derivation is examined and it is found that the use of trapezoidal interpolation was the major cause of inaccuracy. This causes volume overestimation for conical sections, and underestimation for spheres and ellipsoids (noted in practice [47]). An extension to cubic interpolation was hinted at in the original paper [77] and later suggested by Basset [7]:

> The accuracy of the volume measurement can also be improved by modelling the object with cubic polynomial functions (spline). However, it would be difficult to justify the use of a more accurate but therefore more complex calculation method because the measurement accuracy would still depend primarily upon the precision of the operator's hand-drawn contours with respect to the actual contours.

However, the technique *is* justified since it increases the accuracy whilst only adding a few tens of milliseconds to the computation time, compared with the several minutes required in any case to perform the segmentation. More importantly, it also reduces the number of cross-sections required for a given accuracy.

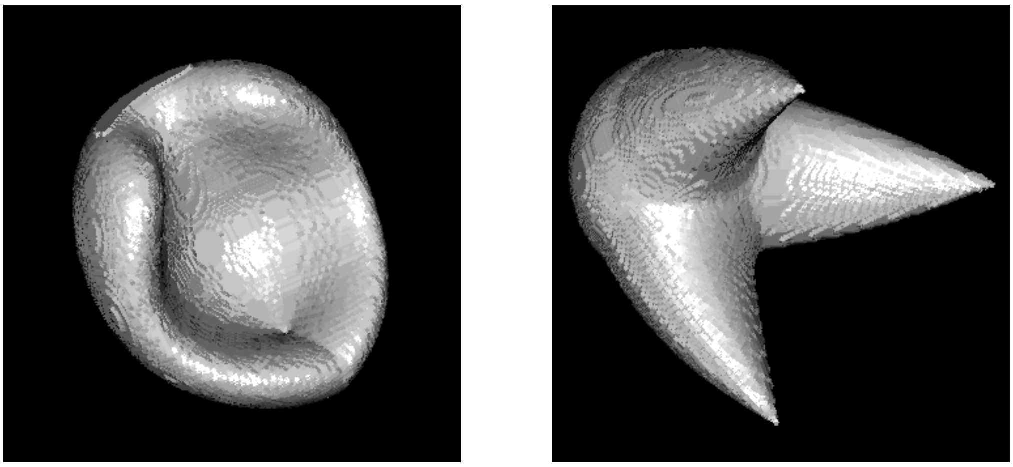
Maximal disc guided, shape based interpolation Shape based interpolation was first suggested by Raya and Udupa [68] and later adapted by Herman [31]. The 3-D form was introduced to interpolate parallel MRI slices, in order to improve the inter-plane resolution of the data set. Shape based interpolation is extended here in three ways:

- Shape based interpolation is applied to arbitrarily orientated rather than parallel planes. It is used to find the surface of the desired object directly, rather than to interpolate cross-sections in new scan planes.

- The shape based interpolation technique is extended (for both parallel and non-parallel cases) to improve the handling of complex shapes, by allowing the direction of interpolation to vary within the scan planes. This variation in direction is itself guided by the shape of the object cross-sections, as defined by a set (smaller than the minimal set [13]) of maximal discs.

- The distance field values used during the interpolation are also used to improve the 3-D display of the surface by providing more accurate surface normals than would otherwise be possible.

The resulting algorithm is simple and fast, and produces a smooth, realistic surface for both simple and complex cases.



(a) ‘Baseball glove’ object

(b) ‘Jester’s hat’ object

Figure 3: **Mathematically generated test objects.** The rendered surfaces were reconstructed from approximately 80 parallel scans. The surface texture is a result of the granularity of the scans.

2 Systems and tools

2.1 Freehand 3-D ultrasound system

In-vivo ultrasound data was recorded using a Toshiba Powervision 7000[1] with a 3.75MHz convex curvilinear array probe. A Polhemus FASTRAK[2] magnetic field position sensor was mounted on this probe. The position signal from this, in addition to the video output of the ultrasound machine, were then fed to a Silicon Graphics Indy workstation[3] with a digital acquisition card. Acquisition of the ultrasound images and position readings, calibration of the system, and segmentation of the data were all performed using Stradx v5.0[4] software [62, 63, 64]. The calibration process gave accuracies of typically ±1mm in all directions. The manual segmentation process is described in Appendix A.

Version 5.1 of the Stradx software includes the cubic planimetry and maximal disc shape based interpolation presented in this report.

Processing of the Stradx v5.0 data was done both on a Silicon Graphics Indigo 2 workstation with 256Mb RAM, and a 166MHz Pentium MMX with 32Mb RAM running Linux, with similar performance on each platform.

2.2 Generation of test objects

In order to validate the volume calculations, mathematical objects were generated with known volume, which could then be scanned in arbitrary (sequential) sweeps. Each point on each

[1] Toshiba America Medical Systems, Tustin, California

[2] Polhemus Incorporated, Colchester, Vermont

[3] Silicon Graphics Incorporated, Mountain View, California

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

scan was tested against the equation for the object to determine whether it was inside or outside, thus generating pre-segmented object cross-sections in the same format as those generated after manual contour tracing of real ultrasound scans.

More complex objects were generated from binary combinations of simple objects (sphere, ellipsoid, cone, cylinder, cube). Coordinate system transformations were also used to 'stretch' or 'twist' these objects so that the generated models better approximated real applications.

The scanning pattern could be controlled in position, azimuth, elevation and roll. A random jitter could also be added to the positions and orientations, in order to represent a real sweep with a slightly unsteady hand. In addition, random errors could be introduced to the scan plane locations, in order to represent possible registration errors in the real system.

The non-trivial objects used are shown in Figure 3. The 'baseball glove' was generated from an ellipsoid twisted around all three axes, and stretched along one. The 'jester's hat' was generated from a sphere and three cones, again twisted around all three axes.

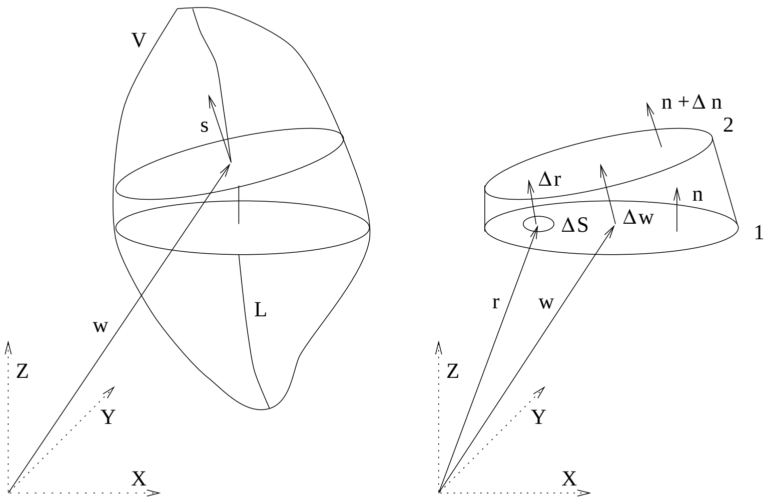


Figure 4: **An object with volume V, and cross-sections S.**

3 Cubic planimetry

3.1 Volume from arbitrarily orientated planes

The equation for the volume v of any object defined from serial cross-sections is given by Watanabe [77]:

$$v = \left| \int_L \vec{s} \cdot d\vec{\omega} \right| \tag{1}$$

where $\vec{\omega}$ is the position vector of the centroid of the cross-sectional surface S whose vector area is given by $\vec{s}$, and L is the path of $\vec{\omega}$ as the object is scanned. Equation (1) can be implemented discretely by approximating the integral using the trapezoidal rule between each pair of slices, which gives :

$$v = \left| \sum_{i=2}^{N} \frac{1}{2} \left(\vec{s_i} + \vec{s}_{i-1} \right) \cdot \left(\vec{\omega_i} - \vec{\omega}_{i-1} \right) \right| \tag{2}$$

where the N cross-sections have vector areas $\vec{s_1}, \ldots, \vec{s_N}$ and centroids $\vec{\omega_1}, \ldots, \vec{\omega_N}$. This approximation is equivalent to assuming that the surface area projected onto a plane normal to the path of the centroids, L, varies linearly from one slice to the next. This is clearly true for objects whose cross-sectional area does not vary, e.g. prisms, and in this case equation (2) is the exact solution. Paraboloids also have this property. However, objects which are either more concave or more convex than a paraboloid will not be correctly approximated by this equation. For example, the volume of a cone will be overestimated, and that of a sphere or an ellipsoid will be underestimated. This error increases as the number of scan planes reduces.

Equation (2) can be easily implemented on a computer once the cross-sections have been determined and then the areas and centroids calculated. In practice, the first step is by far the most time consuming, typically taking half a minute or so for each cross-section assuming it is outlined manually. Once this has been done, the calculation of the volume is trivially fast in comparison (much less than one second).

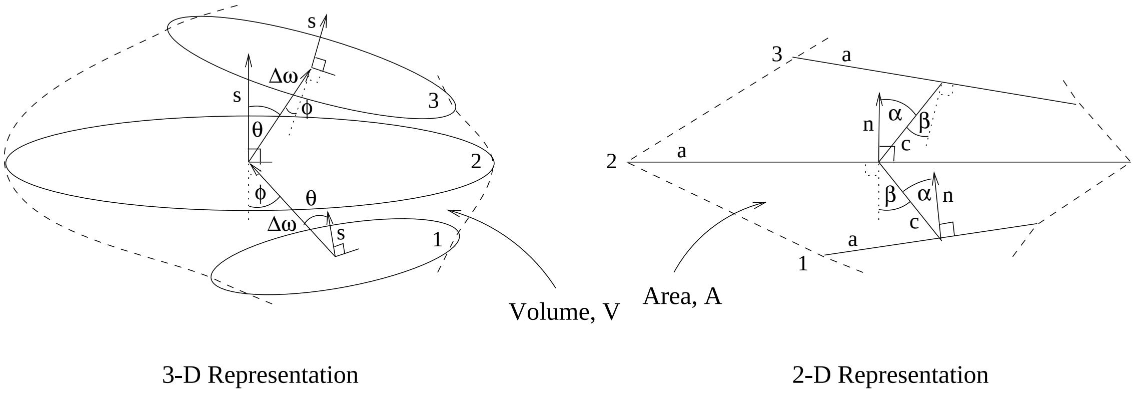


Figure 5: **3-D and 2-D representation equivalence.** i) The length of a 2-D line, a, is equivalent to the area of the cross-section, $|\vec{s}|$. ii) The length of the line, c, is equal to the magnitude of the vector, $\left|\vec{\Delta\omega}\right|$. iii) The angle, α, between the line c joining the centres of each line a and the normal to those lines is equal to the angle, θ, between the vector area $\vec{s}$ and the vector $\vec{\Delta\omega}$ joining the centroids of each area. iv) Similarly, the angle, β, is equal to the angle, ϕ.

Clearly, some form of cubic rather than trapezoidal interpolation would increase the accuracy of the volume estimate and eliminate the bias towards paraboloids or prisms. It has been argued [7] that the small increase in accuracy this would represent does not justify the additional complexity that would be required. However, two points can be made in defence of this approach. Firstly, the additional complexity is completely transparent to the user — once the algorithm has been implemented, the user performs precisely the same operations (i.e. outlining of the cross-sections) in both cases. Secondly, the reduction in the number of cross-sections required for an accurate volume estimation with cubic planimetry is very welcome, since this is the time consuming step in the process. We present results to demonstrate this advantage in Section 5.

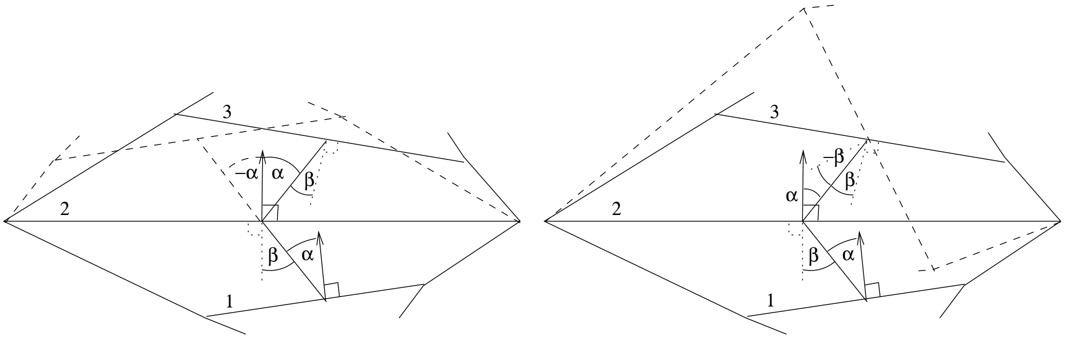
In order to estimate equation (1) using cubic interpolation, two main decisions must be made. The first is the selection of the points or vectors to interpolate. The second is the selection of the cubic representation for the interpolation.

3.2 2-D representation of the problem

Interestingly, the whole problem can be reduced to finding the area of a carefully constructed 2-D graph which represents a combination of the original 3-D object with the scanning pattern. The equivalence between the 3-D and 2-D representations is shown in Figure 5.

The area enclosed within the dashed and solid lines in the 2-D representation is equivalent to the volume which would be calculated by Watanabe's trapezoidal equation from the 3-D representation. This can be easily proved by considering the 2-D representation to have a nominal thickness of 1 unit, and then applying equation (2) to calculate the area:

$$A = \left| \sum_{i=2}^{N} \frac{1}{2} \left(a_i \hat{n}_i + a_{i-1} \hat{n}_{i-1} \right) \cdot \left(\vec{c}_{i-1} \right) \right|$$



α- variation

β - variation

Figure 6: **Choice of angles for 2-D representation.**

$$
\begin{aligned}
&= \left| \sum_{i=2}^{N} \frac{|\vec{c_{i-1}}|}{2} \left(a_i \hat{n}_i \cdot \hat{c}_{i-1} + a_{i-1} \hat{n}_{i-1} \cdot \hat{c}_{i-1} \right) \right| \\
&= \left| \sum_{i=2}^{N} \frac{|\vec{c_{i-1}}|}{2} \left(a_i \cos\beta + a_{i-1} \cos\alpha \right) \right| \qquad (3)
\end{aligned}
$$

Equation (2) can be similarly re-written as:

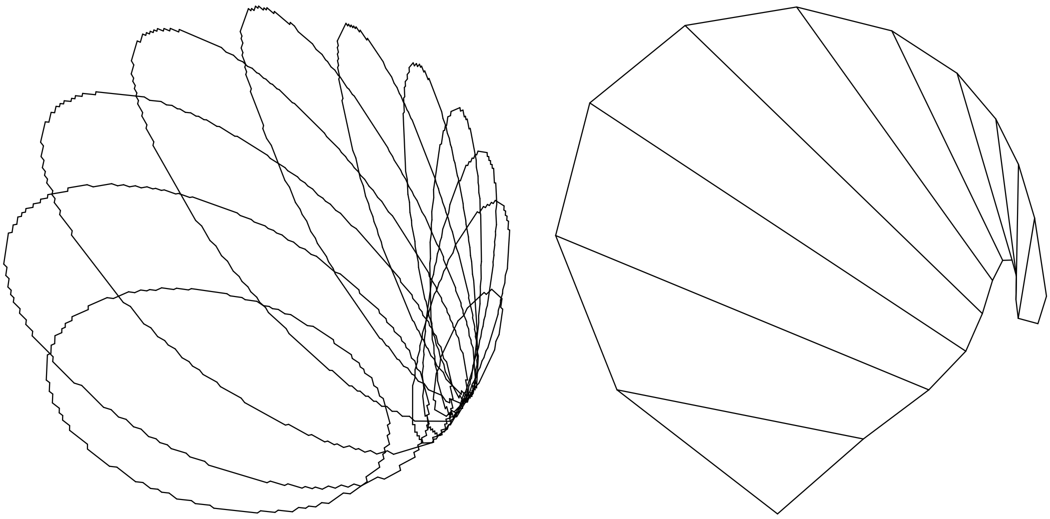
$$
v = \left| \sum_{i=2}^{N} \frac{\left|\Delta\vec{\omega_{i-1}}\right|}{2} \left(|\vec{s_i}| \cos\phi + |\vec{s}_{i-1}| \cos\theta \right) \right| \qquad (4)
$$

If the variables in equations (3) and (4) are equated for all values of i, then $A \equiv v$. There is, however, significant redundancy in this conversion. Firstly, only the multiple of the lengths of the lines a and c is used, and hence an arbitrary scale factor can be multiplied into one, so long as it is divided from the other. This has the effect of stretching or shrinking the 2-D graph, but has no bearing on the volume calculation. Secondly, only the cosine of the angles α and β are used, hence the angles can be arbitrarily positive or negative. The effect of this choice is shown in Figure 6.

Although this choice has no effect on the volume calculated by the trapezoidal method, it clearly does affect how well the 2-D representation matches the original 3-D representation. Cubic interpolation involves the use of information from several sequential slices and, therefore, an additional heuristic rule is required to ensure that the angles α and β are chosen correctly.

For each choice of angle α between the line joining centroids c_i and the area representation a_i, the angle which c_i makes with c_{i-1} is also calculated. The value of α is then chosen for which this calculated angle is closest to the 3-D version (i.e. the angle which $\Delta\omega_i$ makes with $\Delta\omega_{i-1}$). A similar rule is employed for the angle β, using the area normals rather than the lines joining the centroids as the reference.

The result of this entire process is shown for a sphere in Figure 7. The sphere was sliced with a scanning pattern which varied in position, azimuth, elevation and roll. The resulting



(a) original cross-sections

(b) 2-D graph representation

Figure 7: **Scanned sphere in 3-D and 2-D representations.**

2-D graph retains some of the shape of the sphere but also reflects the way in which the sphere was scanned.

3.3 Cubic interpolation of 2-D representation

If instead of joining the end points of the lines a with straight lines, a smooth curve is fitted between them, then the area enclosed by these curves should represent a more accurate measure of the volume of the original object. The curves must at least be cubic, since we would like to have continuity in at least the first derivative (i.e. the curves are smooth at the joints). They must also be defined parametrically, since we expect them to have multiple values in both x and y directions.

The smoothest possible curve could be calculated by taking the end-points from all the slices simultaneously and using an appropriate radial basis function to calculate the curve parameters. However, radial basis functions are in general very costly to compute, which would violate one of the motivations for improving the volume calculation, namely that the increase in processing time is negligible. A less optimal but much faster solution can be found by using parametric cubic splines.

Parametric cubic splines In depth descriptions of the various forms of splines can be found, for example, in [26]. Briefly, they are curves which in 2-D are represented parametrically by sets of coefficients:

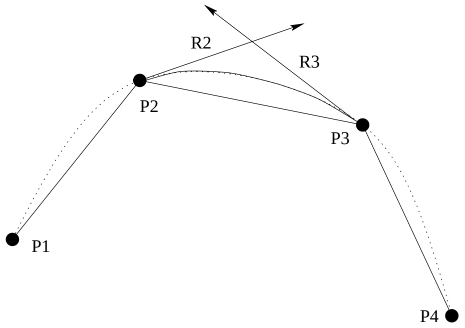


Figure 8: **Selection of points for cubic splines.**

$$\begin{aligned}
\begin{bmatrix} x(t) & y(t) \end{bmatrix} &= \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} x_3 & y_3 \\ x_2 & y_2 \\ x_1 & y_1 \\ x_0 & y_0 \end{bmatrix} \\
&\equiv \mathbf{TC} \\
&\equiv \mathbf{TMP} \qquad (5)
\end{aligned}$$

where **C** is a matrix of coefficients, and **T** is the parameter matrix. The matrix **C** can be generated from a list of control points **P** and the spline transformation matrix **M**. The choice of the control point and transformation matrices determines how the cubic spline will be defined. In particular, we require a spline which *interpolates* the control points, i.e. the resulting curve passes through the points which are used to define it.

One of the simplest cubic spline forms which has this property is the Hermite, in which case:

$$\mathbf{P} = \begin{bmatrix} \vec{p_2} \\ \vec{p_3} \\ \vec{r_2} \\ \vec{r_3} \end{bmatrix}, \quad \text{and} \quad \mathbf{M} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad (6)$$

where the points $\vec{p_2}$ and $\vec{p_3}$ and the gradient vectors $\vec{r_2}$ and $\vec{r_3}$ are defined in Figure 8. The choice of gradient vectors is not obvious in this case, since we only have a set of connected points. An alternative cubic spline form invented by Catmull and Rom [14] overcomes this by using the four points $\vec{p_1}$ to $\vec{p_4}$ directly. In this case, the point and transformation matrices are:

$$\mathbf{P} = \begin{bmatrix} \vec{p_1} \\ \vec{p_2} \\ \vec{p_3} \\ \vec{p_4} \end{bmatrix}, \quad \text{and} \quad \mathbf{M} = \frac{\mathbf{1}}{\mathbf{2}} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 1 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad (7)$$

With this form of cubic spline, the control points $\vec{p_1}$ to $\vec{p_4}$ are the points to be interpolated (in our case the end points of the lines a), and the curve is fitted between $\vec{p_2}$ and $\vec{p_3}$, as in

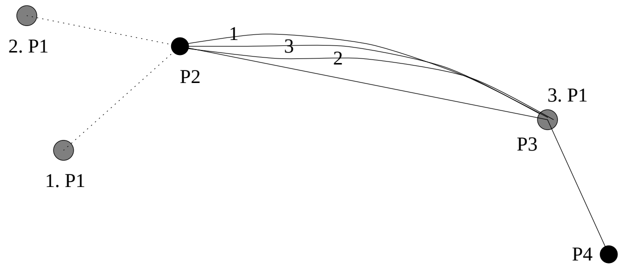


Figure 9: **Handling of end points for cubic splines.**

Figures 8 and 9. This is equivalent to using the Hermite form, with the gradient vectors defined as:

$$\vec{r_2} = \frac{1}{2}\left(\vec{p_3} - \vec{p_1}\right), \quad \text{and} \quad \vec{r_3} = \frac{1}{2}\left(\vec{p_2} - \vec{p_4}\right), \tag{8}$$

Handling of end points The first and last curve segments are necessarily a special case, since only three points can be used to fit the curve. There are a variety of ways of handling this, which can all be implemented by inventing the additional missing point, as in Figure 9.

1. If the additional point is chosen such that the point set $\vec{p_1}$ to $\vec{p_4}$ is symmetrical about the mid-point of $\overrightarrow{p_2p_3}$, then the final curve will also be symmetrical, i.e. the gradients at each end will be equal and opposite.

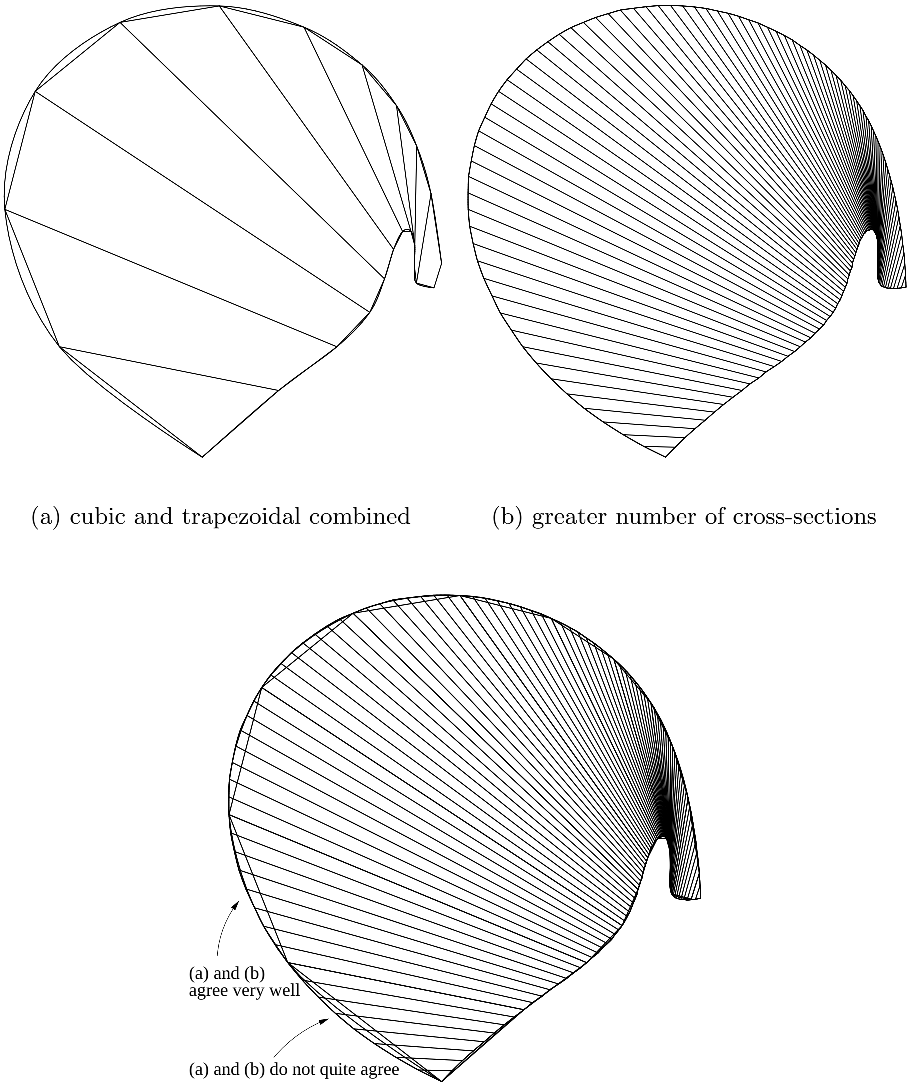
2. If the additional point lies along the extension of the line $\overrightarrow{p_3p_2}$, then the gradient at the last point will be equal to that of the line $\overrightarrow{p_3p_2}$.

3. If the additional point is identical to $\vec{p_3}$, this has the effect of placing a null gradient at the end point. The resulting curve has a rate of change of gradient of zero at the end point.

The last solution is the simplest and most natural, and is therefore employed in this technique. The resulting curve, for the same situation as in Figure 7, is shown in Figure 10, together with the actual object curve which results from scanning in smaller steps.

Calculation of the enclosed area Once the curves joining the end points of the lines a have been defined, the area enclosed by them can be calculated directly from the parametric coefficients of each curve. This calculation is based on the application of equation (1), and is given in Appendix B.

Interpolation of alternative points Rather than fitting curves to the end points of the lines a, they can be fitted directly to the vector normals to these lines $\hat{n}$ and the centroid difference vectors $\vec{c}$. The result of this, overlaid on the other interpolation technique, is shown in Figure 11(a). In fact, the resulting curves can be deduced from each other, and the volume calculation is identical for each method.

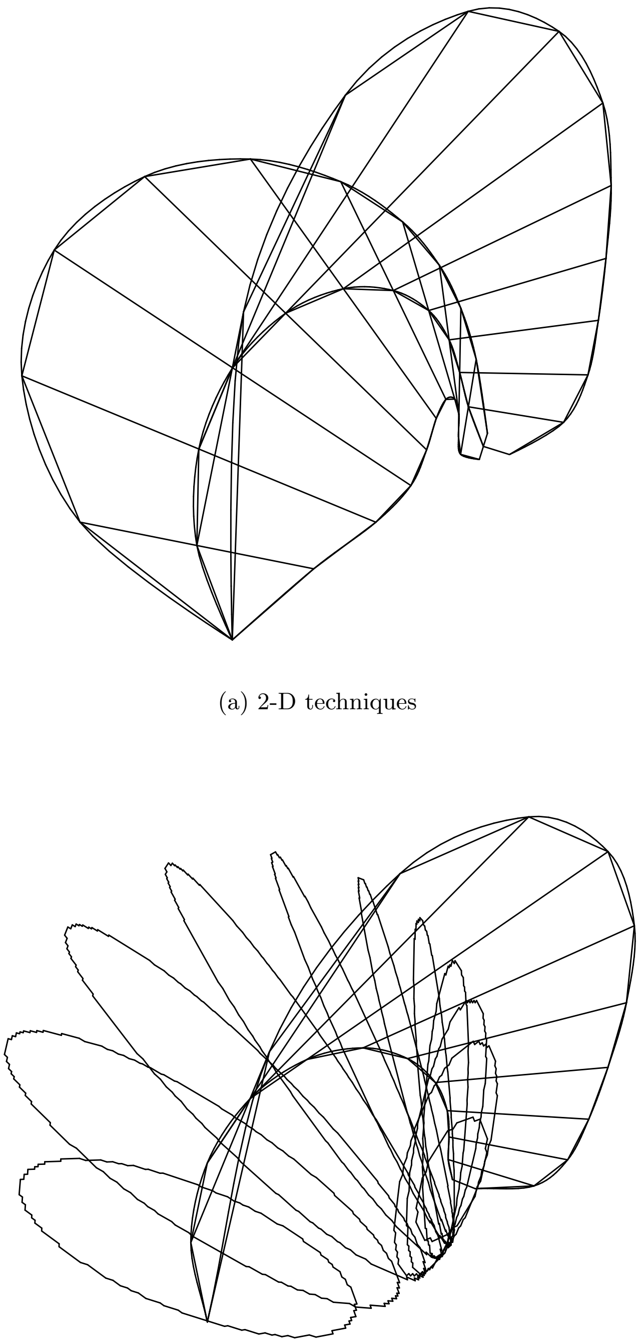


(a) cubic and trapezoidal combined

(b) greater number of cross-sections

(c) previous figures superimposed

Figure 10: **Cubic and trapezoidal interpolations compared with actual 2-D graph for a scanned sphere.**



(a) 2-D techniques

(b) 3-D technique and original cross-sections

Figure 11: **Alternative cubic interpolation techniques applied to a sphere.**

3.4 Cubic interpolations directly from 3-D representation

In fact, the cubic splines can be fitted directly to the original 3-D vector areas $\vec{s}$ and centroid difference vectors $\vec{\Delta\omega}$. In this case the 2-D representation is not required, and the resulting curves produce a representation which is flat, but not planar. This is hard to visualise in 2-D, but is shown in Figure 11(b) superimposed on the original 3-D cross-sections. The vector areas $\vec{s}$ emanate from the centres of the original cross-sections, and the two curves join the end points of these vectors.

In this case, the curve parameters are now defined in 3-D, hence equation (5) and the calculation of enclosed area in Appendix B both require an additional $z(t)$ coordinate. In addition, using method 3 to handle the end points results in poorly conditioned curves, hence method 2 is adopted for this case.

The resulting volume is slightly different from that obtained using the 2-D graph method, although the implementation is simpler since the translation from 3-D to 2-D representation is no longer required. The accuracy of the various techniques is investigated in Section 5.

4 Maximal disc guided, shape based interpolation

4.1 Object based interpolation methods

Interpolation is frequently required in medical imaging in order to fit the input data to a regular grid, with the aim of simplifying later processing. Linear interpolation between parallel scans is the simplest form, however this can lead to unwanted artifacts where the original scans contain objects which have changed shape or position between slices [68]. In order to avoid these artifacts, *object based* interpolation schemes, which take account of the shape of the object in each scan, have been investigated.

Dynamic elastic interpolation This method, introduced by Chen [16], starts with the definition of the object cross-sections in parallel scan planes. For each pair of scans, a "force field" is applied to the first cross-section in order to distort it to the shape of the second cross-section. Once calculated, this field can then be used iteratively to calculate intermediate cross-sections. These are then mapped to a grid, from which the eventual surface is reconstructed. Alternatively, a coarser surface can be generated simply by stacking up the interpolated cross-sections. This method has been extended to handle bifurcations, but the force field calculation is quite complex.

Shape based interpolation First proposed by Raya and Udupa [68], this method also assumes that the object has already been segmented. The first step is to calculate for each pixel in the scan, the distance from the closest point on the object cross-section, with the convention that this distance is positive for points inside the object and negative otherwise. Secondly, these transformed images are interpolated using a linear or cubic function, to generate new transformed images. Finally, the new images can be thresholded at zero such that any pixels greater than zero are considered to be inside the objects on the new scan planes. A faster method was also proposed which restricted the distance calculation to pixels which were contained by the object cross-section on only one of the parallel scan planes. They suggested that the method could be adapted for near-parallel images.

Herman [31] later introduced a more accurate and efficient algorithm for calculating the distance values, based on a distance transformation introduced by Borgefors [11]. This transformation used a chamfer code, which is a better estimate of Euclidean distance than Raya and Udupa's city-block distance. At about the same time, Montanvert [50] also arrived at a near identical algorithm for use in 2-D granulometry and 3-D reconstruction, once again by interpolating Borgefor's chamfer code. In this case, the distance maps were interpreted as "fate maps" — for each pixel, the distance value indicated a date, where zero indicated the "birth" of the pixel in the interpolated images. The results were, however, identical to the linear interpolation used by Herman. Another near identical algorithm was presented later in [39], although here the distances were calculated by a more complex method. Ohashi [59] used morphological erosion and dilation operators to produce similar results, this time applied to pore geometry.

All of the above shape based methods can handle bifurcations and relatively complex shapes with minimal fuss. However, they can also produce undesirable artifacts when the shape on each contour changes radically, or undergoes significant translation. The effect of translation can be easily demonstrated by examining the results for a simple cylinder, scanned at an oblique angle (e.g. an artery) — see Figure 12.

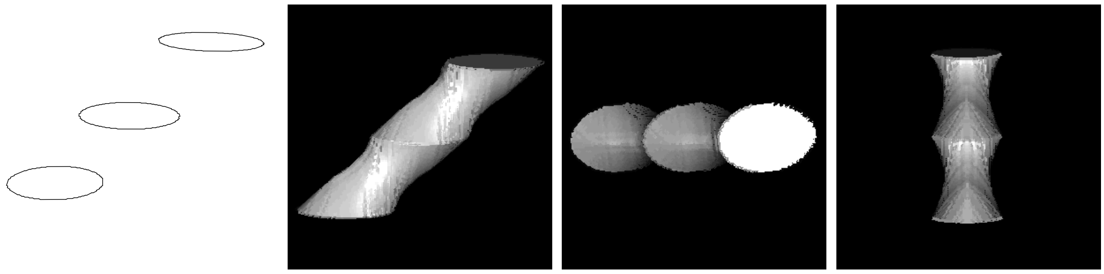
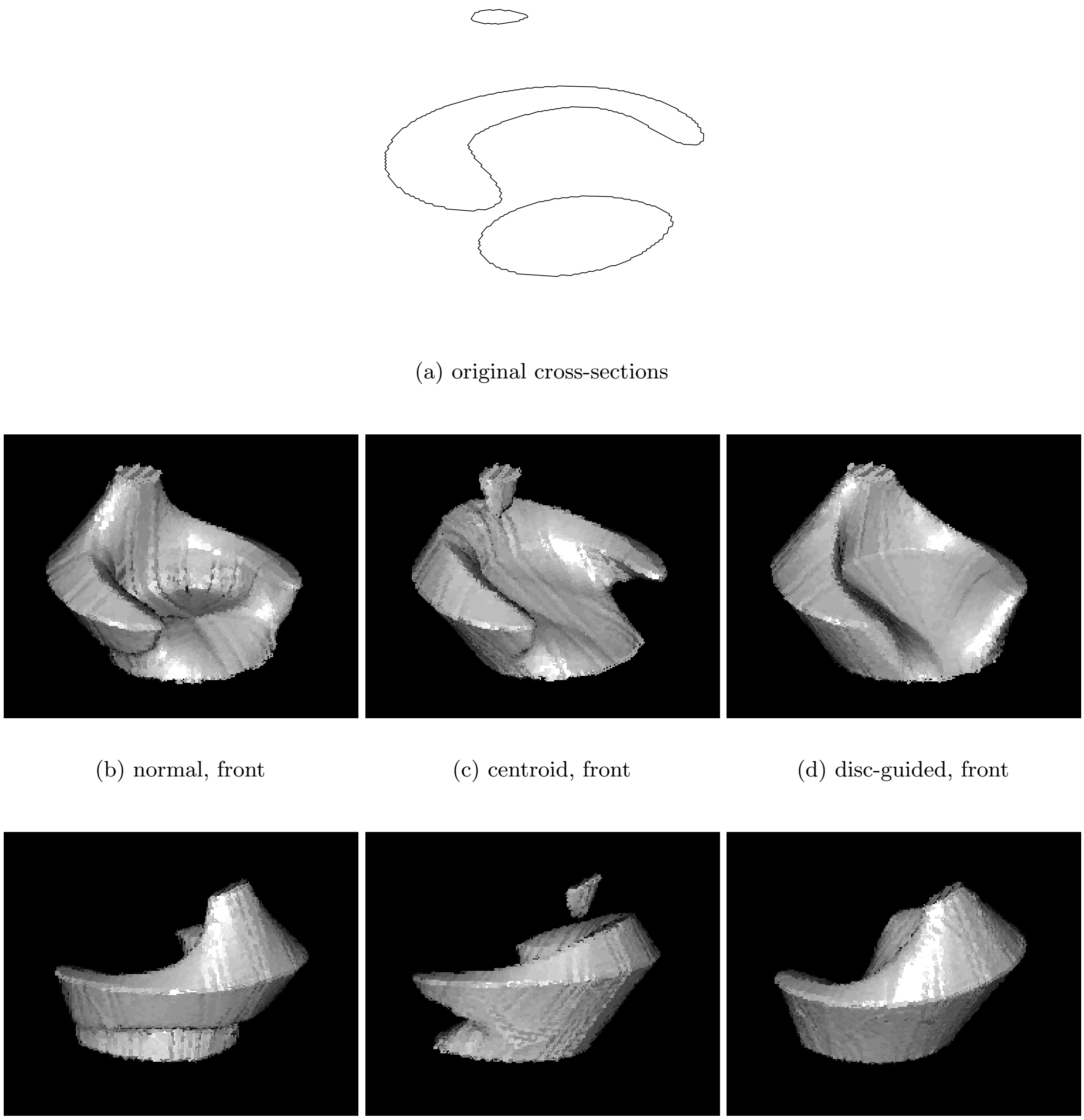

(a) original contours (b) side (c) top (d) front

Figure 12: **Limitations of conventional shape based interpolation:** simple cylinder.

The side view is nearly correct, however the front view shows that the surface of the cylinder is in fact concave. This is because the interpolated object can only exist within the bounds of the original cross-sections, projected in the direction of interpolation, as seen from the top view. Higgins addressed this problem in order to use shape based interpolation on tree-like objects [32]. Several improvements to the original scheme were suggested. Rather than interpolating parallel to the scan planes, the interpolation was performed in a direction connecting the centroids of cross-sections of the same object. Individual objects were treated separately, and the interpolated objects binary-ored to create the final object contours. Higgins also suggested using the grey-scale information in the original scan to enable interpolation along a more natural curve joining the object centroids, rather than a straight line.

These modifications successfully correct the translation problem, but have no effect on the problem of radical shape changes — in fact they can make these effects worse. Figure 13 is an example of such a shape. Here the baseball glove object presented in Figure 3 has been scanned in three (nearly) parallel planes, and reconstructed by shape based methods. The equivalent of Herman's method, (b) and (e), using planar normals, produces a connected object but with severe gradient discontinuities, which clearly have no basis in the original cross-sections. If the interpolation is performed along the line connecting object centroids, (c) and (f), the surface gradients are improved at the front and back in the middle sections — but at the cost of heavy invagination at the sides and a disconnected top. Interpolation using maximal discs, described in the following sections, is shown in (d) and (g).

Apparently, the only attempt to correct this problem has been recently made by Liu [45]. Liu's interpolation method, *edge-shrinking interpolation*, is a close relative of shape based interpolation, in that the intermediate cross-sections are still derived from distance transforms calculated using a chamfer code. In this case, the start contour is eroded by a morphological disc of varying size, determined by the distance code at that point in the contour. The results of this operation are, however, very similar to those achieved with a more straightforward linear interpolation. Liu uses an interpolation direction similar to connecting centroids, but based on minimisation of the distance field of one contour, sampled along the other contour. In addition, one contour can be "shrunk" before interpolation in order to fit inside the other contour, thus reducing the effects of radical shape changes (a method which would give



(a) original cross-sections

(b) normal, front

(c) centroid, front

(d) disc-guided, front

(e) normal, back

(f) centroid, back

(g) disc-guided, back

Figure 13: **Limitations of conventional shape based interpolation:** baseball glove object.

reasonable results on Figure 13 only if the middle contour was reduced to fit the bottom contour, but not the other way around). This technique improves the interpolation result, but is much more complex than the other techniques. In addition, the treatment of multiple contours is the same as for Higgins in that the results are binary-ored, generating gradient discontinuities at the junctions.

Shape based interpolation has also recently been extended to grey-level rather than binary images [30].

Other similar techniques Several other object based interpolation strategies have been suggested. *Cores* can be extracted from the grey-level image (without segmentation) and used to interpolate intermediate parallel slices [65]. Morphing (more famously known for its use in films such as Terminator II) has also recently been used on object cross-sections to guide the interpolation [73]. The idea of interpolating in the direction of object centroids has also been used to reconstruct an object from "staircases" formed by iteratively connecting points on two cross-sections [78].

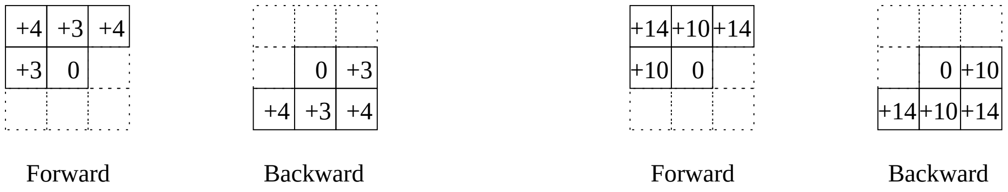
Advantages and disadvantages Shape based interpolation is a method which is inherently simple (in its simplest form only involving two operations, distance transformation and interpolation, each acting uniformly on all the data) and yet can handle shapes which force triangulation algorithms to reach for their exception handling routines. Simplicity is not just important for increasing processing speed — it can also act as a better guarantee against unusual behaviour in unexpected situations. However, there are clearly still drawbacks in the surfaces which this technique produces. Furthermore, to the authors' knowledge, it has never been extended to the more general situation of non-parallel scan planes.

4.2 Calculation of distance transform

The first step in shape based interpolation is to calculate for each pixel of interest the minimum distance from the object cross-section. This can either be performed as a pre-processing step, yielding a distance transformed image, or as required during the interpolation. Chamfer coding [11] is an efficient way of calculating a distance transform within a bounding rectangle, however additional storage is required for this transformed portion of the image. In addition, it is not necessarily obvious how large the bounding rectangle should be — especially if the planes are not parallel and the direction of interpolation is allowed to vary.

For these reasons, a two phase approach has been adopted in this case. Firstly, chamfer coding is performed in a rectangle which just bounds the object cross-section for a given scan plane. Secondly, an additional algorithm is implemented which can calculate far distance codes using the edge information from the chamfer coded region. When a distance code is required for a given point, the process handling this operation will either return a value from the coded region, or calculate a new value if the pixel is outside this region. Most of the points will be contained within the chamfer coded region, so the interpolation processing time is kept to a minimum. However, the far point algorithm guarantees a correct distance code for any arbitrary point, should that be required.

Near point calculation The various ways of estimating Euclidean distance from an object are reviewed by Borgefors [12]. It is suggested that any of the reviewed methods (city block, chess-board, octagonal, chamfer 3 × 3 or chamfer 5 × 5) may be appropriate in different



Forward

+4	+3	+4
+3	0	

Backward

	0	+3
+4	+3	+4

(a) 3-4 chamfer code

Forward

+14	+10	+14
+10	0	

Backward

	0	+10
+14	+10	+14

(b) 10-14 chamfer code

Figure 14: 3×3 **chamfer code masks.**

situations. In practice, the slight error in distance calculation from the chamfer 3×3 code has very limited effect on the interpolated surface, hence this would seem to be the most suitable method.

Chamfer coding can be efficiently performed on a binary segmented image by passing a “mask” first forward (top to bottom, left to right) then backward (bottom to top, right to left) through the image. Each pixel in the image is summed with the respective non-zero mask pixel, and the minimum of these sums represents the new value for the centre pixel. Examples of these masks are given in Figure 14. Borgefors suggests a good 3×3 mask is the 3-4 code.

Herman suggested a couple of minor changes to this process. Firstly, the slightly different 10-14 mask is used, rather than the 3-4 mask. The image is then initialised such that non-object pixels are given a very negative number, object pixels a very positive number, and pixels which share an edge with the object contour either $+5$ or -5, for object and non-object pixels respectively. This ensures that the distance values are measured as from the contour itself, rather than the outermost object pixels. Secondly, the masking operation is inverted in both passes for pixels which are outside the object (i.e. negative), in that the mask is *subtracted* from the image and the *largest* value used. This enables the simultaneous calculation of the distance fields both inside and outside the object.

An example of a typical cross-section which has been chamfer coded in this way is given in Figure 15. This is from the “jester’s hat” object, sampled on a 200×200 grid, and the distance transformation calculated using both the near and far field algorithms. The left hand contour clearly shows the effect of using this estimate of Euclidean distance — the far field from this contour is an eight sided shape similar to an octagon. However, the estimate is good in the regions near the contours, which are also the regions which have the greatest effect on the eventual object surface.

A fast method for calculating a Euclidean distance field, based on city block and chessboard transformations, has also been suggested [15]. This would increase the smoothness of the surface, however the additional processing time is not considered to be worth the minimal effect it would have on the surface.

Far point calculation The simplest way of calculating the distance field for a single point would be to iterate through the contour points, calculating the distance of each from the selected point, and keeping the minimum. However, it is possible to calculate distance fields for arbitrary points more efficiently by making use of the pre-calculated rectangular region

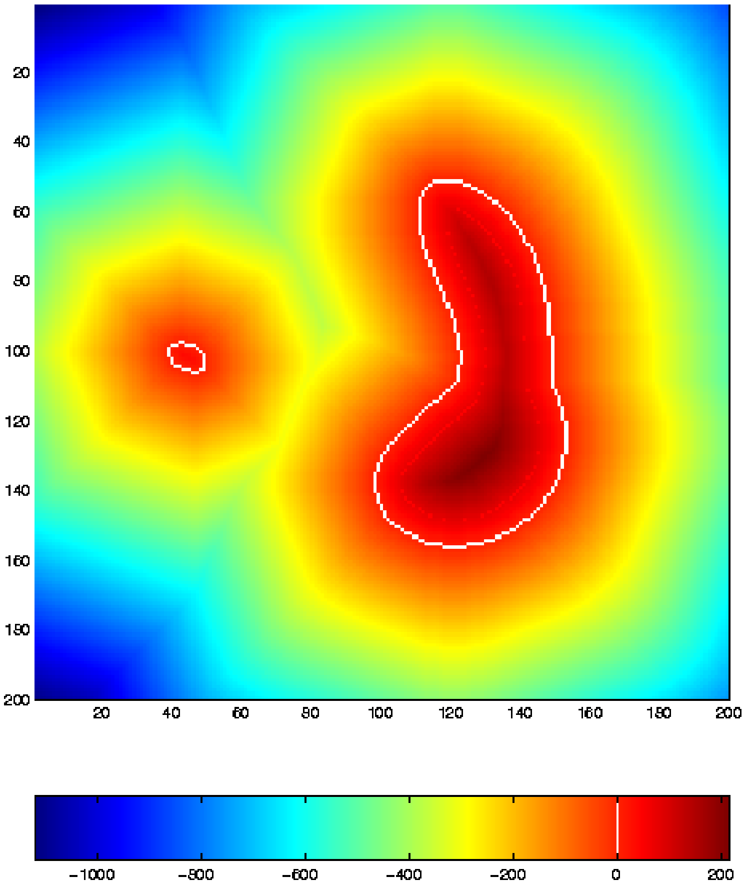


Figure 15: **Example of a 10-14 chamfer coded object contour.**

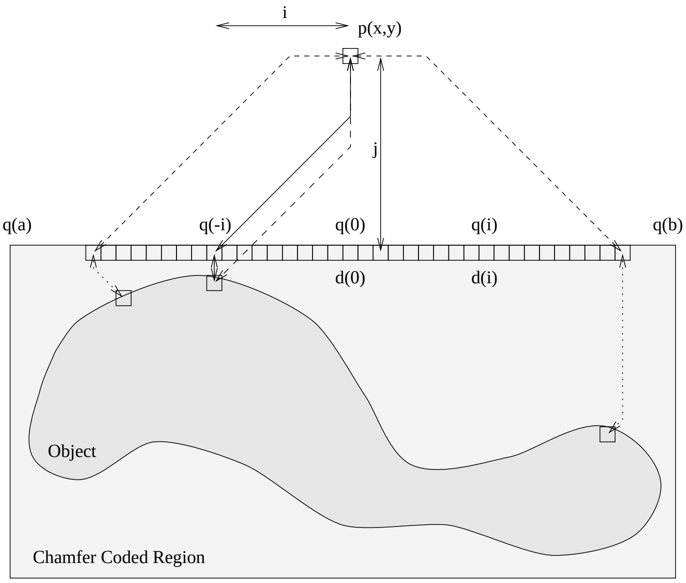


Figure 16: **Far distance code calculation from pre-calculated rectangle.**

surrounding the object cross-section.

Figure 16 is an example of such a cross-section, where the distance code for a far point $p_{x,y}$ is required. Distances in a chamfer 3×3 code can be represented by the Euclidean distance along a path formed entirely from lines parallel or at 45° to the image edges. Hence the distance from $p_{x,y}$ to a point q_i for a 10-14 chamfer is:

$$\begin{aligned} p_{x,y}q_i &= \max\left(10(|j| - |i|) + 14\,|i|\,, 10(|i| - |j|) + 14\,|j|\right) \\ &\equiv \max\left(10\,|i| + 4\,|j|\,, 10\,|j| + 4\,|i|\right) \end{aligned} \tag{9}$$

where i and j are the distances in pixels from $p_{x,y}$ to q_i in the x and y directions respectively. The chamfer distance from the point $p_{x,y}$ to the nearest object point can be calculated by minimising the sum of the distance to the edge of the chamfer region, $p_{x,y}q_i$, and the chamfer code at the edge, d_i. This property of the chamfer 3×3 code is demonstrated by Embrechts [22]. Hence, if $d_{x,y}$ is the chamfer code at point $p_{x,y}$, then:

$$d_{x,y} = \min_{a \le i \le b} \left(p_{x,y}q_i + d_i\right) \tag{10}$$

where a and b are the distances in the x direction from the point $p_{x,y}$ to each corner of the chamfered region. The search space can be further limited by noting that if the distance $p_{x,y}q_i$ is already greater than the minimum $d_{x,y}$ found so far, there is no need to search for greater (or smaller if i is negative) values of i. The resulting process, for calculation of distance field points directly above the chamfer coded region, is described in Figure 17.

The distance code for points immediately below, to the sides, or diagonal from the corners of the region can be calculated in a similar manner. For the corner regions, the search proceeds from the nearest corner point along each of the region edges connected to that point.

```
$i := 0$;    $d_{min} := j + d_0$;    $d_{part} := j$;
while $d_{part} < d_{min}$ and ( $-i \leq a$ or $i \leq b$ ) do
    $d_{part} := \max(10i + 4j,\ 10j + 4i)$;
    if $-i \leq a$ then
        $d_{min} := \min(d_{part} + d_{-i},\ d_{min})$;
    if $i \leq b$ then
        $d_{min} := \min(d_{part} + d_i,\ d_{min})$;
    $i := i + 1$;
```

Figure 17: **Far distance code algorithm.** d_{min} is the current distance code estimate for the point, d_{part} is the chamfer distance from the point to the current edge point. Other variables are as previously indicated.

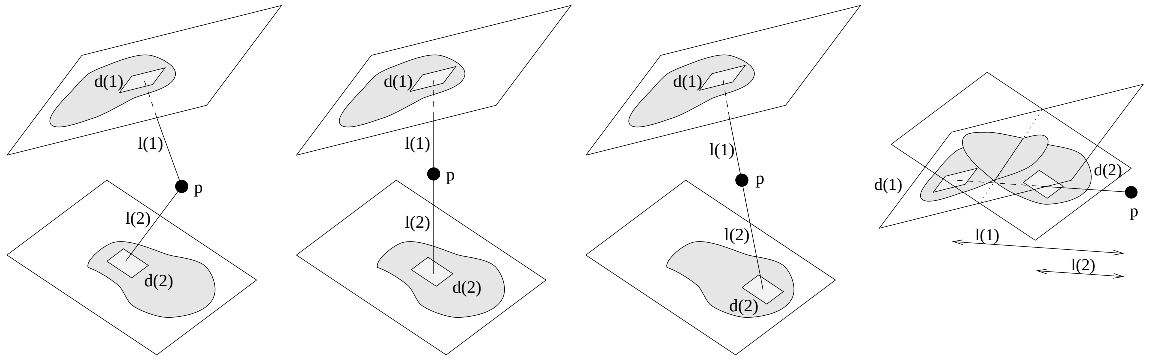
4.3 Interpolation from arbitrarily orientated planes

The next step in the construction of the surface requires interpolation of the distance field values for the object cross-sections. This interpolation is similar to the interpolation performed in transforming scan-plane based image data to a voxel-based representation. There are two main classes of interpolation techniques — those which interpolate data only from the planes which bound the point of interest, and those which interpolate data from points on any plane within some bounding volume of the point of interest. The latter class includes all of the many techniques for interpolating scattered data (for instance, radial basis functions). However, these can be quite complex and involve large processing times, which goes against the aim of producing a simple, fast surface interpolation method.

The alternative class, *bounding plane* interpolation, includes as a sub-set those techniques which apply to parallel planes (cubic interpolation is better seen as a type of bounding plane interpolation which also uses data from the set of planes bounding the inner set, than as a bounding volume interpolation). In fact, very few such techniques exist for non-parallel planes, and most of these have been developed with 3-D freehand ultrasound specifically in mind — they have already been introduced in Section 1.3.2.

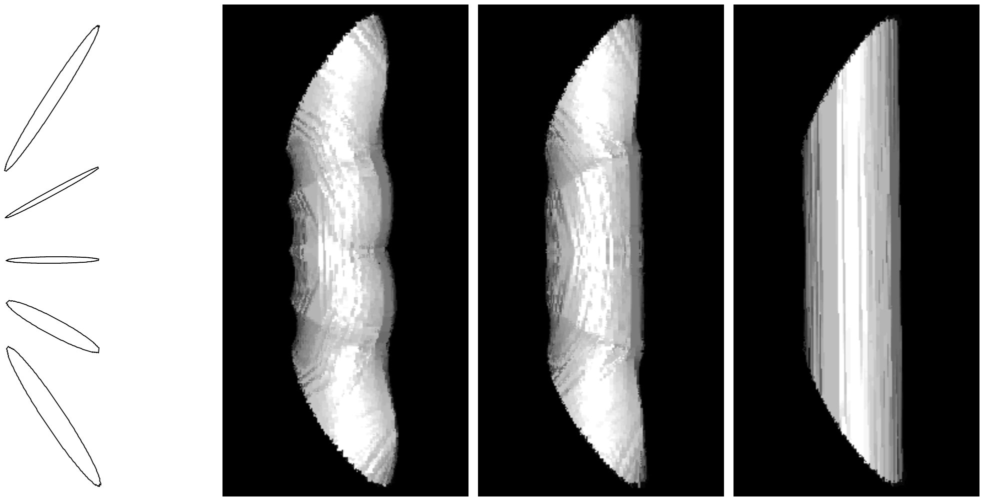
Figure 18 shows three possible techniques for interpolating between two planes bounding the point of interest[5]. In each of these cases, the distance field value at the point p is calculated by weighting the distance codes with the distances from the point, as in equation (11):

[5] For data which comes from randomly orientated planes, any given point may be bounded by many of these planes, hence the interpolating data may be from more than two planes. The techniques described here are appropriate for sequential data, in which case the two interpolating planes will be those which bound the point in sequence. Freehand ultrasound data is inherently sequential, and the manner in which it is acquired produces gradually varying, rather than randomly orientated, planes. The treatment of scans which pass over the same data point more than once (which is often the case with freehand ultrasound) is outlined later.



(a) closest point (b) average normal (c) centroids (d) overlapping

Figure 18: **Bounding plane interpolation possibilities for sequential data.** d_1 and d_2 are the sampled distance field values in each plane.



(a) original contours (b) closest point (c) avg. normal (d) centroids

Figure 19: **Effect of interpolation direction for non-parallel planes:** simple cylinder.

$$d_p = \frac{l_2 d_1 + l_1 d_2}{l_1 + l_2} \tag{11}$$

where d_p is the interpolated distance code at the point p. The effect of using each of these schemes is shown for a cylinder scanned with a fanning action in Figure 19. Although the closest points method is the simplest when applied to parallel data, it produces unacceptable artifacts when the planes are highly non-parallel. This is because the line of interpolation is bent at the interpolated point, causing the surface also to be curved. It is more natural to connect the contours with surfaces which are as straight as possible, hence the line of interpolation must also be straight. Interpolating by using the average normal to both planes corrects this tendency, but suffers from the same problems as were outlined in Section 4.1 for parallel planes.

Using the line connecting the centroids of the shapes produces the correct result for the cylinder in this example. However, this method can produce spurious results in situations were the scan planes overlap. Figure 18(d) is an example of such a situation. If the distance codes for each of the interpolating points are similar, the point p will always be positive, irrespective of the magnitude of the distances l_1 and l_2. This is equally the case whether or not l_1 is considered to have the same sign as l_2. In effect, the object "spills out" in the direction of interpolation, potentially to infinity.

We therefore require an interpolation strategy which includes the benefits of centroid interpolation, but without the "spilling out" effect, and preferably which improves on the problems mentioned in Section 4.1.

4.4 Maximal disc guided interpolation

A better interpolated surface can be produced if the direction of interpolation is allowed to vary across the cross-section. At each interpolated point, we would like to interpolate between points on the enclosing scan planes which are in similar regions of the object. This can be achieved in three steps. Firstly, a vector field is calculated for each object cross-section, where the vector at each point is directed towards the "local centroid" of the object, with a magnitude equal to the distance from that centroid. Secondly, the vector fields for the two scan planes enclosing the interpolated point are sampled near that point, and the resulting vectors combined to give a final interpolation vector. Lastly, this vector is used to sample the distance field from each scan plane, and combined using equation (11) to give the interpolated point value.

An example of a varying interpolation direction, applied to parallel slice data, is given by Moshfeghi [52]. Here a simple scheme is used to interpolate magnetic resonance angiography data, where the centroid of each separate contour is first calculated, then the connectivity between centroids on adjacent scans is determined. The vector connecting matched contours is then used to interpolate data within those contours, and the data *between* contours is interpolated with a varying direction which itself is a linear interpolation of centroid connection vectors. This works well for multiple simple shapes with clear connectivity, but not for complicated shapes, since only a single centroid is calculated for each contour.

Calculation of maximal disc set The distance field itself contains information about the cross-sections from which it might be hoped some definition of local centroid could be formed. For instance, the direction of the gradient of the distance field always points towards

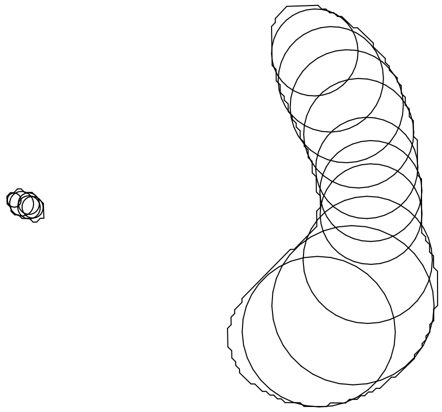


Figure 20: **Maximal disc set used in interpolation.** The cross-section is the same as in Figure 15.

the closest object point. Unfortunately, although the distance field itself is contiguous, the gradient is not — in fact for the 10-14 chamfer it is everywhere +10 except at the ridges and canyons, (i.e. the local maxima and minima), where it is between +10 and 0. In order to generate a smooth surface, the change in interpolation direction must also be smooth, so this gradient can clearly not be used directly — another representation of the object shape is required.

One possibility is the use of the multi-scale medial axis (MMA). This has been used in image registration and it is suggested that it could form the basis for object based interpolation [27]. However, the MMA is calculated directly from the grey-scale information in the image, whereas we are interested in interpolating a surface derived from segmented cross-sections.

It was Blum [9] who first suggested an object based representation which consisted of a set of discs just enclosed by an object. The locus of centres of these discs forms the *symmetric axis*, which has also been termed the *skeleton* or *medial axis* of the object. It is also possible to represent an object with any other shape, for instance an ellipsoid [40]. Using ellipsoids can reduce the number of elements required for a given representational accuracy, but is over complicated for our purpose.

Rather conveniently, this set of discs can be derived from the maxima of the distance transform which has already been calculated for each object cross-section. Firstly, all possible internal discs which touch the cross-section at some point, and are not entirely enclosed within another disc (the *maximal set* or MSD) are found. Secondly, this set is reduced to eliminate any discs which are entirely enclosed by the union of other discs, giving finally the *minimal set of maximal discs* (MSMD). This process is entirely loss-less, and the MSMD can be used to exactly reconstruct the original cross-section.

For city-block and chess-board distances, the MSD can be found simply from the local maxima of the distance field. The derivation of the MSD for chamfer 3 × 4 codes is given by Arcelli [4]. Essentially, any pixel p is a disc centre if the following equation is satisfied :

```
while $A$ exists do
    $m = \max_{d \in A}(radius(d))$
    remove $m$ from $A$
    add $m$ to $B$
    for all $j \in A$ do
        $l =$ distance from $m$ to $j$
        $r = radius(m) - 0.5 \times radius(j)$
        if $l^2 < r^2$ then
            remove $j$ from $A$
```

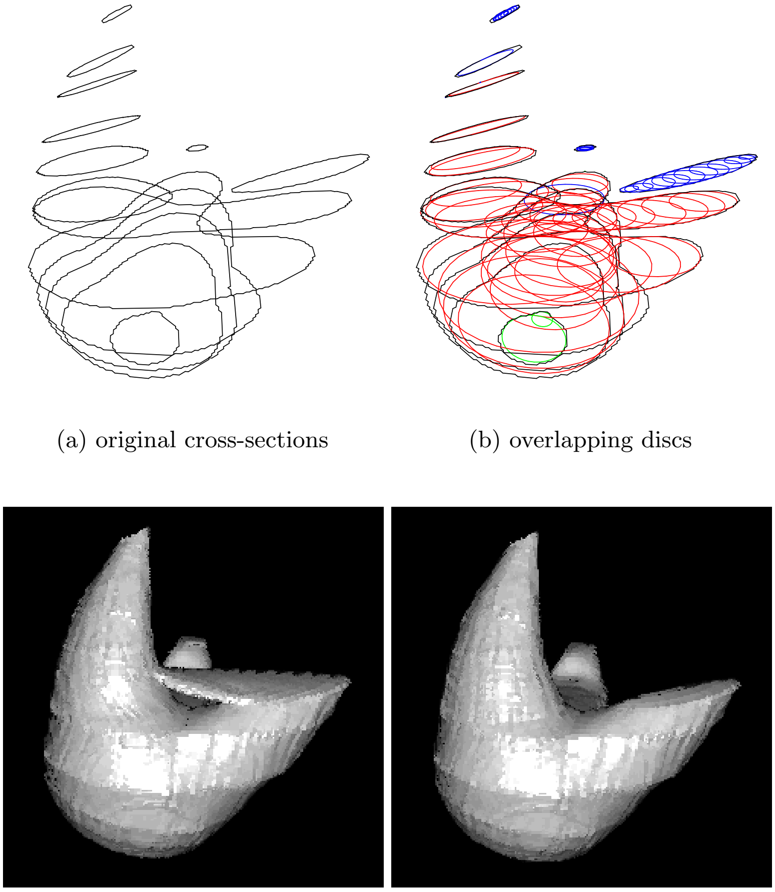
Figure 21: **Algorithm to reduce set of maximal discs.** A is the original set of maximal discs, B is the final reduced set. Discs are removed if they protrude from a previously chosen disc by less than half of their radius.

$$p > q_{ij} - \begin{cases} 10, & i,j \in (\ 1,0 \quad -1,0 \quad 0,1 \quad 0,-1\ ) \\ 14, & i,j \in (\ 1,1 \quad 1,-1 \quad -1,1 \quad -1,-1\ ) \end{cases} \tag{12}$$

where i, j are the pixel coordinates referenced to p, and q_{ij} is the distance field value of the pixel at i, j. However, it is possible for discs in the MSD to be completely enclosed by the union of several other discs, making them redundant in terms of shape representation. Several methods have been suggested for reducing this set to the MSMD. Nilsson [57] considers border coverage and builds a "relation table" which relates border pixels to maximal discs, which is then reduced by iteration. Borgefors [13] suggests a simpler technique in which a second image is created, each pixel representing the number of maximal discs which cover it. Redundancy can then be found by checking for discs which have a value greater than one for every contained pixel. These discs are then removed and the process iterated.

In this case we do not require a set of discs which completely cover the object — the object border position is already guaranteed by the distance field which will be interpolated. We simply need enough discs to adequately define the local centroid for each point. Having a smaller set of discs than the MSMD also decreases the processing time when calculating the interpolation direction for each point. The chosen method for reducing the disc set, outlined in Figure 21, preserves the discs which best approximate the geometry of the object. The resulting discs for the cross-section in Figure 15 are shown in Figure 20.

Selection of discs for interpolation If all the discs in the resulting disc set are used in the interpolation, this results in very strong connectivity between neighbouring scan planes. In fact, every contour on one plane is guaranteed to be connected to at least one contour on the



(a) original cross-sections

(b) overlapping discs

(c) surface using all discs

(d) surface using reduced set

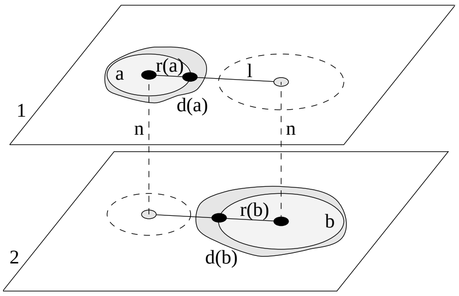
Figure 22: **Choice of disc set for interpolation :** Jester's hat object. Blue discs are used only for interpolation to the lower scan plane, green to the upper scan plane, and red for both planes.

other plane. Whilst this may be an interesting result, in practice there are many situations where it is undesirable to connect every contour to another one. Figure 22 demonstrates such a situation for the jester's hat object shown in Figure 3.

This behaviour can be modified by careful selection of the set of discs for each plane which are used in the interpolation with another neighbouring plane. It is natural to determine connectivity on whether two shapes "overlap"[6], and most methods do in fact use this criteria. This can be extended to discs by considering whether the part of the object cross-section represented by a disc overlaps with the part of the object cross-section on the neighbouring plane represented by any of its discs.

Figure 23 shows a simple example with one disc representing each object cross-section. The addition of the sampled distance field values d_a and d_b ensures that discs will be used

[6] For non-parallel planes, "overlapping" is taken to mean that the projection of the objects onto a plane normal to the average plane normal intersects.



```
for all $a \in B_1$ do
  $m$ = projection of $a$ on plane 2
  for all $b \in B_2$ do
    $l$ = distance from $m$ to $b$
    calculate $d_a$ and $d_b$ for these discs
    $r = r_a + d_a + r_b + d_b$
    if $l^2 < r^2$ then
      add $a$ to $C_1$
      add $b$ to $C_2$
```

Figure 23: **Calculation of disc overlap.** r_a, r_b are the radii of the discs a, b. d_a, d_b are the distance fields sampled at the closest points on the circumference of the discs. l is the distance between the centre of the discs measured in either plane. C_1 and C_2 are the resulting disc sets.

if the corresponding objects do in fact overlap. The set of discs for each plane used in the interpolation, C, is calculated for each pair of planes using the algorithm of Figure 23.

Calculation of interpolation direction Once a set of discs has been created for a given object, the "local centroid" $\vec{c}$ can be calculated for any point p using equation (13). This is a vector which indicates an approximate direction and distance to the nearest object points, but which is continuous across the object.

$$\vec{c} = \frac{1}{\sum_{d \in C} \frac{r_d}{\left|\vec{pk_d}\right|^2}} \sum_{d \in C} r_d \frac{\vec{pk_d}}{\left|\vec{pk_d}\right|^2} \tag{13}$$

where k_d is the centre of disc d and r_d is its radius. C is the chosen set of interpolating maximal discs for this pair of scan planes. The vector field resulting from this process is shown in Figure 24. It can be seen from this figure that the use of only a few discs to represent the object has limited effect on the smoothness of the magnitude, and only affects the phase in regions which are guaranteed to be inside the object, and therefore will not affect the smoothness of the surface.

For each interpolated point in between two planes, the local centroid $\vec{c}$ for each plane is calculated at the intersection of the average plane normal $\hat{n}$ with these planes. These local centroid vectors can then be combined to give an interpolation direction which better represents the change of object shape and position, as in Figure 25. The most natural way to do this would be simply to join the local centroids on each plane, as in Figure 25(a).

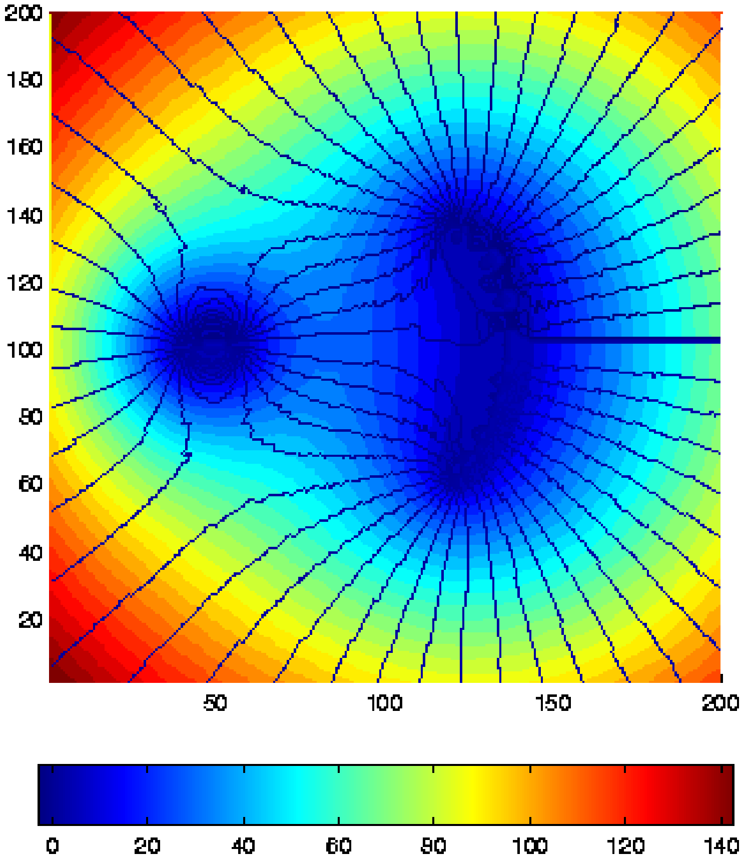
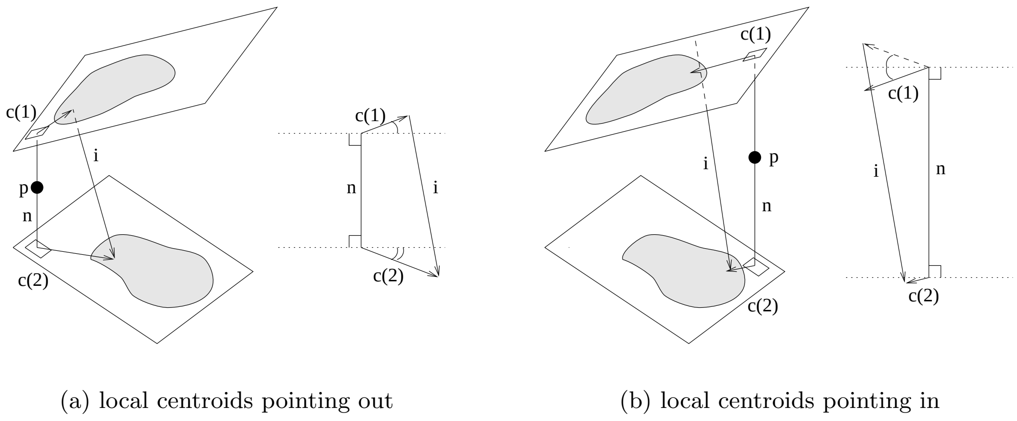


Figure 24: **Interpolation direction from maximal discs.** The colour gives the magnitude of the interpolation vector, in pixels. The radial contours give the phase transition points for these vectors. The cross-section is the same as in Figure 15.



(a) local centroids pointing out

(b) local centroids pointing in

Figure 25: **Construction of interpolation direction from local centroids.**

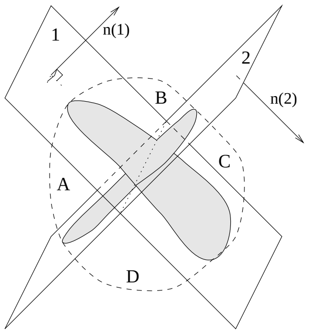


Figure 26: **Object scanned with two orthogonal scan planes.**

However, this can lead to the "spilling out" artifact mentioned earlier in situations where the local centroids point inwards with respect to the average plane normal $\hat{n}$. The problem is easily rectified by always ensuring that the local centroids point outwards — for situations where this is not the case, the reflection of the local centroid about a plane normal to $\hat{n}$ is used instead, as in Figure 25(b).

Interpolation of distance fields and volume measurement The last stage in the process is the interpolation of the distance transformed image. The calculated interpolation vector for a point p is used to define a line passing through this point. The distance field is then sampled at the intersections of this line with each bounding plane (bi-linear interpolation is used within the scan plane from the four neighbouring distance field values). Distance weighting values l_1 and l_2 are also measured along this line, and the final interpolated value is then given by equation (11). The volume can be calculated from the sum of the positive interpolated points, multiplied by the volume represented by each point.

There are two additional complications which arise because of the possibility of the scan planes intersecting each other. An extreme version of this situation, with two orthogonal planes, is shown in Figure 26. The first problem is to decide which regions should be inter-

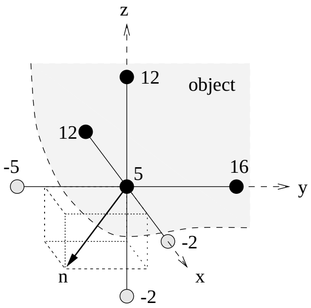


Figure 27: **Calculation of surface point normal.**

polated from these planes. The four regions in the example are all equally "enclosed" but presumably only two of them should be interpolated. Since the scan planes are sequential, we would like to interpolate only the area which was swept out during the scan. This can be deduced from the direction of the normals to each plane — regions which are "in front" of one plane and "behind" the other plane are interpolated (corresponding to B and D in Figure 26).

Secondly, if the intersection of the scan planes passes through the object then part of the swept volume between these scan planes must have already been swept out by previous planes. The volume for these regions should be negative, else the total volume will be an overestimate. This can once again be resolved by examining the plane normals — if the normals point more from the first towards the second plane, then the volume is positive (region B), else it is negative (region D), i.e.

$$\mathrm{sign(volume)} = (\vec{n_1} \cdot \vec{n_{avg}}) > 0 \tag{14}$$

where $\vec{n_{avg}}$ is the average of the plane normals $\vec{n_1}$ and $\vec{n_2}$.

4.5 Storage and display of surface

It is well known that good definition of surface normals can vastly improve the display of a surface. This is because it is the surface normal which determines the grey-scale value in the final image — and images with a gradually varying grey-scale (i.e. accurate definition of surface normals) look much "smoother" even if the actual pixel resolution is poor. For this reason, triangulated surfaces are sometimes improved by defining the vertex normals directly from the grey-scales in the original image. This results in an apparently better quality image, though there is little justification for defining the normals in this way.

Interestingly, in this case the interpolation results in *distance* values which indicate the location of the actual object surface more precisely than could be defined by simply thresholding them at zero. Surface normals can be deduced directly from these interpolated values which actually represent an increased *accuracy* of surface definition, rather than just a better looking image.

Figure 27 shows how a surface normal can be extracted from the voxels surrounding a central point. The numbers represent the interpolated distance field values for that voxel. In each dimension, the surface normal vector is simply the gradient of these values, with the convention that negative slopes indicate an outward pointing normal.

A simple method for displaying the surface is to generate, for each surface point, a square positioned at that point, and with an orientation appropriate to the local surface gradient. These squares are simple geometric objects which can be efficiently generated and manipulated with standard graphics hardware. Provided they are large enough to cover the gaps due to the surface point resolution, the result is a smooth surface which can be manipulated in real time. All of the surface images in this report have been produced by this technique.

With this display technique, it should be possible to avoid storage of the interpolated distance field values in a voxel array — the volume being incremented and surface points stored as the object is traversed. Currently, however, a voxel representation *is* being used as an intermediate step between interpolation and surface point extraction, the elimination of this memory overhead being a current research topic.

4.6 Overview of method

An overview of the whole process is given in Figure 28. For non-parallel planes, one of the main factors determining the speed is the "intersection" calculation, which must be done four times for each interpolated point. This can be efficiently implemented from the solution for i_x, i_y and i_z of equations (15) :

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 0 \end{bmatrix} + i_z \begin{bmatrix} n_x \\ n_y \\ n_z \\ 0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} i_x \\ i_y \\ 0 \\ 1 \end{bmatrix}, \quad \text{i.e.}$$

$$\begin{aligned} p_x + i_z n_x &= \mathbf{M_{11}} i_x + \mathbf{M_{12}} i_y + \mathbf{M_{14}} \\ p_y + i_z n_y &= \mathbf{M_{21}} i_x + \mathbf{M_{22}} i_y + \mathbf{M_{24}} \\ p_z + i_z n_z &= \mathbf{M_{31}} i_x + \mathbf{M_{32}} i_y + \mathbf{M_{34}} \end{aligned} \tag{15}$$

where $\vec{p}$ is the point in free space, $\hat{n}$ is the unit interpolation direction vector and $\mathbf{M}$ is the 4×4 plane position matrix, as defined in [63]. i_x, i_y then give the plane intersection in pixels, and i_z the distance to the plane along the interpolation direction.

```
$volume := 0$
for each segmented frame do
    calculate bounding rectangle for contours
    calculate distance transform for rectangle
    extract set of maximal discs
    if not first segmented frame then
        calculate disc overlap with last frame
        $\hat{n}$ = average plane normal
        for each $voxel$ between $frames$ do
            $\vec{i_1}$ = intersect $\hat{n}$ at $voxel$ with frame 1
            $\vec{c_1}$ = local centroid field at $\vec{i_1}$
            $\vec{i_2}$ = intersect $\hat{n}$ at $voxel$ with frame 2
            $\vec{c_2}$ = local centroid field at $\vec{i_2}$
            $\hat{c}$ = interpolation direction from $\vec{c_1}$ and $\vec{c_2}$
            $\vec{i_1}$ = intersect $\hat{c}$ at $voxel$ with frame 1
            $d_1$ = distance field at $\vec{i_1}$
            $\vec{i_2}$ = intersect $\hat{c}$ at $voxel$ with frame 2
            $d_2$ = distance field at $\vec{i_2}$
            $d = \left(\left|\vec{i_2}\right| d_1 + \left|\vec{i_1}\right| d_2\right) / \left(\left|\vec{i_1}\right| + \left|\vec{i_2}\right|\right)$
            if $d > 0$ then
                if $(\hat{n_1} \cdot \hat{n}) > 0$ then $volume = volume + 1$
                else $volume = volume - 1$
scale $volume$ for voxel size
```

Figure 28: **Overview of surface interpolation process.**

5 Results

5.1 Simulated scanning of geometrical objects

In order to verify the volume measurement and surface reconstruction techniques, several objects were "scanned" in simulation, using the tool described in Section 2.2. Six different techniques were tested:

- Linear planimetry.
- 2-D cubic planimetry.
- 3-D cubic planimetry.
- Shape based interpolation using average plane normals.
- Shape based interpolation using object centroids.
- Maximal disc guided shape based interpolation.

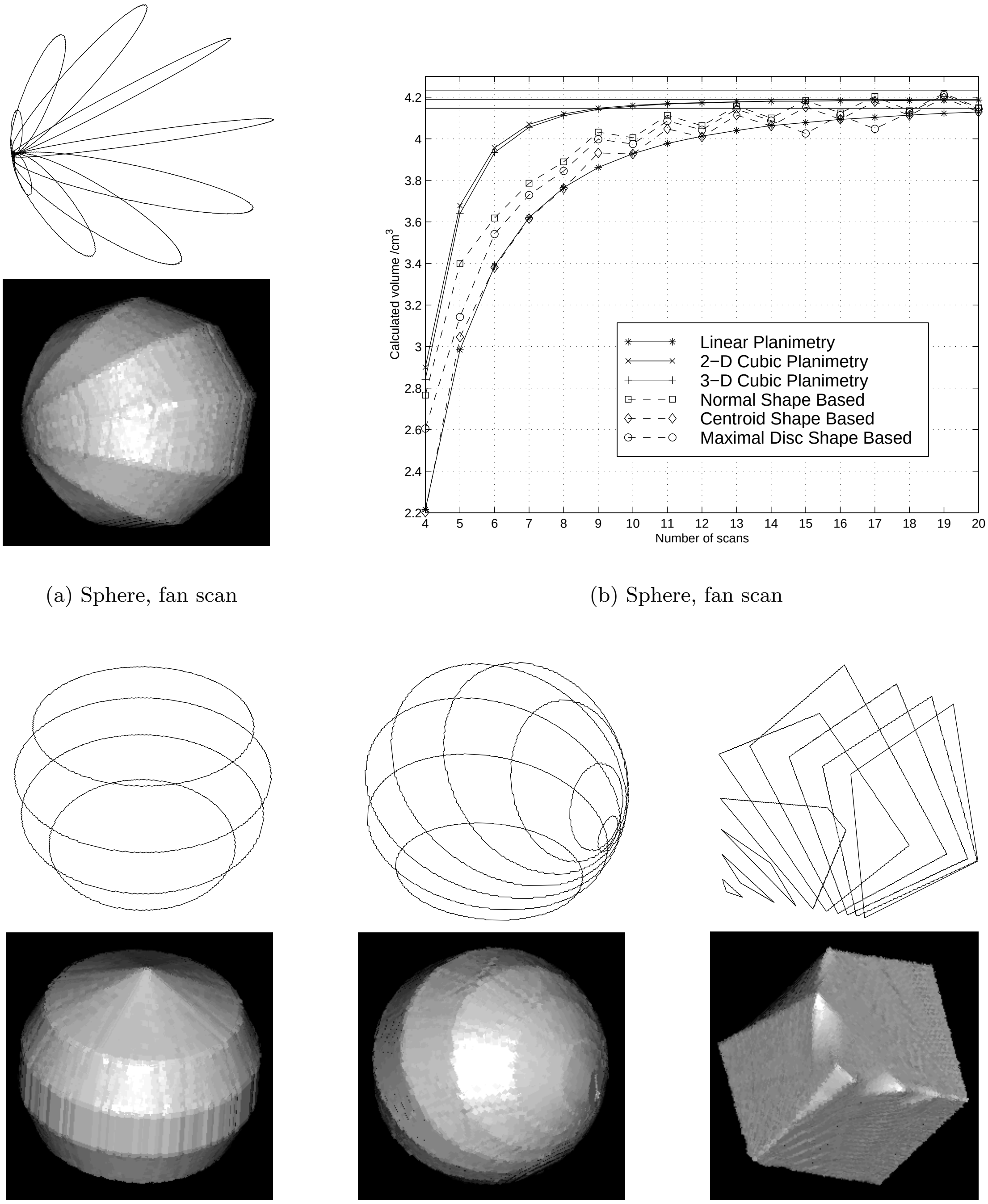
Both the segmentation accuracy and the positioning of the first and last scan planes also have an effect on the volume measurement, independent of the technique used. The segmentation error in the simulation was negligible since the cross-sections were defined directly from the original objects. The positioning of the first and last planes was deliberately chosen in each case such that they just intersected the object.

The size of each pixel in the scan planes was 0.012cm. Objects were scaled to just fit inside a cube of 2cm × 2cm × 2cm, hence length measurements had an error of approximately ±0.3%. Surface reconstructions were performed at half this resolution, giving volume errors of $3 \times 2 \times 0.3 = \pm 1.8\%$. Planimetry is inherently less dependent on resolution, being based on area measurements, giving volume errors of ±0.6%.

The time taken for each measurement was approximately 20ms for the cubic planimetry methods, and approximately 7s for maximal disc guided shape based interpolation, the latter being irrespective of the number of scans. Processing was performed on a Silicon Graphics Indigo 2 workstation, and a 166MHz Pentium MMX running Linux, with similar results on each platform. The surface interpolation process required less than 4Mb of memory, once the segmentation had been performed manually.

Different test objects and scan patterns Each of the test objects was scanned with linear, fan or freehand sweeps. The number of scans in the sweep was varied from 4 to 20 (including the first and last scans) to show the change in volume measurement as the sweep "resolution" increased. The results are shown in Figures 29, 30, 31 and 32. The solid horizontal lines on the graphs show the actual volume of the object and a margin of ±1% around this value. Object contours and surface reconstructions are shown with the minimum number of scans for which the cubic planimetry volume is within this margin. Table 1 shows this number of scans for each object, and the percentage error for 20 scans.

Registration and unsteady hand effects In practice, the scan planes are generally not evenly spaced and their location and orientation is not known precisely, even with good calibration. In order to examine the effects of these errors, the baseball glove was scanned



(a) Sphere, fan scan

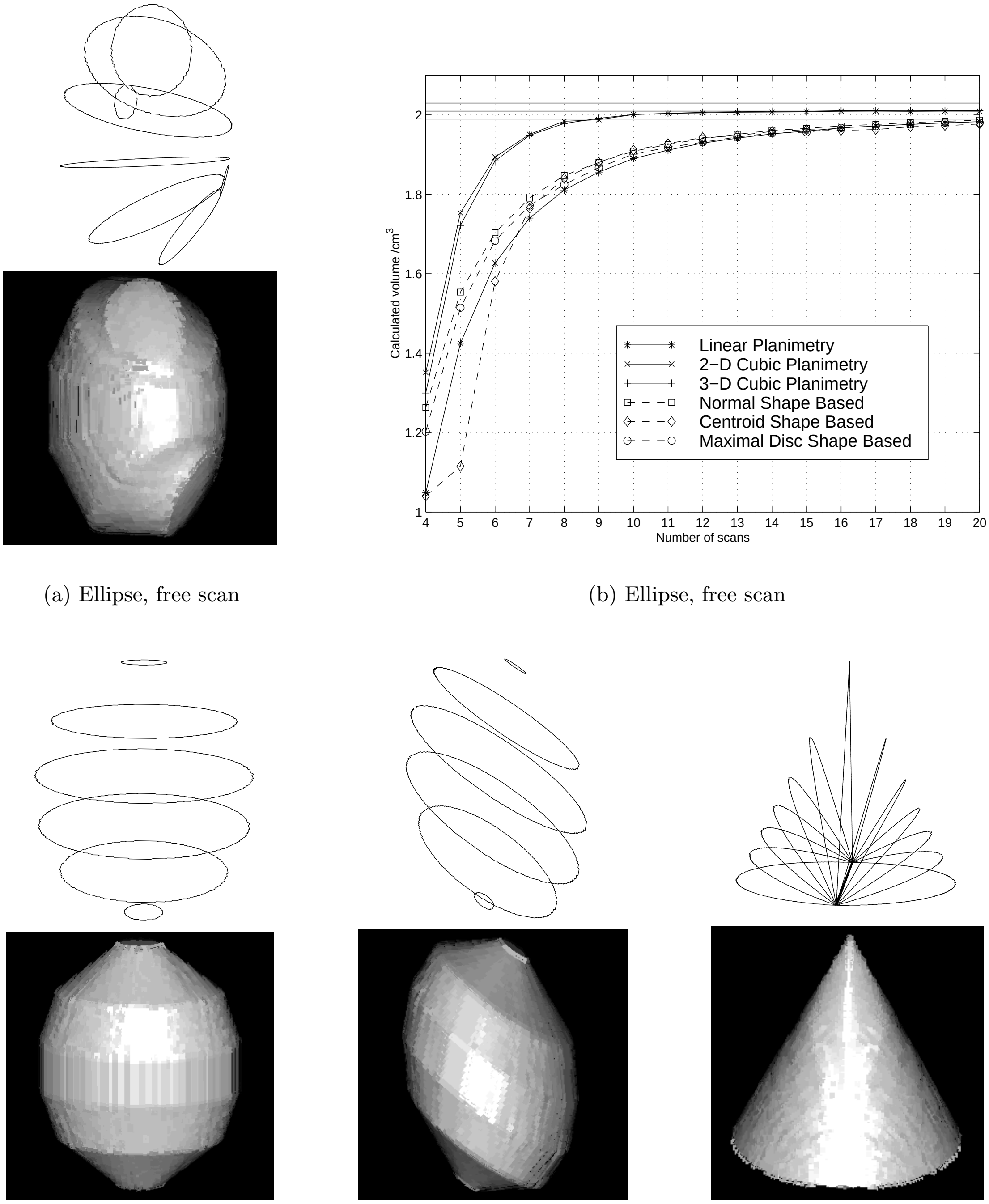
(b) Sphere, fan scan

(c) Sphere, linear scan

(d) Sphere, free scan

(e) Cube, free scan

Figure 29: **Simulation Results:** Sphere and cube.



(a) Ellipse, free scan

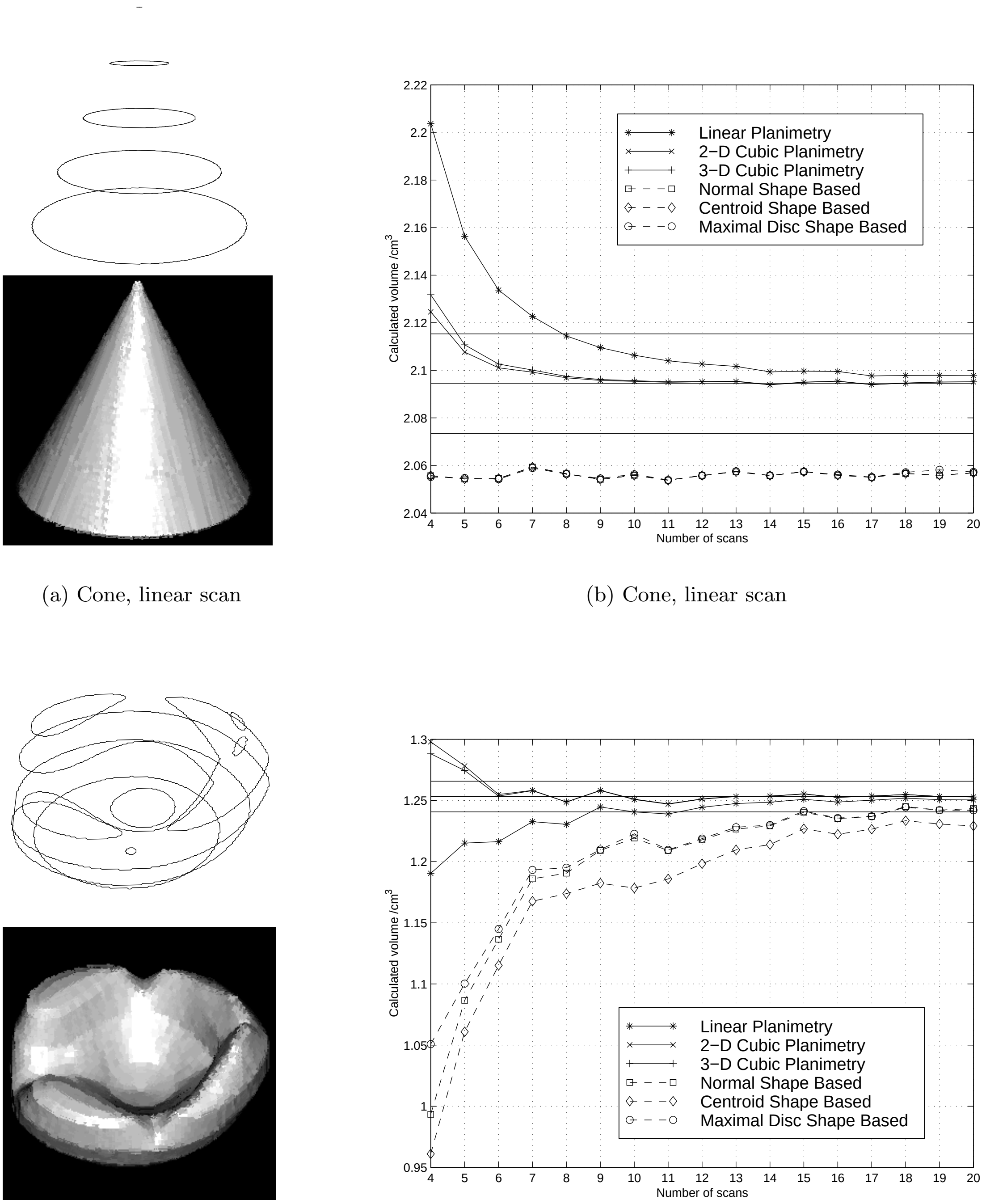
(b) Ellipse, free scan

(c) Ellipse, linear scan

(d) Ellipse, oblique scan

(e) Cone, fan scan

Figure 30: **Simulation Results:** Ellipse and cone.



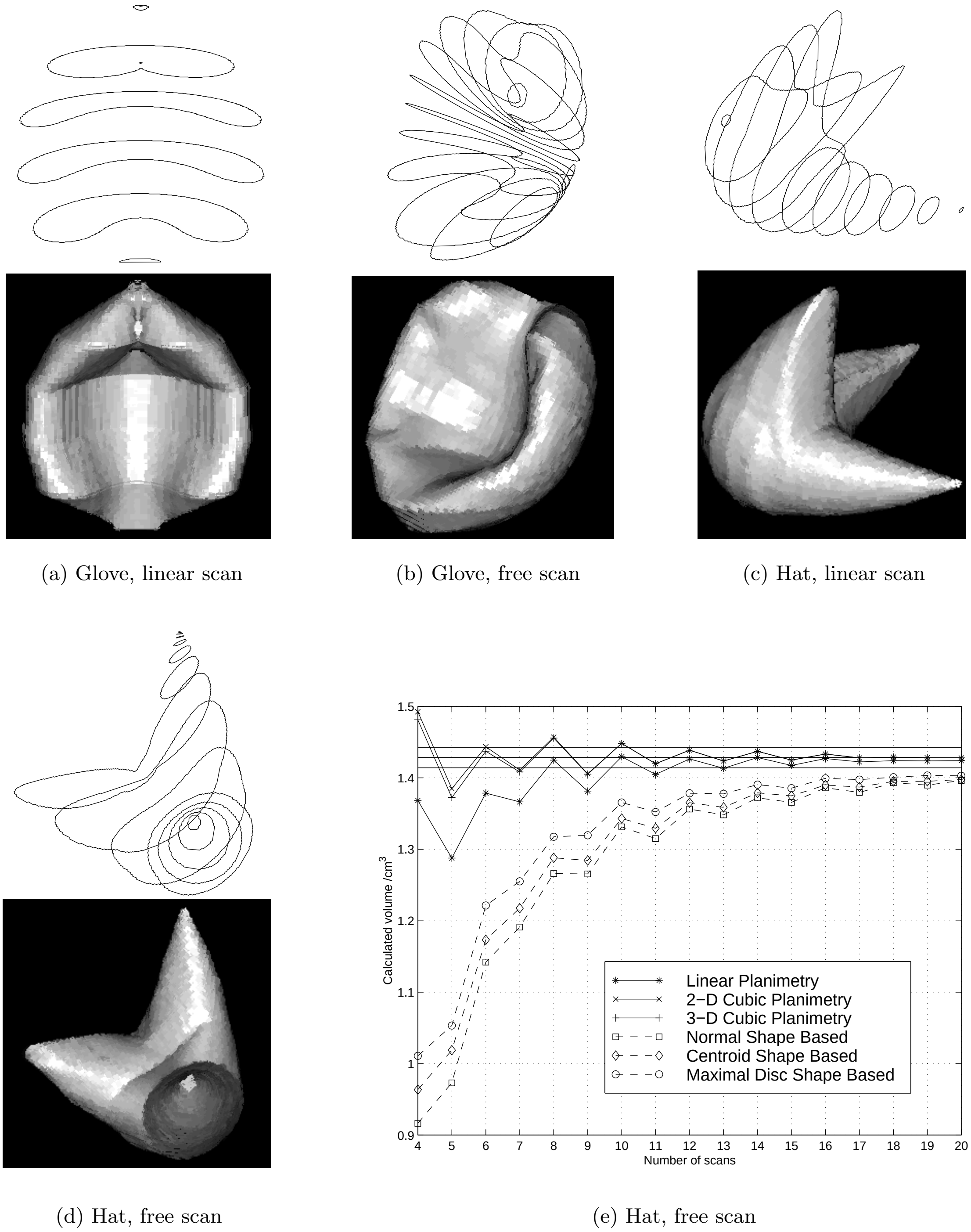
(a) Cone, linear scan

(b) Cone, linear scan

(c) Glove, linear scan

(d) Glove, linear scan

Figure 31: **Simulation Results:** Cone and “baseball glove”.



(a) Glove, linear scan

(b) Glove, free scan

(c) Hat, linear scan

(d) Hat, free scan

(e) Hat, free scan

Figure 32: **Simulation Results:** “Baseball glove” and “Jester’s hat”.

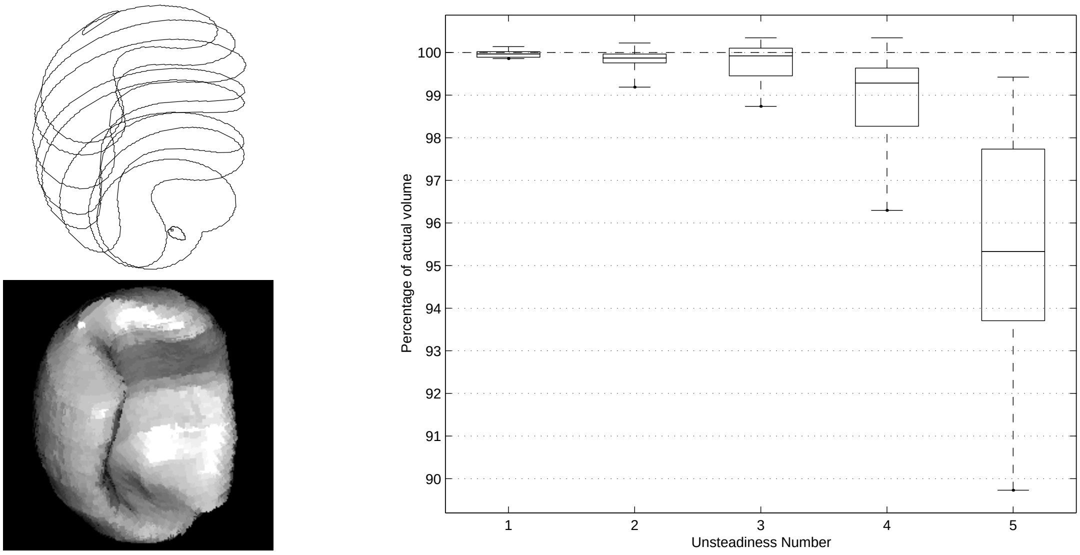
Object	**Scan**	**Planes for < 1% error**						**% error at 20 planes**					
		1	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**
Sphere	linear	11	6	6	20	20	20	−0.2	+0.1	+0.1	−0.8	−0.8	−0.8
	fan	-	9	10	19	19	19	−1.4	−0.1	−0.1	−0.9	−1.4	−1.1
	free	20	9	9	15	16	16	−0.9	−0.1	−0.1	+0.0	−0.1	−0.1
Ellips.	linear	11	6	6	15	15	15	−0.3	−0.0	−0.0	−0.5	−0.5	−0.5
	oblique	11	6	6	17	17	17	−0.2	+0.1	+0.1	−0.6	−0.6	−0.6
	free	-	9	9	-	-	-	−1.3	−0.0	+0.1	−1.1	−1.6	−1.4
Cone	linear	8	5	5	-	-	-	+0.2	+0.0	+0.0	−1.8	−1.8	−1.8
	fan	-	9	9	17	17	17	−1.8	−1.3	−1.3	−1.8	−1.6	−1.7
Cube	free	12	8	8	14	14	12	−0.5	−0.0	−0.0	−0.5	−0.5	−0.3
Glove	linear	12	6	6	15	15	15	−0.3	+0.1	+0.1	−1.2	−1.0	−0.9
	linear	9	6	6	15	-	15	−0.2	−0.0	−0.0	−0.8	−1.9	−0.9
	free	-	16	15	-	-	-	−2.4	−0.6	−0.3	−1.5	−1.5	−1.5
Hat	linear	10	9	9	-	-	-	−0.3	−0.0	−0.0	−2.2	−2.2	−1.8
	free	-	14	14	-	-	18	−1.9	−0.0	−0.0	−1.4	−1.9	+0.1

Table 1: **Simulation Results :** All objects. **1**: Linear planimetry, **2**: 2-D cubic planimetry, **3**: 3-D cubic planimetry, **4**: Normal shape based, **5**: Centroid shape based, **6**: Maximal disc shape based. Missing values imply > 20 scans. The values for the cone fan scan were the result of the symmetry of the situation — odd numbers of sweeps gave more accurate volumes than even.

with 10 linear scans, in the same pattern as for Figure 32(a). Random errors were applied to the location of the scans, first before scanning the object (unsteady hand simulation) then after scanning the object (mis-registration simulation). Errors were added to all three position distances and all three orientation angles simultaneously. Five "scales" of errors were used in each case, corresponding to the numbers on the x-axis of the graphs in Figures 33 and 34, as follows :

- **1** ±0.02cm location, ±1° orientation.
- **2** ±0.05cm location, ±2.5° orientation.
- **3** ±0.1cm location, ±5° orientation.
- **4** ±0.2cm location, ±10° orientation.
- **5** ±0.4cm location, ±20° orientation.

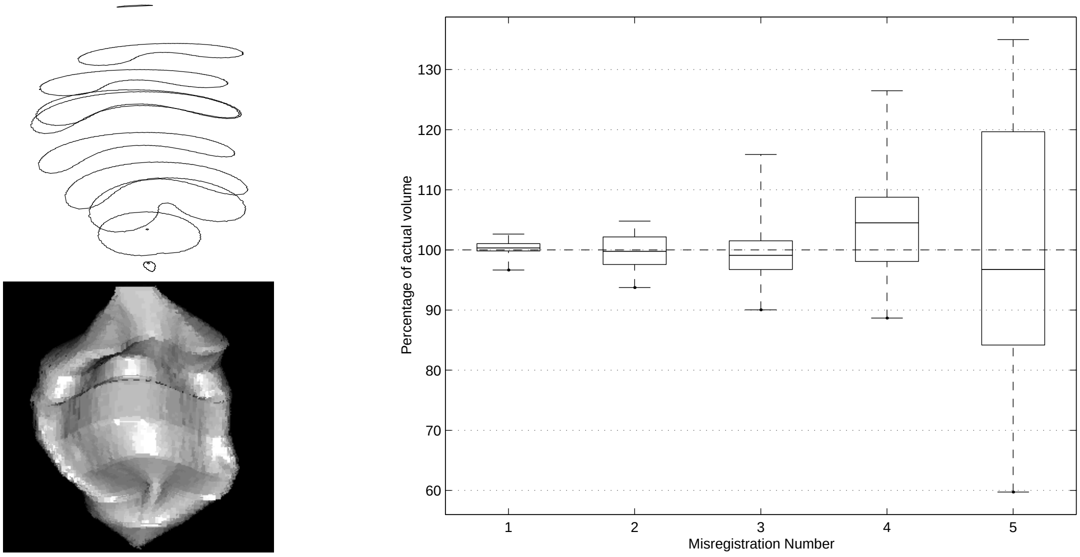
The object dimensions were approximately 2cm × 2cm × 2cm, hence these represent reasonable to extremely bad errors. 2-D cubic planimetry volumes were measured for 20 different samples at each scale of error — the graphs show the quartiles (rectangular boxes) and minimum and maximum values (dotted vertical lines) for each of these distributions. The distributions were similar for the other volume measurement techniques.



(a) Sweep and object

(b) Distribution of cubic planimetry volumes

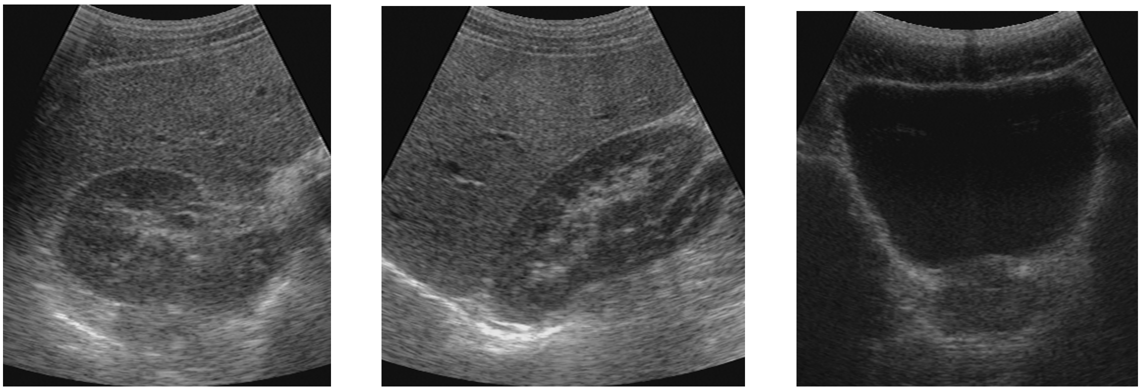
Figure 33: **Simulation Results:** Unsteady sweeps of "Baseball glove". Rectangles show quartile values, dotted lines show minima and maxima.



(a) Sweep and object

(b) Distribution of cubic planimetry volumes

Figure 34: **Simulation Results:** Mis-registered sweeps of "Baseball glove". Rectangles show quartile values, dotted lines show minima and maxima.



(a) Kidney, transverse

(b) Kidney, longitudinal

(c) Bladder and prostate

Figure 35: **Typical *in-vivo* ultrasound B-scans.**

5.2 *In-Vivo* results

The same six techniques were also tested on data measured with the free-hand 3-D ultrasound system described in Section 2.1. These measurements have the disadvantage of including both registration and segmentation errors, but do therefore represent volume measurements achievable in practice. Three areas of the human anatomy were examined:

- Kidney.
- Hepatic system.
- Bladder.

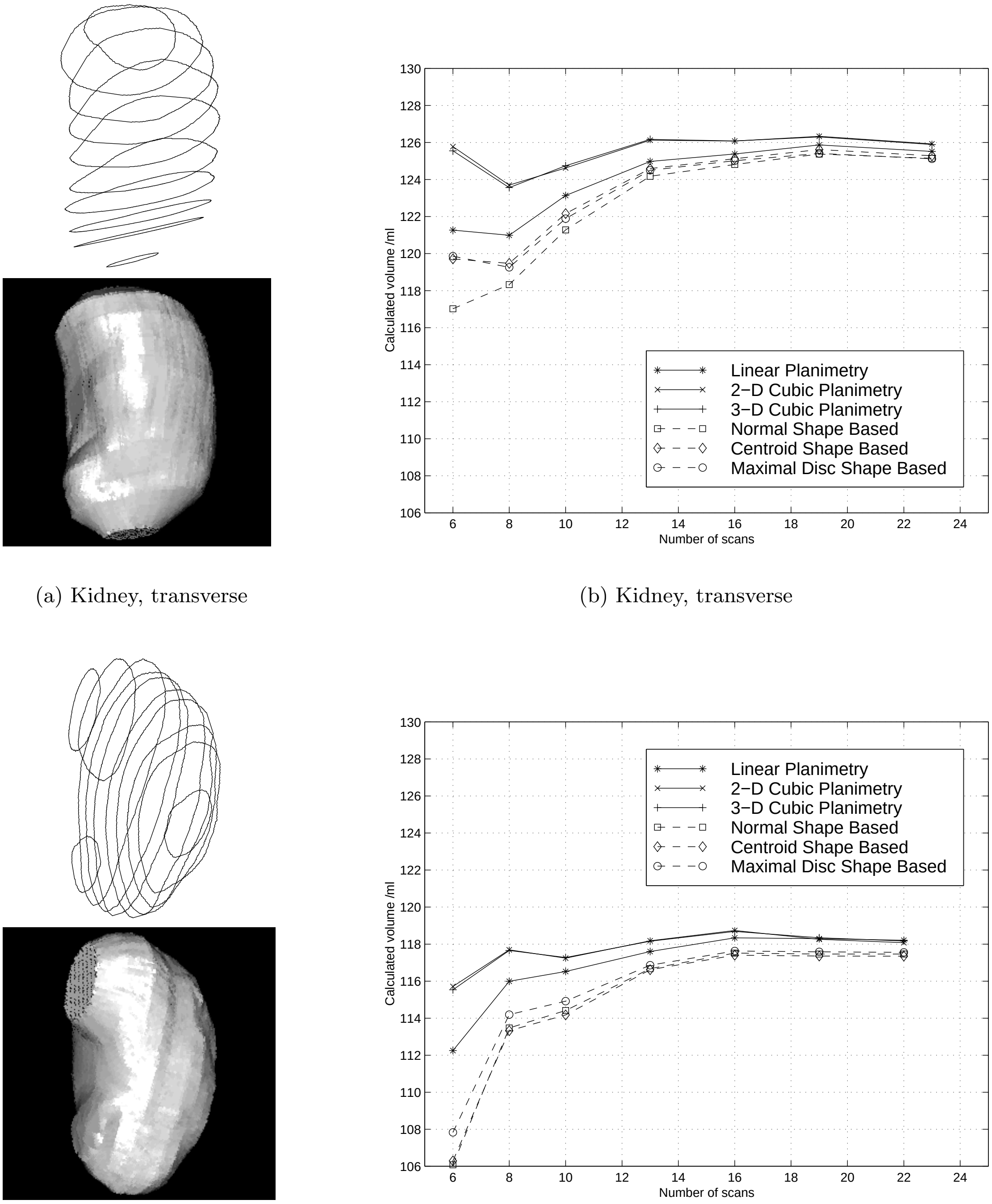
Typical B-scans for the kidney and bladder are shown in Figure 35. The scan plane pixel size for these scans was 0.035cm, hence for typical volumes of 150ml the volume errors due to resolution were ±0.7% for planimetry and ±2.1% for surface reconstruction (performed at half the in-plane resolution, as with the simulations).

5.2.1 Human kidney

Four scans were performed of the same kidney, two using a transverse and two a longitudinal scanning pattern. Each of these was then segmented with approximately 25 cross-sections, spaced reasonably evenly and ensuring that the organ extremities were sufficiently covered. Each set of segmented object cross-sections was then gradually reduced, closest ones being removed first, and volume measurements made at each step.

The actual volume of the kidney is not known — however the volume measurements should agree both for the varying number of scans within a sweep, and across each of the sweeps. Graphs of the volume measurements for each of the six techniques are shown in Figure 36, for transverse and longitudinal scans. Object cross-sections and surface reconstructions are included for ten scans.

Figure 37 shows two views of the surface reconstructions for each of the sweeps, again for ten scans. The volumes calculated for these scans are contained in Table 2.



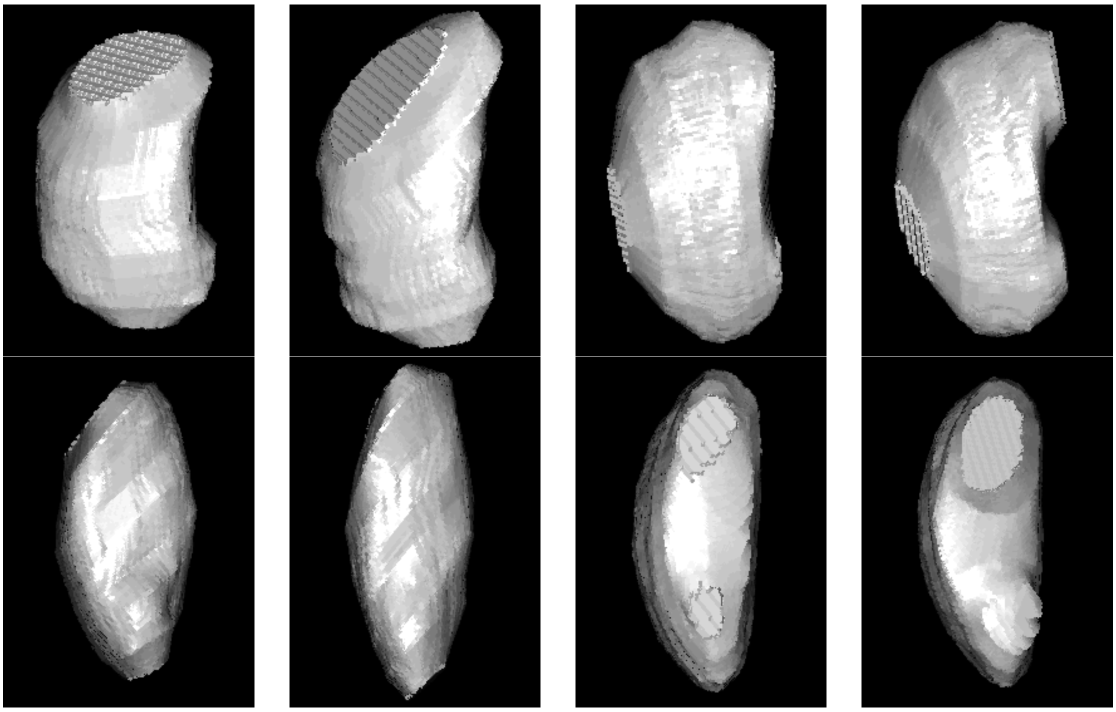
(a) Kidney, transverse

(b) Kidney, transverse

(c) Kidney, longitudinal

(d) Kidney, longitudinal

Figure 36: ***In-Vivo* Results:** Human kidney.

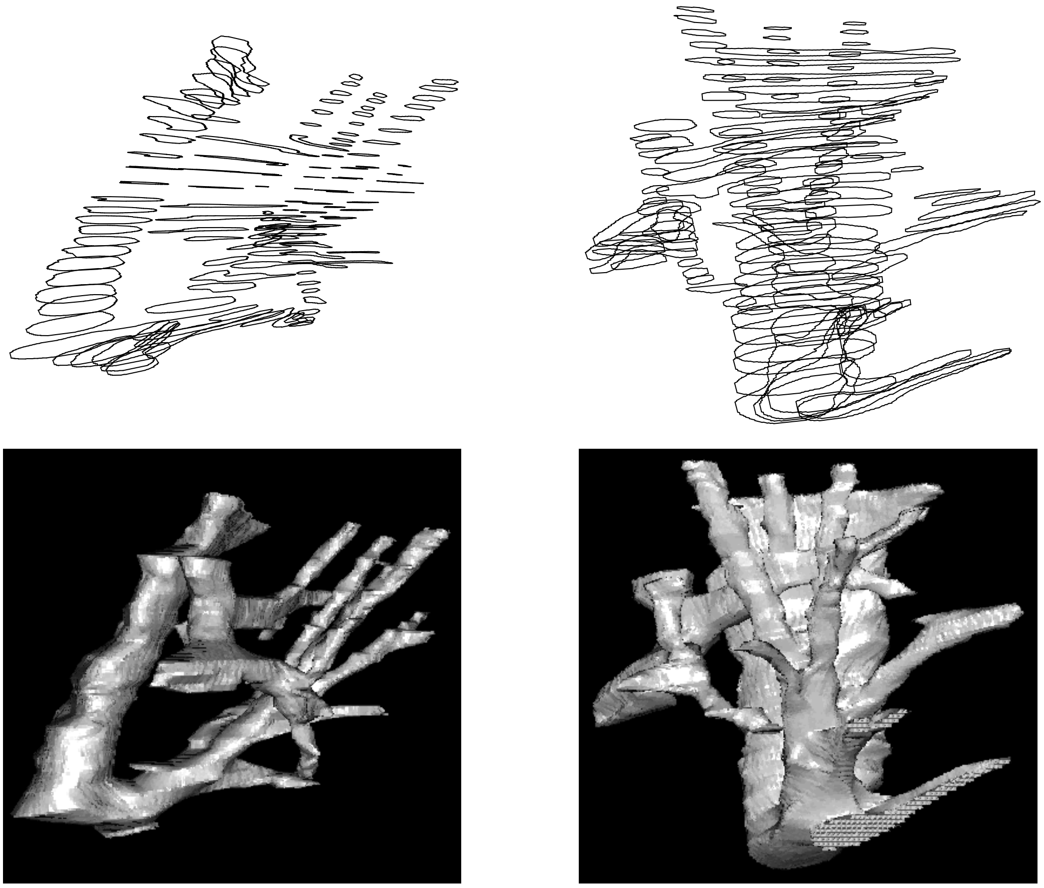


(a) 1st transverse (b) 2nd transverse (c) 1st longitudinal (d) 2nd longitudinal

Figure 37: ***In-Vivo* Results:** Comparison of kidney surfaces.

Sweep	**Scan**	**Volume measurement in ml**					
		1	**2**	**3**	**4**	**5**	**6**
Transverse	1	127.2	128.4	128.3	125.7	126.2	126.2
	2	123.1	124.6	124.7	121.3	122.2	121.9
Longitudinal	1	123.9	125.7	125.4	122.0	122.1	122.6
	2	116.5	117.2	117.3	114.4	114.2	114.9

Table 2: ***In-Vivo* Results:** Human kidney, 10 scans. **1**: Linear planimetry, **2**: 2-D cubic planimetry, **3**: 3-D cubic planimetry, **4**: Normal shape based, **5**: Centroid shape based, **6**: Maximal disc shape based.



(a) Hepatic, side view

(b) Hepatic, front view

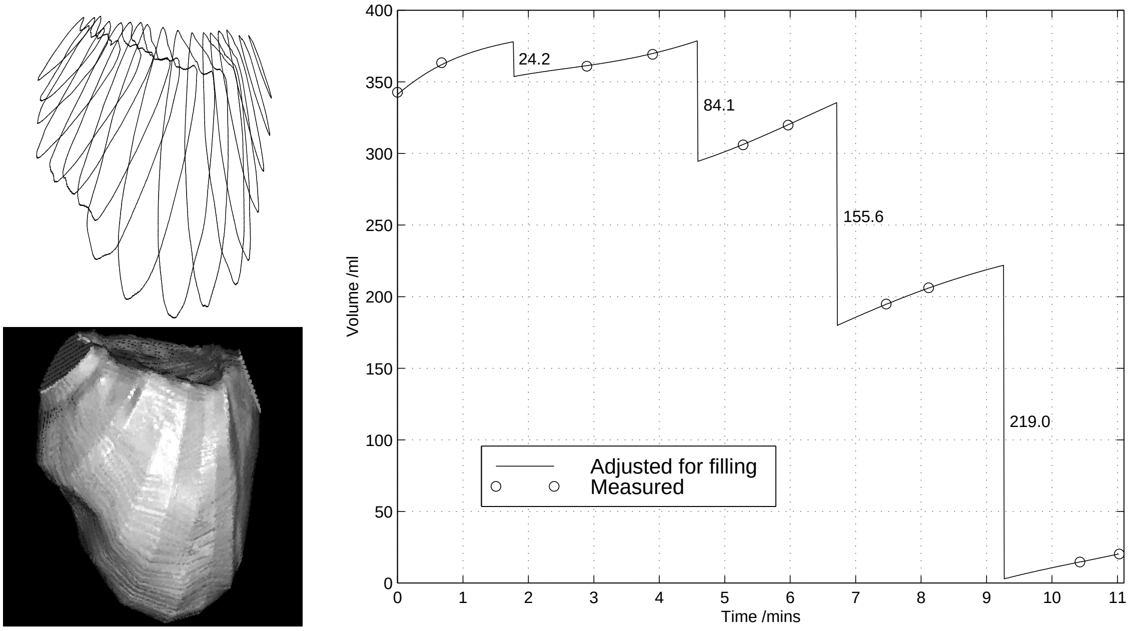
Figure 38: ***In-Vivo* Results:** Hepatic system.

5.2.2 Human hepatic system

Although of little importance for volume measurement, the hepatic system provides a good test for surface reconstruction. Figure 38 shows part of the hepatic system, reconstructed using maximal disc guided shape based interpolation. Two unconnected but interlinked ducts are correctly reconstructed in the same process. The resulting surface, although complex, is much easier to interpret than the original object cross-sections.

5.2.3 Human Bladder

In order to validate *in-vivo* volume measurements, the actual volume must be known by an alternative, more accurate, method. This is possible for the bladder, which is a collapsible sack whose volume at any time is equal to the amount of fluid inside it. The output of the bladder can be easily measured from the amount of voiding. The input to the bladder is more difficult to measure, but can be estimated from sequential volume measurements in periods with no voiding. In addition, the bladder wall is very well defined by ultrasound, and is



(a) Organ surface

(b) Volume change over time

Figure 39: ***In-Vivo* Results:** Human bladder. The bladder was partially voided four times during the examination. Two sets of scans were recorded between each partial voiding. The data is contained in Table 3.

therefore easier to segment than, for instance, the kidney.

Ten scans were performed of an initially full bladder, in pairs, with partial voiding between each pair. The bladder was completely voided after the eighth scan. The scans were performed in fast sequence, the output being collected for later measurement, in order to limit the amount of bladder filling during the experiment. The output was then measured using a 20ml or 60ml graded syringe (dependent on the volume) to an accuracy of approximately 1ml. The stored ultrasound B-scans were then segmented, using 15 to 20 cross-sections per examination: Figure 39(a) shows an example of this. Volumes were calculated from these

Time, m:s	00:00	00:40	02:54	03:54	05:17	05:58	07:28	08:07	10:26	11:02
Volume, ml	342.7	363.4	360.9	369.2	306.0	319.8	194.8	206.2	14.7	20.3
Fill, ml/m	30.7		8.3		20.0		17.5		9.3	
Diff, ml		24.2		84.1		155.6		219.0		
Void, ml		25		74		156		234		
Error, %		1.6		6.8		0.1		3.2		

Table 3: ***In-Vivo* Results:** Human bladder. **Fill** is the estimated rate at which the bladder was filling. **Diff** is the calculated difference in volumes, adjusted for bladder filling, and **Void** is the actual measured output. The same data is shown graphically in Figure 39(b).

cross-sections using cubic planimetry, and are contained in Table 3.

The amount of bladder filling was estimated in three stages. Firstly, the linear rate of filling was calculated, for each pair of scans, from the volume measurements. Secondly, cubic splines were used to interpolate these values and give a continuous bladder filling rate. Thirdly, this function was integrated, to give the estimated amount by which the bladder had filled at any point during the experiment. This information, along with the measured volumes, was then used to estimate the actual bladder volume at any point (for this purpose the voiding was considered to be instantaneous at the mid-point between pairs of scans). The resulting curve is shown in Figure 39(b), along with the estimated amount of voiding from this curve.

The errors in Table 3 are calculated for the *actual volume measurements*, rather than the amount of voiding. The amount of voiding is a complicated function of the volume measurements, due to the adjustments for bladder fill rate, making these errors hard to estimate. Since the amount of voiding is essentially a measure of difference, the actual volume errors are assumed to add to give the voiding error. Hence, these errors are approximated to be half the voiding error.

6 Discussion

Accuracy of volume measurement Cubic planimetry gave the most accurate volume measurements with fewer segmented scans of the object in all cases, for both simulated and *in-vivo* experiments. In fact, typically only ten cross-sections were required to give an accuracy (due to the volume measurement technique alone) of better than ±1% (see Table 1). This was the case for both linear and complex scanning patterns, and simple or complex objects, even the sharp-edged cube in Figure 29.

In most cases 2-D cubic planimetry performed better than 3-D cubic planimetry, although the difference was generally slight. This difference is mostly due to the fitting of cubic splines to the end-points, which was not the same for the 2-D and 3-D methods.

For linear or nearly linear scanning patterns, linear planimetry gave more accurate volume measurements than any of the shape based methods, for example the cone and baseball glove in Figure 31. However, as the scanning patterns increased in complexity, this method gave less accurate volumes, and the shape based methods tended to be more accurate. See for example the sphere in Figure 29 and the longitudinal scan of the kidney in Figure 36. Linear planimetry assumes a linear change in area between scan planes, whereas shape based interpolation fits an approximately linear surface — which of them gives the better volume measurement is therefore a function of the relative accuracy of these assumptions.

There was little difference in volume measurement accuracy between the three shape based methods, although in general the centroid based technique performed worst. There were some notable exceptions where the maximal disc guided technique gave significantly better volumes — namely for the freehand scan of the jester's hat and the cube (see Table 1). These are both complex shapes which are more difficult to represent using the simpler shape based methods.

In-vivo volume measurement precision was achieved to ±6ml, ±5% for the kidney, and accuracy to ±15ml, ±7% for the bladder. The greater proportion of this error is probably attributable to the accuracy of segmentation, in particular for the first and last cross-sections, which were difficult to define for both organs.

Accuracy of surface reconstructions The "accuracy" of a reconstructed object surface is generally quite difficult to assess, particularly where only a few object cross-sections are involved. The same set of cross-sections can legally define an infinite number of surfaces, so it is not necessarily fair to compare a reconstructed surface with a known original — and in any case the original is often not known. Two methods for assessing the reconstructions can be used here. The simulated objects have a mathematically defined surface, and hence can easily be compared to their reconstructions. The *in-vivo* kidney does not, however the various scans shown in Figure 37 were all of the same kidney and hence should have the same shape (save for changes due to varying pressure on the organ).

A brief examination of the objects shown in Figures 29 to 32 shows that the reconstructed surfaces *are* very similar to the actual known surfaces. The cone surface in particular is nearly exact, even for the fan scanning pattern. The more complex baseball glove and jester's hat surfaces are also good representations of the originals shown in Figure 3, even though only a few cross-sections are used in the reconstruction.

The kidney reconstructions in Figure 37 are similar, although some differences can clearly be seen. The second transverse reconstruction was extremely difficult to segment, particularly in the upper region, which may explain the increase in size there. There is also a notable difference in curvature between the transverse and longitudinal reconstructions — this is the result of the clinician changing position for the alternative scanning pattern (he was sitting down and scanning himself for all these scans).

The display of part of the hepatic system in Figure 38 demonstrates that the surface reconstruction can correctly handle more complex tree-like shapes also.

Robustness Two practical situations which can effect the accuracy of volume measurement with freehand 3-D ultrasound were investigated. The effect of the clinician holding the probe with an unsteady hand is very small for all of the methods examined — volume errors are still only about ±1% for unsteadiness of up to 5% of the object size. This slight reduction in accuracy, seen in Figure 33, is more a result of the larger gaps between object cross-sections than the irregularity of the scan positions.

As expected, mis-registration of the scan planes (i.e. errors in the position and orientation information) is a more significant cause of volume measurement error. In this case volume errors are about ±10% for mis-registration errors of up to 5% of the object size. However, the ultrasound system used was accurate to typically ±1mm, or 2% for the kidney reconstructions, giving expected volume errors due to mis-registration of only ±3%.

Even in cases where the data is highly mis-registered, none of the algorithms presented here fail to produce a volume estimation or a surface which is at least as good as the original data.

Processing speed Both cubic planimetry and maximal disc guided shape based interpolation are fast enough to be performed immediately after an examination. For the examples contained here, the processing typically lasted less than 1s or about 15s, respectively, on readily available computer hardware. Both these times are much less than that required to manually segment the object scans, which is estimated to take approximately 30s per scan. Hence the main gain in processing time is actually the reduction in the number of cross-sections required — the entire time from scanning to surface display being about 5 minutes for a typical examination.

7 Conclusions

Cubic planimetry is a highly accurate volume measurement technique, which can be used directly on freehand 3-D ultrasound data. It is more accurate than the already well acclaimed step-section planimetry. In addition, it still produces very good volume measurements with as few as ten or so scan planes, which can significantly reduce the amount of time and effort required in the image segmentation. The calculation requires minimal additional memory or processing time. It appears to give accurate results in all practical situations, as well as being robust in some very extreme ones.

Maximal disc guided shape based interpolation is a robust surface reconstruction technique which is particularly suited to complex objects scanned with only a few arbitrarily orientated planes. In simple cases it performs as well as conventional shape based interpolation, but it can also handle more complex objects and scanning patterns, with exactly the same processing in each case. The technique also gives good definition of surface normals, which improves the display of lower resolution surfaces.

Since both techniques require only a handful of segmented object cross-sections, they are well suited for use on the same data. For instance, cubic planimetry could be used first to give a precise volume measurement. Maximal disc guided shape based interpolation could then be used to provide a reliable object surface. This would give the clinician confidence in the segmentation, and hence also in the cubic planimetry volume measurement.

8 Acknowledgements

The authors would like to acknowledge Robert Rohling and Jonathan Carr, also at the Cambridge University Department of Engineering, for many helpful discussions. Graham Treece would also like to thank his wife, Sarah Treece, for help with the clinical applications of this work, and for her encouragement to start it in the first place. Graham Treece is supported by an EPSRC studentship, and a Newton Trust award from the Cambridge University Department of Engineering.

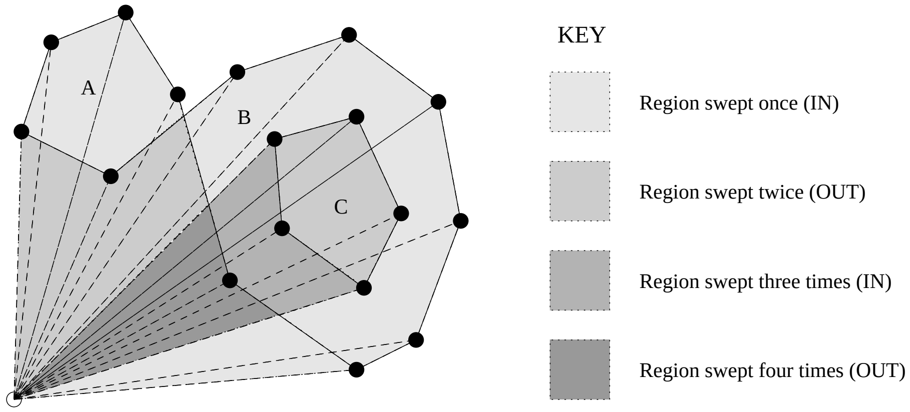


Figure 40: **Manual contour tracing.** Example of the determination of "inside" and "outside" object areas.

A Manual object segmentation

Manual tracing of the object cross-sections in a scan plane results in a list of connected points defining the object border. It is then necessary to decide from this border which areas are inside the object and which are outside. This is similar to the "point in polygon" problem described for instance by O'Rourke [61], except that here we require *all* the points which are inside the polygon.

The method used here is similar in concept to that of *winding numbers* [61]. Figure 40 shows an example of a contour defined by a list of points, sampled with a very coarse resolution. An image resolution is then chosen and a discrete image at this resolution, and large enough to contain the contour, is initialised to zero. Next, an arbitrary point (in our case one of the corners of the image) is chosen, and this is gradually triangulated to the sequential points making up the cross-section, as in the figure. For each of these triangles, any points on the image contained by them are exclusive-ored with 1, which results in toggling of the binary pixel values.

The result of this operation is a binary image where all pixels contained by the object are set to 1. The "containment" in this sense means that the pixel is enclosed by an odd number of closed boundaries. For instance, both the regions A and B on the figure are enclosed by one boundary, however region C is enclosed by two, hence this is regarded as a "hole" in the object.

This method works reliably irrespective of the direction that the boundary is drawn in (clockwise or anti-clockwise), the complexity of the shape, the number of self intersections, or the number of sub-objects (i.e. objects enclosed within holes within other objects).

B Area from parametric cubic splines

Given two curves, defined parametrically:

$$\begin{bmatrix} x_i(t) & y_i(t) \end{bmatrix} = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} x_{i3} & y_{i3} \\ x_{i2} & y_{i2} \\ x_{i1} & y_{i1} \\ x_{i0} & y_{i0} \end{bmatrix} \quad \text{where} \quad 0 \leq t \leq 1$$

If each curve is connected to the other by two straight lines joining the start points, $t = 0$, and the end points, $t = 1$, the enclosed area can be calculated from:

$$A = \left| \int_{t=0}^{1} \vec{s} \cdot d\vec{\omega} \right| \tag{16}$$

where $\vec{s}$ is a vector normal to the line joining the curves at the same value of t, and $\vec{\omega}$ is the position of the centre of that line, each given by:

$$\begin{aligned} \vec{s}(t) &= \begin{bmatrix} y_1(t) - y_2(t) & x_2(t) - x_1(t) \end{bmatrix} \\ &\equiv \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} sx_3 & sy_3 \\ sx_2 & sy_2 \\ sx_1 & sy_1 \\ sx_0 & sy_0 \end{bmatrix} \end{aligned} \tag{17}$$

$$\begin{aligned} d\vec{\omega}(t) &= \frac{1}{2} \begin{bmatrix} dx_1(t) + dx_2(t) & dy_1(t) + dy_2(t) \end{bmatrix} \\ &= \begin{bmatrix} t^2 & t & 1 \end{bmatrix} \frac{1}{2} \begin{bmatrix} 3\,(x_{13} + x_{23}) & 3\,(y_{13} + y_{23}) \\ 2\,(x_{12} + x_{22}) & 2\,(y_{12} + y_{22}) \\ x_{11} + x_{21} & y_{11} + y_{21} \end{bmatrix} dt \\ &\equiv \begin{bmatrix} t^2 & t & 1 \end{bmatrix} \begin{bmatrix} \omega x_2 & \omega y_2 \\ \omega x_1 & \omega y_1 \\ \omega x_0 & \omega y_0 \end{bmatrix} dt \end{aligned} \tag{18}$$

So the area can be calculated as:

$$\begin{aligned} A &= \left| \int_{t=0}^{1} c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0 dt \right| \\ &= \left| \frac{c_5}{6} + \frac{c_4}{5} + \frac{c_3}{4} + \frac{c_2}{3} + \frac{c_1}{2} + c_0 \right| \end{aligned} \tag{19}$$

where the coefficients $c_5, \ldots, c_0$ can be derived from the coefficients of equations (18) and (17), as follows:

$$\begin{aligned} c_5 &= sx_3\omega x_2 + sy_3\omega y_2 \\ c_4 &= sx_3\omega x_1 + sx_2\omega x_2 + sy_3\omega y_1 + sy_2\omega y_2 \\ c_3 &= sx_3\omega x_0 + sx_2\omega x_1 + sx_1\omega x_2 + sy_3\omega y_0 + sy_2\omega y_1 + sy_1\omega y_2 \\ c_2 &= sx_2\omega x_0 + sx_1\omega x_1 + sx_0\omega x_2 + sy_2\omega y_0 + sy_1\omega y_1 + sy_0\omega y_2 \\ c_1 &= sx_1\omega x_0 + sx_0\omega x_1 + sy_1\omega y_0 + sy_0\omega y_1 \\ c_0 &= sx_0\omega x_0 + sy_0\omega y_0 \end{aligned} \tag{20}$$

References

[1] R. G. Aarnink, J. J. M. C. H. de la Rosette, F. M. J. Debruyne, and H. Wijkstra. Reproducibility of prostate volume measurements from transrectal ultrasonography by an automated and a manual technique. *British Journal of Urology*, 78:219–223, 1996.

[2] R. G. Aarnink, A. L. Huynen, R. J. B. Giesen, J. J. M. C. H. de la Rosette, F. M. J. Debruyne, and H. Wijkstra. Automated prostate volume determination with ultrasonographic imaging. *The Journal of Urology*, 153:1549–1554, May 1995.

[3] K. Altmann, Z. Shen, L. M. Boxt, D. L. King, W. M. Gersony, L. D. Allan, and H. D. Apfel. Comparison of three-dimensional echocardiographic assessment of volume, mass, and function in children with functionally single left ventricles with two-dimensional echocardiography and magnetic resonance imaging. *The American Journal of Cardiology*, 80:1060–1065, October 1997.

[4] C. Arcelli and G. Sanniti di Baja. Finding local maxima in a pseudo-Euclidean distance transform. *Computer Vision, Graphics and Image Processing*, 43:361–367, 1988.

[5] F. G. Balen, C. M. Allen, J. E. Gardener, N. C. Siddle, and W. R. Lees. 3-dimensional reconstruction of ultrasound images of the uterine cavity. *The British Journal of Radiology*, 66:588–591, 1993.

[6] C. D. Barry, C. P. Allott, N. W. John, P. M. Mellor, P. A. Arundel, D. S. Thomson, and J. C. Waterton. Three-dimensional freehand ultrasound: Image reconstruction and volume analysis. *Ultrasound in Medicine and Biology*, 23(8):1209–1224, 1997.

[7] O. Basset, G. Gimenez, J. L. Mestas, D. Cathignol, and M. Devonec. Volume measurement by ultrasonic transverse or sagittal cross-sectional scanning. *Ultrasound in Medicine and Biology*, 17(3):291–296, 1991.

[8] M. Belohlavek, D. A. Foley, J. B. Seward, and J. F. Greenleaf. 3D echocardiography: Reconstruction algorithm and diagnostic performance of resulting images. *Proceedings of SPIE*, 2359:680–692, 1994.

[9] H. Blum. Biological shape and visual science (Part I). *Journal of Theoretical Biology*, 38:205–287, 1973.

[10] F. Bonilla-Musoles, F. Raga, N. G. Osborne, and J. Blanes. Control of intrauterine device insertion with three-dimensional ultrasound: Is it the future? *Journal of Clinical Ultrasound*, 24:263–267, June 1996.

[11] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27:321–345, 1984.

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

[13] G. Borgefors and I. Nyström. Efficient shape representation by minimizing the set of centres of maximal discs/spheres. *Pattern Recognition Letters*, 18:465–472, 1997.

[14] E. Catmull and R. Rom. A class of local interpolating splines. In R. Barnhill and R. Risenfeld, editors, *Computer Aided Geometric Design*, pages 317–326. Academic Press, San Francisco, 1974.

[15] H.-H. Chang and H. Yan. Skeletonization of binary digital patterns using a fast Euclidean distance transformation. *Optical Engineering*, 35(4):1003–1008, April 1996.

[16] 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.

[17] L. T. Cook, P. N. Cook, K. R. Lee, S. Batnitzky, B. Y. S. Wong, S. L. Fritz, J. Ophir, S. J. Dwyer III, L. R. Bigongiari, and A. W. Templeton. An algorithm for volume estimation based on polyhedral approximation. *IEEE Transactions on Biomedical Engineering*, 27(9):493–500, September 1980.

[18] G. Coppini, R. Poli, and G. Valli. Recovery of the 3-D shape of the left ventricle from echocardiographic images. *IEEE Transactions on Medical Imaging*, 14(2):301–317, June 1995.

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

[20] J. Deng, J. E. Gardener, C. H. Rodeck, and W. R. Lees. Fetal echocardiography in three and four dimensions. *Ultrasound in Medicine and Biology*, 22(8):979–986, 1996.

[21] D. B. Downey, D. A. Nicolle, M. F. Levin, and A. Fenster. Three-dimensional ultrasound imaging of the eye. *Eye*, 10:75–81, 1996.

[22] H. Embrechts and D. Roose. MIMD divide-and-conquer algorithms for the distance transformation. Part II: chamfer 3-4 distance. *Parallel Computing*, 21:1077–1096, 1995.

[23] A. Fenster and D. B. Downey. 3-D ultrasound imaging: A review. *IEEE Engineering in Medicine and Biology*, 15(6):41–51, November 1996.

[24] A. Fenster, S. Tong, S. Sherebrin, D. B. Downey, and R. N. Rankin. Three-dimensional ultrasound imaging. *Proceedings of SPIE*, 2432:176–184, 1995.

[25] D. Fine, S. Perring, J. Herbetko, C. N. Hacking, J. S. Fleming, and K. C. Dewbury. Three-dimensional (3D) ultrasound imaging of the gallbladder and dilated biliary tree: reconstruction from real-time B-scans. *The British Journal of Radiology*, 64:1056–1057, 1991.

[26] J. D. Foley et al. *Computer Graphics: Principles and Practice*. Addison-Wesley Systems Programming Series. Addison-Wesley, second edition, 1996.

[27] D. S. Fritsch, S. M. Pizer, B. S. Morse, D. H. Eberly, and A. Liu. The multiscale medial axis and its applications in image registration. *Pattern Recognition Letters*, 15:445–452, 1994.

[28] O. H. Gilja, A. I. Smievoll, N. Thune, K. Matre, T. Hausken, S. Ødegaard, and A. Berstad. *In vivo* comparison of 3D ultrasonography and magnetic resonance imaging in volume estimation of human kidneys. *Ultrasound in Medicine and Biology*, 21(1):25–32, 1995.

[29] A. S. Gopal, M. J. Schnellbaecher, Z. Shen, O. O. Akinboboye, P. M. Sapin, and D. L. King. Freehand three-dimensional echocardiography for measurement of left ventricular mass: *In Vivo* anatomic validation using explanted human hearts. *Journal of the American College of Cardiology*, 30(3):802–810, September 1997.

[30] G. J. Grevera and J. K. Udupa. Shape-based interpolation of multidimensional grey-level images. *IEEE Transactions on Medical Imaging*, 15(6):881–892, 1996.

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

[32] 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.

[33] T. C. Hodges, P. R. Detmer, D. H. Burns, K. W. Beach, and D. E. Strandness Jr. Ultrasonic three-dimensional reconstruction: *in vitro* and *in vivo* volume and area measurement. *Ultrasound in Medicine and Biology*, 20(8):719–729, 1994.

[34] D. Howe, T. Wheeler, and S. Perring. Measurement of placental volume with real-time ultrasound in mid-pregnancy. *Journal Clinical Ultrasound*, 22:77–83, 1994.

[35] S. W. Hughes, T. J. D'Arcy, D. J. Maxwell, W. Chiu, A. Milner, J. E. Saunders, and R. J. Sheppard. Volume estimation from multiplanar 2D ultrasound images using a remote electromagnetic position and orientation sensor. *Ultrasound in Medicine and Biology*, 22(5):561–572, 1996.

[36] S. W. Hughes, T. J. D'Arcy, D. J. Maxwell, J. E. Saunders, C. F. Ruff, W. S. C. Chiu, and R. J. Sheppard. Application of a new discreet form of Gauss' theorum for measuring volume. *Physics in Medicine and Biology*, 41:1809–1821, 1996.

[37] P. K. Jensen and M. K. Hansen. Ultrasonographic, three-dimensional scanning for determination of intraocular tumor volume. *Acta Ophthalmologica*, 69:178–186, 1991.

[38] B. Jin, L. Turner, B. Crawford, A. Birrel, and D. J. Handelsman. The development of the baboon prostate: Ultrasound methodology, modelling, and natural history. *Journal of Andrology*, 17(4):342–352, 1996.

[39] M. W. Jones and M. Chen. A new approach to the construction of surfaces from contour data. *Computer Graphics Forum*, 13(3):C–75–C–84, 1994.

[40] T. Kimoto and Y. Yasuda. Shape description and representation by ellipsoids. *Signal Processing: Image Communication*, 9:275–290, 1997.

[41] D. L. King, A. S. Gopal, A. M. Keller, P. M. Sapin, and K. M. Schröder. Three-dimensional echocardiography: Advances for measurement of ventricular volume and mass. *Hypertension*, 23(1):I–172–I–179, January 1994.

[42] H.-M. Klein, R. W. Günther, M. Verlande, W. Schneider, D. Vorwerk, J. Kelch, and M. Hamm. 3D-surface reconstruction of intravascular ultrasound images using personal computer hardware and a motorized catheter control. *Cardiovascular and Interventional Radiology*, 15:97–101, 1992.

[43] R. C. Lalouche, D. Bickmore, F. Tessler, N. J. Mankovich, H. K. Huang, and H. Kangarloo. Three-dimensional reconstruction of ultrasound images. *Proceedings of SPIE*, 1092, 1989. 450–457.

[44] A. Lee, J. Deutinger, and G. Bernaschek. Three dimensional ultrasound: abnormalities of the fetal face in surface and volume rendering mode. *British Journal of Obstetrics and Gynaecology*, 102:302–306, April 1995.

[45] 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.

[46] G. Magni, Q.-L. Cao, L. Sugeng, A. Delabays, G. Marx, A. Ludomirski, M. Vogel, and N. G. Pandian. Volume-rendered, three-dimensional echocardiographic determination of the size, shape, and position of atrial septal defects: Validation in an *in vitro* model. *American Heart Journal*, 132(2):376–381, 1996.

[47] L. S. Marks, F. J. Dorey, M. L. Macairan, C. Park, and J. B. de Kernion. Three-dimensional ultrasound device for rapid determination of bladder volume. *Urology*, 50(3):341–348, 1997.

[48] R. W. Martin, G. Bashein, P. R. Detmer, and W. E. Moritz. Ventricular volume measurement from a multiplanar transesophageal ultrasonic imaging system: An *In Vitro* study. *IEEE Transactions on Biomedical Engineering*, 37(5):442–449, 1990.

[49] E. Merz, F. Bahlmann, G. Weber, and D. Macchiella. Three-dimensional ultrasonography in prenatal diagnosis. *Journal of Perinatal Medicine*, 23:213–222, 1995.

[50] A. Montanvert and Y. Usson. Discrete distances applied to 2D granulometry and 3D reconstruction. In *8th Scandinavian Conference on Image Analysis*, pages 1153–1160, 1993.

[51] W. E. Moritz, A. S. Pearlman, D. H. McCabe, D. K. Medema, M. E. Ainsworth, and M. S. Boles. An ultrasonic technique for imaging the ventricle in three dimensions and calculating its volume. *IEEE Transactions on Biomedical Engineering*, 30(8):482–492, August 1983.

[52] M. Moshfeghi. Directional interpolation for magnetic resonance angiography data. *IEEE Transactions on Medical Imaging*, 12(2):366–379, June 1993.

[53] M. S. Nathan, K. Seenivasagam, Q. Mei, J. E. A. Wickham, and R. A. Miller. Transrectal ultrasonography: why are estimates of prostate volume and dimension so inaccurate? *British Journal of Urology*, 77:401–407, 1996.

[54] T. R. Nelson and T. T. Elvins. Visualization of 3D ultrasound data. *IEEE Computer Graphics and Applications*, pages 50–57, November 1993.

[55] T. R. Nelson and D. H. Pretorius. Visualisation of the fetal thoracic skeleton with three-dimensional sonography. *American Journal of Roentgenology*, 164(6):50–57, 1995.

[56] K. J. Ng, J. E. Gardener, D. Rickards, W. R. Lees, and E. J. G. Milroy. Three-dimensional imaging of the prostatic urethra — an exciting new tool. *British Journal of Urology*, 74:604–608, 1994.

[57] F. Nilsson and P.-E. Danielsson. Finding the minimal set of maximum disks for binary objects. *Graphical Models and Image Processing*, 59(1):55–60, 1997.

[58] E. O. Ofili and N. C. Nanda. Three-dimensional and four-dimensional echocardiography. *Ultrasound in Medicine and Biology*, 20(8):669–675, 1994.

[59] Y. Ohashi. Three-dimensional reconstruction of pore geometry from serial sections — image algebraic approach. In R. Pflug and J. W. Harbaugh, editors, *Computer Graphics in Geology*, volume 41 of *Lecture Notes in Geology*, pages 63–76. Springer, 1992.

[60] R. Ohbuchi, D. Chen, and H. Fuchs. Incremental volume reconstruction and rendering for 3D ultrasound imaging. *Proceedings of SPIE*, 1808:312–323, 1992.

[61] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1993.

[62] R. W. Prager, A. H. Gee, and L. Berman. Stradx: real-time acquisition and visualisation of free-hand 3D ultrasound. Technical Report CUED/F-INFENG/TR 319, Cambridge University Department of Engineering, April 1998.

[63] R. W. Prager, R. N. Rohling, A. H. Gee, and L. Berman. Automatic calibration for 3-D free-hand ultrasound. Technical Report CUED/F-INFENG/TR 303, Cambridge University Department of Engineering, September 1997.

[64] R. W. Prager, R. N. Rohling, A. H. Gee, and L. Berman. Rapid calibration for 3-D free-hand ultrasound. *Ultrasound in Medicine and Biology*, 24(6):855–869, 1998.

[65] D. T. Puff, D. Eberly, and S. M. Pizer. Object-based interpolation via cores. *Proceedings of SPIE*, 2167:143–150, 1994.

[66] A. Rahmouni, A. Yang, C. M. C. Tempany, T. Frenkel, J. Epstein, P. Walsh, P. K. Leichner, C. Ricci, and E. Zerhouni. Accuracy of *In-Vivo* assessment of prostatic volume by MRI and transrectal ultrasonography. *Journal of Computer Assisted Tomography*, 16(6):935–940, 1992.

[67] R. N. Rankin, A. Fenster, D. B. Downey, P. L. Munk, M. F. Levin, and A. D. Vellet. Three-dimensional sonographic reconstruction: Techniques and diagnostic applications. *American Journal of Roentgenology*, 161:695–702, 1993.

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

[69] R. N. Rohling, A. H. Gee, and L. Berman. Three-dimensional spatial compounding of ultrasound images. *Medical Image Analysis*, 1(3):177–193, April 1997.

[70] K. Rosenfield, P. Boffetti, J. Kaufman, R. Weinstein, S. Razvi, and J. M. Isner. Three-dimensional reconstruction of human carotid arteries from images obtained during non-invasive B-mode ultrasound examination. *The American Journal of Cardiology*, 70:379–384, August 1992.

[71] A. Salustri and J. R. T. C. Roelandt. Ultrasonic three-dimensional reconstruction of the heart. *Ultrasound in Medicine and Biology*, 21(3):281–293, 1995.

[72] C. M. Sehgal, G. A. Broderick, R. Whittington, R. J. T. Gorniak, and P. H. Arger. Three-dimensional US and volumetric assessment of the prostate. *Radiology*, 192(1):274–278, July 1994.

[73] 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.

[74] H. Steiner, A. Staudach, D. Spitzer, and H. Schaffer. Three-dimensional ultrasound in obstetrics and gynaecology: technique, possibilities and limitations. *Human Reproduction*, 9(9):1773–1778, 1994.

[75] M. K. Terris and T. A. Stamey. Determination of prostate volume by transrectal ultrasound. *The Journal of Urology*, 145:984–987, May 1991.

[76] J. W. Trobaugh, D. J. Trobaugh, and W. D. Richard. Three-dimensional imaging with stereotactic ultrasonography. *Computerized Medical Imaging and Graphics*, 18(5):315–323, 1994.

[77] Y. Watanabe. A method for volume estimation by using vector areas and centroids of serial cross sections. *IEEE Transactions on Biomedical Engineering*, 29(3):202–205, 1982.

[78] P. N. Werahera, G. J. Miller, G. D. Taylor, T. Brubaker, F. Daneshgari, and E. D. Crawford. A 3-D reconstruction algorithm for interpolation and extrapolation of planar cross sectional data. *IEEE Transactions on Medical Imaging*, 14(4):765–771, 1995.