

# Real-Time Tools for Freehand 3D Ultrasound

Richard Prager<sup>1</sup>, Andrew Gee<sup>1</sup>, and Laurence Berman<sup>2</sup>

<sup>1</sup> Department of Engineering, University of Cambridge, UK

<sup>2</sup> Department of Radiology, University of Cambridge, UK

**Abstract.** One of the drawbacks of many 3D ultrasound systems is the two-stage nature of the process. The strict separation of data acquisition and visualisation disturbs the continuity of the ultrasound examination. Furthermore, some systems require the clinician to wait for an unacceptable amount of time while the voxel array is constructed. In this paper, we describe tools which allow arbitrary slices to be viewed through the 3D data set *in real-time as the examination is taking place*. This leads to a natural integration of the scanning and visualisation processes.

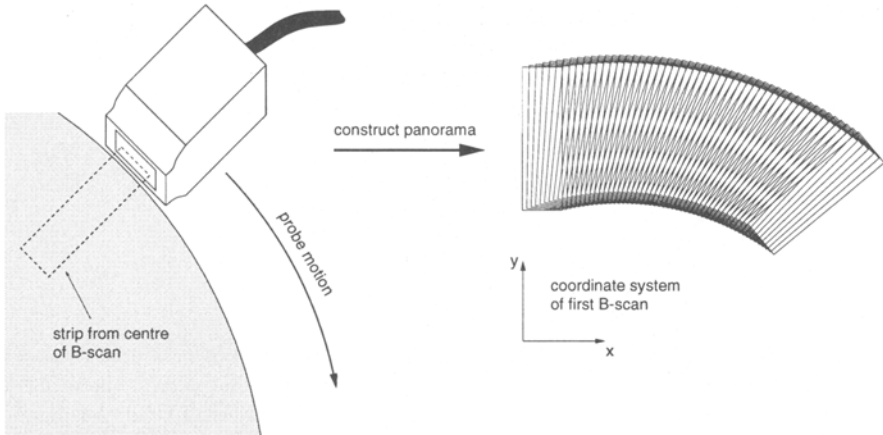
## 1 Introduction

Conventional diagnostic ultrasound imaging is performed with a hand-held probe which transmits ultrasound pulses into the body and receives the echoes. The magnitude and timing of the echoes are used to create a 2D grey scale image (B-scan) of a cross-section of the body in the scan plane.

Using a technique called **3D freehand ultrasound imaging**, it is possible to construct 3D data sets from a series of 2D B-scans. A position sensor is attached to the probe, so that each B-scan is labelled with the position and orientation of the scan plane. It is then possible to slot each B-scan into a voxel array, producing a regularly sampled 3D ultrasound data set. The data can be visualised by any-plane slicing or volume rendering. Organs can be segmented, enabling estimation of their volumes and graphical rendering of their surfaces.

What emerges is a two stage process: the clinician first performs the data acquisition and then, when this is complete, uses a separate set of controls for reslicing and visualisation. This disturbs the interactive nature of the ultrasound examination. In this paper, we describe two tools for real-time display of data from a freehand 3D ultrasound system. These tools enable the clinician to visualise any slice taken at any angle while the scan is actually taking place.

The first tool enables the visualisation of a long structure in a single plane. It involves extracting vertical strips from a series of ultrasound images as the probe is moved over the structure, and aligning them to produce a composite, panoramic picture of the complete length of the structure. Recent interest in this type of imaging has been triggered by the release of the Siemens Sonoline Elegra ultrasound machine, which offers a panoramic imaging facility called “SieScape”. The Siemens machine achieves accurate registration by correlating consecutive images in real-time. This requires special-purpose, dedicated hardware that contributes to the high cost of the machine. We achieve a similar result using the 3D position and orientation of the probe provided by the position sensing device.



**Fig. 1. Panoramic ultrasound.** As the probe is moved in the plane of the B-scans, thin strips from the centre of the B-scans are used to construct a panorama in real-time.

The second tool uses the same live information about the current position of the probe to build up a picture showing a slice through the body in any required direction. The clinician defines the position of the slice and then, as the probe is moved along, the slice image is gradually built up on the computer screen.

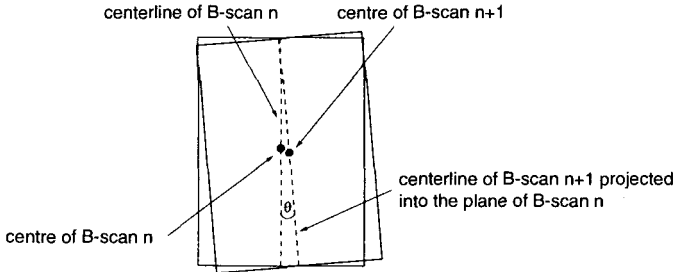
## 2 Panoramic ultrasound

Data for a panoramic image is acquired by translating the probe in the plane of the B-scans. The B-scans are then accurately registered and stitched together to produce a seamless composite — see Figure 1. This is performed in real-time, so the clinician sees the panorama as the probe is moved over the region of interest.

### 2.1 Registration

We use the readings from the position sensor to register sequential B-scans and hence construct a panoramic image. Consider B-scans  $n$  and  $n+1$ . Since we know the 3D position and orientation of both scans, it is straightforward to calculate the rigid body transformation  $\mathbf{T}$  between the local coordinate systems of the two images. We can then map pixels in B-scan  $n+1$  into B-scan  $n$ 's coordinate system, and display both images on the same axes.

There is one complication. Unless the probe is moved perfectly in the plane of the B-scans,  $\mathbf{T}$  will involve not only a within-plane translation and rotation, but also an out-of-plane transformation. We need to “flatten” this transformation to warp the surface traced out by the B-scans onto a flat plane. This can be achieved as follows. We use  $\mathbf{T}$  to transform the centerline of B-scan  $n+1$  into B-scan  $n$ 's coordinate system. The centre and endpoints of the centerline will not



**Fig. 2. Registering consecutive B-scans.** The position readings are used to map the centerline of B-scan  $n+1$  into B-scan  $n$ 's coordinate system. In general, the centerline lies outside the plane of B-scan  $n$ , so its endpoints are projected onto the plane by zeroing their  $z$ -coordinates. It is then simple to measure the angle  $\theta$  and deduce  $(\delta x, \delta y)$ , such that an in-plane rotation  $\theta$  followed by a translation  $(\delta x, \delta y)$  maps the centre of B-scan  $n+1$  on to its projected position in B-scan  $n$ 's coordinate system.

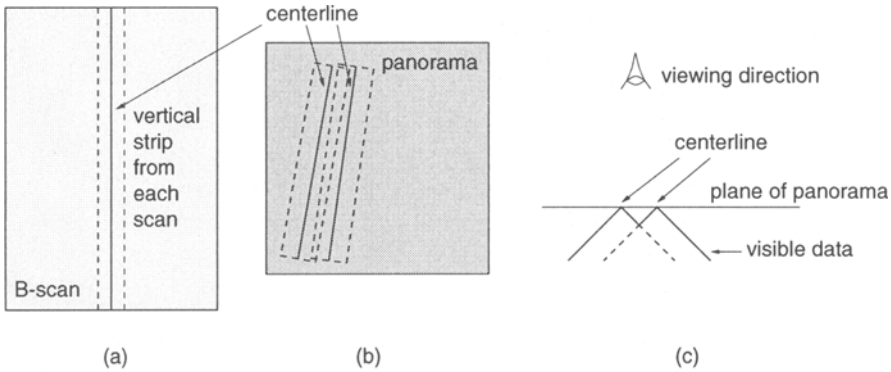
generally have zero  $z$ -coordinates in the new coordinate system. We flatten the transformation by setting these coordinates to zero. We then calculate the angle  $\theta$  and the displacement  $(\delta x, \delta y)$ , such that a within-plane rotation  $\theta$  followed by a translation  $(\delta x, \delta y)$  maps the centre of B-scan  $n+1$  on to its projected position in B-scan  $n$ 's coordinate system — see Figure 2. We can easily construct a planar Euclidean transformation matrix  $\mathbf{T}'$  from  $\theta$  and  $(\delta x, \delta y)$ . The panorama is then created using  $\mathbf{T}'$  to map pixels in B-scan  $n+1$  into B-scan  $n$ 's coordinate system.

We do not need to render every B-scan in its entirety to construct the panoramic image. Consecutive B-scans will overlap, and it suffices to use only a narrow strip from the centre of each B-scan. The panorama is constructed in the coordinate system of the *first* B-scan.  $\mathbf{T}'$  is calculated for each pair of consecutive B-scans, and the frame-to-frame transformations are concatenated to refer the current B-scan back to the coordinate system of the first B-scan.

## 2.2 Wedge-based interpolation

Where adjacent strips overlap, we would like to see pixels from the centre of the strips and not the edges. This can be achieved by mapping the intensity data not onto a flat strip, but a wedge, such that the projection of the wedge onto the plane is the same as the original strip — see Figure 3. To an observer viewing the plane from above, a single wedge looks exactly the same as the flat strip. However, intensities associated with the centre of each strip are close to the viewer, while those closer to each edge are further away. If the graphics system renders the wedges with hidden surface removal, then the panorama will reveal only those portions of the strips that are as close as possible to their centerlines.

This technique is free of parameters. It is necessary to specify the width of each strip, but this can be determined automatically, using  $\mathbf{T}'$  to calculate the minimum width such that there are no gaps between adjacent strips. Since the



**Fig. 3. Correct treatment of overlapping strips.** Where strips overlap, the panorama should show data from the centre of the B-scan, not the edges. This can be accomplished by “folding” each strip around its centerline to form a wedge. If the graphics system renders the wedges with hidden surface removal, then the slice will reveal only those portions of the strips that are as close as possible to their centerlines.

wedges determine the parts of each strip that are displayed, using slightly too wide a strip only affects the rendering speed, not the image quality.

### 2.3 Implementation

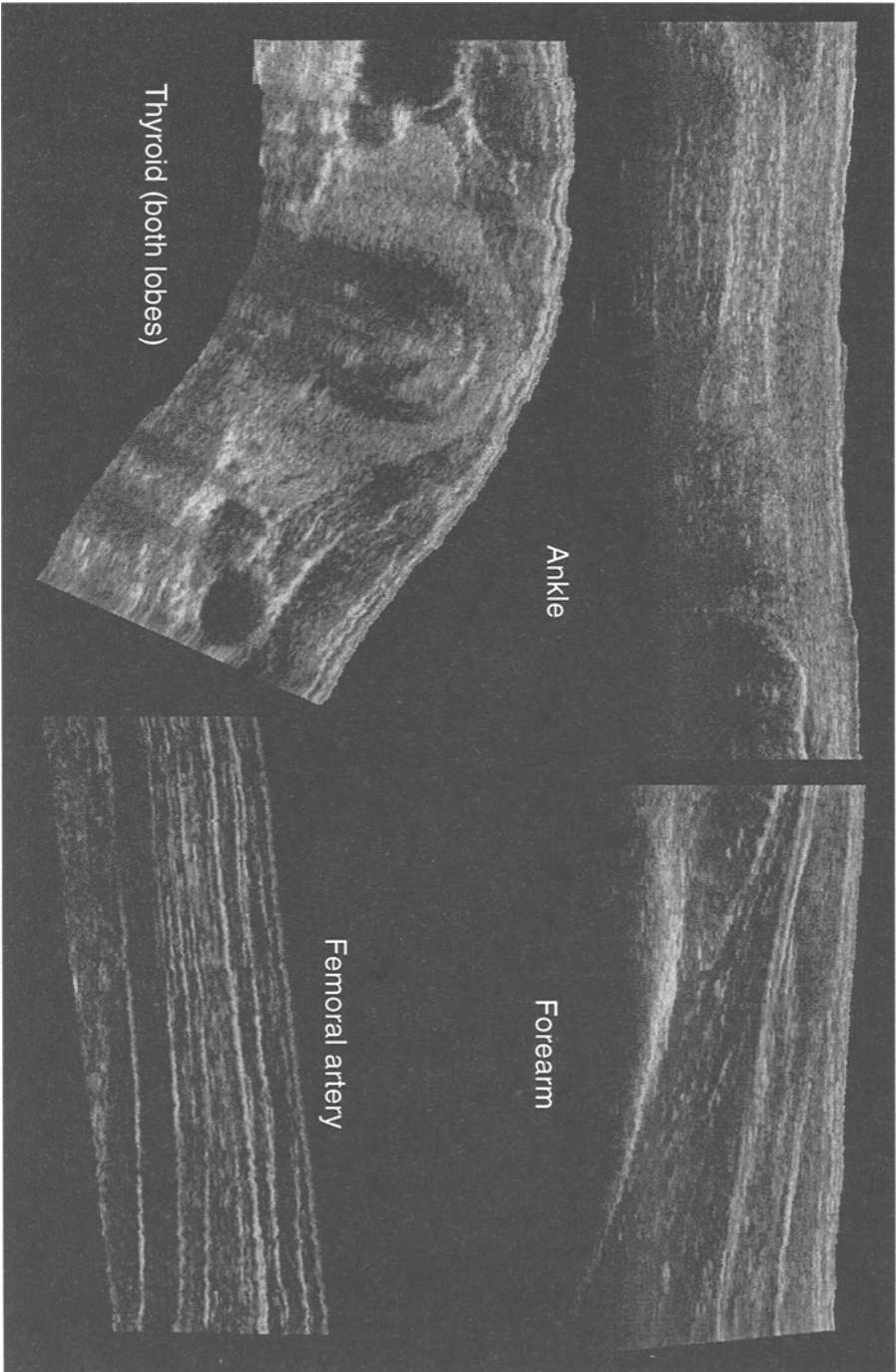
Central to the success of this approach is the need to develop a *fast* implementation. To this end, we exploit the standard graphics accelerator hardware found in many of today’s desktop computers.

The wedges can be rendered using one of two techniques. The first, 2D texture mapping, defines each wedge as two rectangles and projects the appropriate grey level pattern from the B-scan onto the rectangles. This is the preferred technique with hardware that offers flexible, accelerated texture mapping. Alternatively, we can extract adjacent columns of pixels from the B-scans, and place each pixel at its correct position on the wedge. These points can then be rendered as the vertices of a dense mesh of triangles, using Gouraud shading to fill in the gaps.

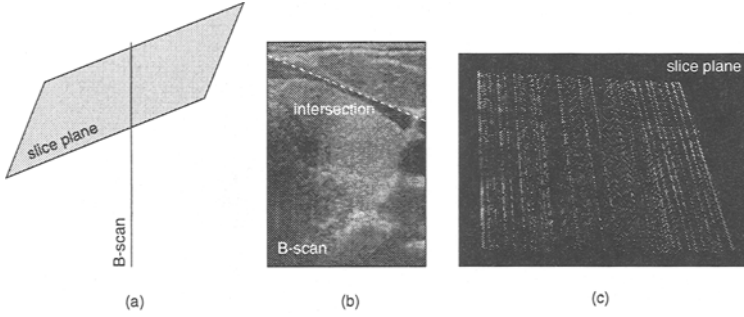
Given suitable (but not extravagant) graphics hardware, the panorama can be rendered very rapidly. For instance, using the Gouraud technique, the panorama is easily rendered on a Silicon Graphics Indy workstation at the video frame rate. Typical panoramas constructed using this technique can be found in Figure 4.

## 3 Real-time any-plane slicing

The normal way to produce an arbitrary slice through a 3D ultrasound data set is to resample the data onto a regular grid [1] and then use fast, integer arithmetic to extract the voxels straddling the desired slice. If we are to compute the slice



**Fig. 4. Typical real-time panoramic images.**



**Fig. 5. Reslicing thin B-scans.** The slice plane intersects each B-scan along a line, as shown in (a) and (b). Intensities can be extracted from the B-scans along these lines and drawn on the slice plane. However, this produces a fragmented image (c).

as the data is acquired, we cannot perform the initial voxelisation procedure. The system must work only with the raw B-scans and their positions.

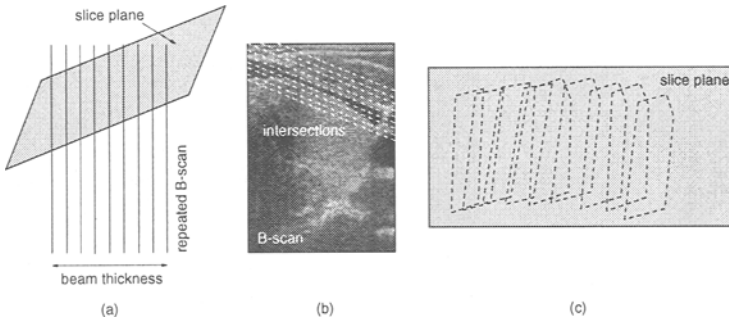
We also need to design a suitable tool which allows the clinician to define the desired slice plane without reference to the full 3D data volume (which has not yet been acquired). In our current implementation, the clinician places the probe orthogonal to the required slice and then uses a touch screen to draw a line at a suitable position on the current B-scan: this defines the line of intersection of the slice with the current B-scan. The clinician can then sweep the probe over the volume of interest, while the slice appears in real-time on the computer monitor.

A naive reslicing algorithm might extract intensity values along the lines of intersection of each B-scan with the slice plane, and paint these intensities onto the slice plane — see Figure 5. The problem with this approach is that the slice comprises a set of fragmented line segments, as shown in Figure 5(c).

To improve the quality of the slice we clearly need to interpolate between the line segments. Figure 6 illustrates an interpolation scheme which takes into account the anisotropic resolution of the 3D ultrasound data. Since the ultrasound beam has a finite thickness, perhaps as much as 10 mm [2], we should think not of a single, thin B-scan in space, but a continuum of repeated B-scans smeared across the width of the ultrasound beam<sup>1</sup>. The intersection of the slice plane with the smeared B-scan is now a polygon. Intensity values can be extracted within this polygon and drawn at the appropriate position on the slice plane. Repeating this process for all B-scans tiles the slice plane with a set of overlapping polygons filled with intensity data.

It is only left to decide which intensity to display where two or more polygons overlap on the slice plane. Intuitively, we would like to see data gathered along the centerline of the ultrasound beam, and not the periphery. We therefore use the wedge-based strategy described in Section 2.2 above.

<sup>1</sup> For simplicity, we assume that the width of the ultrasound beam is constant, though in reality the width varies with distance from the probe face [2].



**Fig. 6. Reslicing thick B-scans.** In practice, the ultrasound beam is not focussed out of the plane of the B-scan. In effect, the B-scan can be imagined repeated in space over the extent of the beam thickness (a). The intersection of each B-scan with the slice plane is now a polygon (b). Intensities can be extracted from within the polygons and drawn on the slice plane. This typically produces a set of filled, overlapping polygons (c).

Orthogonal slices, rendered using the Gouraud scheme, are shown in Figures 7(d)–(f)<sup>2</sup>. For comparison, Figures 7(a)–(c) show similar slices through a regular voxel array constructed off-line from the same data. The voxels were cubic, with the same dimension as a pixel in the original B-scans. Gaps in the voxel array were filled by averaging intensities in a  $5 \times 5 \times 5$  local neighbourhood (this took several minutes on a powerful workstation). Close inspection of the two sets of slices shows that the grey scale texture is better preserved by the voxel-less scheme, since there is *no* averaging of the raw intensity data.

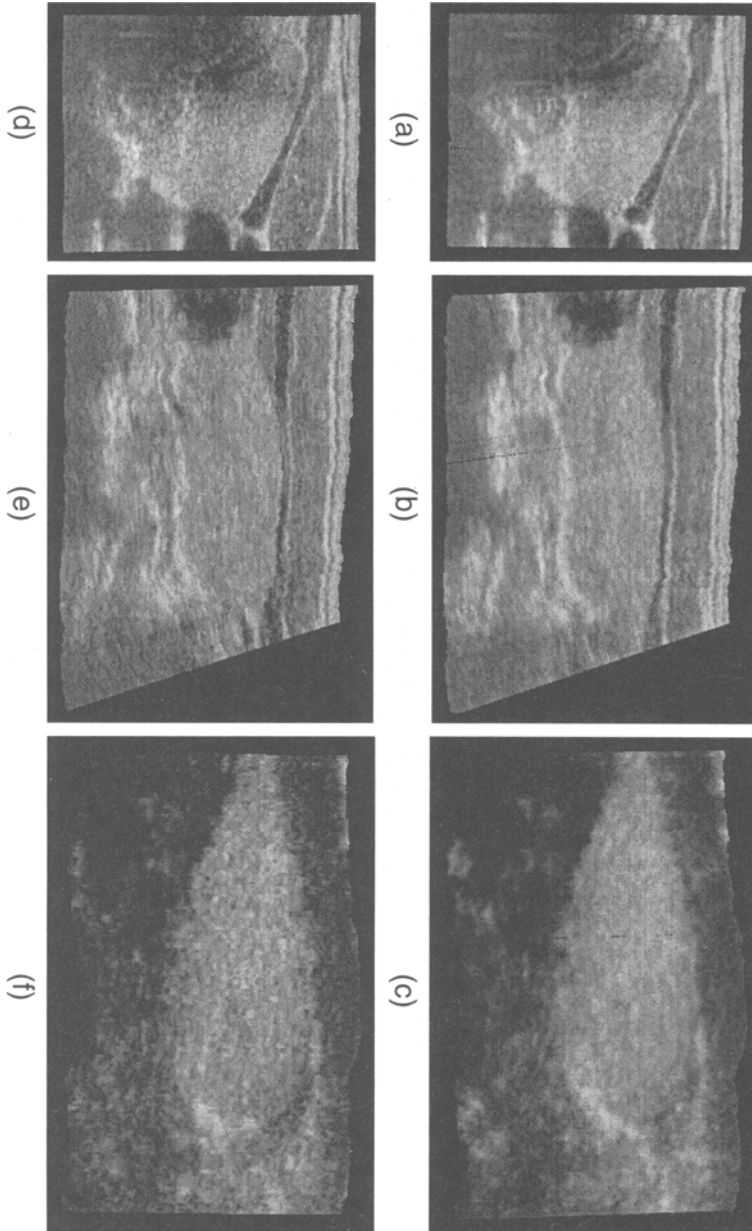
## 4 Conclusions

We have described two tools for real-time visualization of data from a 3D ultrasound system. By using an interpolation strategy that requires information from only one B-scan at a time, we are able to produce large panoramic images and interactively compute slices through the body at any angle. As the 3D ultrasound data does not need to be stored, this greatly reduces the amount of computer memory required to reconstruct an image.

## References

1. T. R. Nelson and T. T. Elvins. Visualization of 3D ultrasound data. *IEEE Computer Graphics and Applications*, pages 50–57, November 1993.
2. M. L. Skolnick. Estimation of ultrasound beam width in the elevation (section thickness) plane. *Radiology*, 180(1):286–288, 1991.

<sup>2</sup> In order to facilitate comparison with the voxel-based slices, these images have been constructed off-line from a single set of recorded B-scans. They are, however, mathematically and visually identical to those we generate in real-time.



**Fig. 7. Any-plane slicing of a human thyroid gland.** The figure shows results of a voxel-based (a)–(c) and voxel-less (d)–(f) approach. Slices (a) and (d) are virtually parallel to the B-scans. Slices (b) and (e) are orthogonal to these but the skin is still at the top: similar B-scans could have been obtained by rotating the probe through  $90^\circ$ . Slices (c) and (f) are located a constant depth below the skin, and could not have been obtained using conventional 2D ultrasound.