Quantitative visualisation of surfaces from volumetric data

Jonathan C. Carr[1], Andrew H. Gee[1], Richard W. Prager[1] & Kevin J. Dalton[2]
[1]Dept. of Engineering, University of Cambridge, Cambridge, UK. Email: jccc@eng.cam.ac.uk
[2]Dept. Obstetrics & Gynaecology, University of Cambridge, Cambridge, UK.

ABSTRACT

A 3D B-spline quasi-interpolant is used to extract smooth iso-value surfaces from volume data. In this technique discretization and partial voluming artifacts are reduced by approximating sampled data at voxel centres. Surface normals, necessary for realistic shading, are analytically defined by the approximating function rather than estimated by in an ad hoc way from the volume data. We consider an application where bone surfaces are revealed from CT data by ray-casting and the surfaces are then used to construct models and prostheses. Accurate determination and rendering of bone surfaces is required. A z-buffer shading technique is also described for improved rendering of surface depth-maps.

Keywords: Medical imaging, surface rendering, ray-casting, Computed Tomography (CT), prosthesis design

1 INTRODUCTION

Surfaces extracted by ray-casting techniques typically exhibit ripples which arise from interpolation and quantisation artifacts. Fig. 1(a) illustrates a typical surface rendered from CT data. For visualisation purposes these artifacts are masked by volume gradient shading methods [Hohne86]. However, their existence becomes important when quantitative surface measurements are required. In this paper we discuss the accurate rendering of surfaces and fit a function which approximates the sampled volume data thereby resulting in smoother surfaces (Fig. 1(b)). The specific application we consider is that of fashioning cranial prostheses to repair holes and defects in the skull. The problem is illustrated in Fig. 2 and discussed in [Carr97]. Computer graphics are used to extract surfaces from X-ray CT data. Models are constructed directly from the displayed surfaces using a Computer Numerical Controlled (CNC) mill. A mathematical surface is fitted to the corresponding depth-map which interpolates the data and extrapolates across defect regions where the skull surface is either missing or requires repair. A surgeon identifies these regions by 'painting' them on the rendered view using a workstation's mouse. The surgeon assesses the fitted surface on the workstation and a titanium prosthesis in the shape of the surface is automatically constructed. Accurate rendering of bone surfaces is critical to ensuring the good fit and appearance of the prosthesis since intra-operative manipulation of the prosthesis is limited. A binary segmentation of the volume data is implicit in this problem.

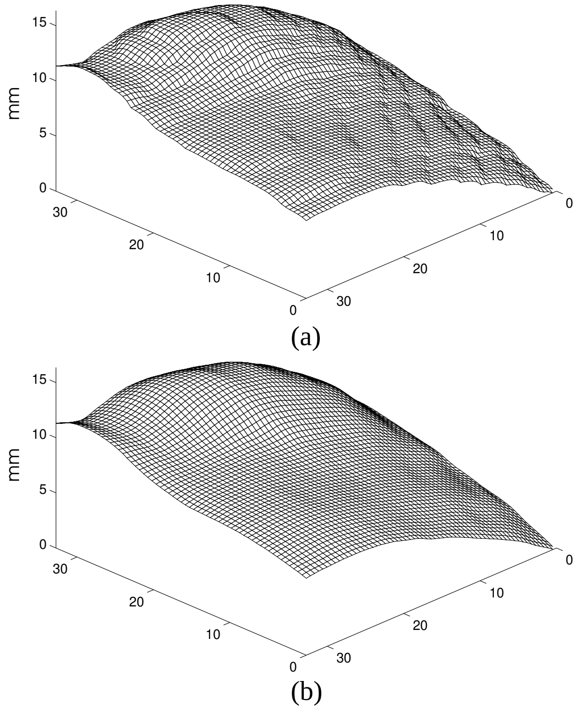

Figure 1: Comparison of ray-casting artifacts in bone surfaces rendered from CT data. (a) Conventional tri-linear interpolation, (b) B-spline approximation. The scales are in mm.

1.1 Medical graphics

The visualisation of 3D medical data differs from traditional computer graphics in two ways; (1) the data from which images are formed is captured from real objects rather than synthesised by a designer or generated mathematically from a computer model, (2) accurate depiction is more important than pleasing images. There are two volume visualisation paradigms used in medical imaging; *surface rendering* and *volume rendering.*

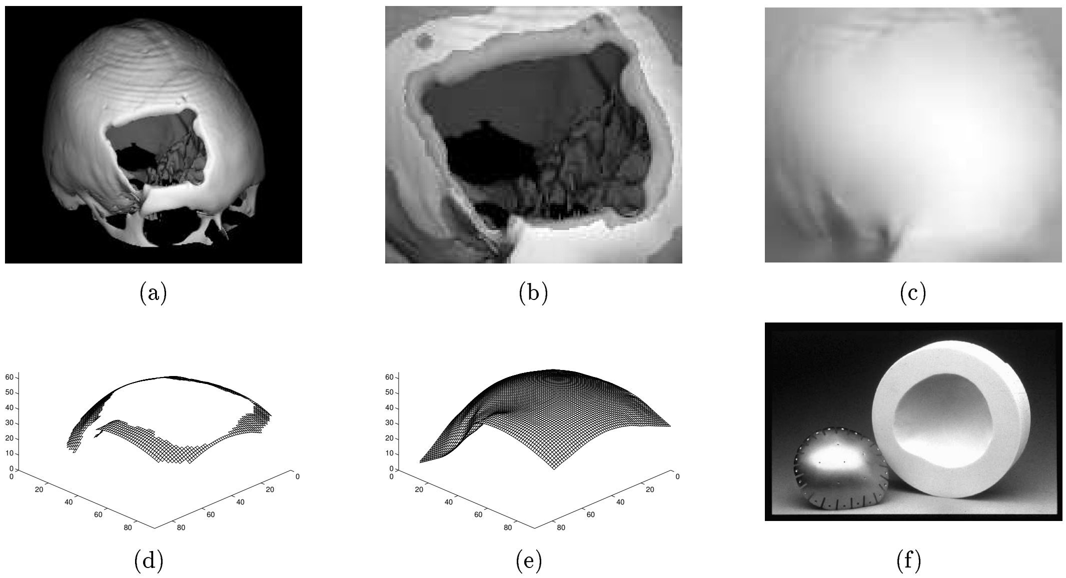

Figure 2: Production of a cranial prosthesis. (a) Bone surface rendered from a CT data set, (b) detailed view of the defect region with the support region for the prosthesis highlighted, (c) z-buffer rendering of the interpolated surface fitted to (b), (d-e) depth-maps corresponding to (b) and (c). (f) Finished titanium prosthesis with its corresponding mold, machined from epoxy-resin in the shape of the interpolated surface.

Surface rendering implicitly assumes that the data possess tangible surfaces that can be extracted and visualised. In effect a binary classification of the data volume is made. Traditional surface rendering algorithms convert the volume data into geometric primitives (e.g., a polygon mesh, a set of contours) and then render these to the screen using conventional computer graphics algorithms [Foley90,PHIGS86]. Some well-known examples are the *cuberille* technique [Chen85], the *marching cubes* algorithm [Loren87,Cline88] and the *dividing cubes* algorithm [Cline88].

Volume rendering avoids the binary classification implicit in a surface extraction process. Light is modelled as being partially attenuated and reflected as it passes through the data volume. The images generated are dependent on the assignment of properties such as opacity, colour, luminescence, etc, to the data [Drebl88,Levoy88].

Proponents of volume rendering argue that in cases where binary segmentation is difficult, such as when rendering ultrasound data [Nelso93], a semi-transparent modelling of an object can improve the visualisation. Thresholding is normally a robust segmentation technique for visualising bone surfaces from CT data but can fail when bone structures are thin. The result is apparent holes in the rendered surface which do not actually exist. Such fine bone structures occur in the orbits and nasal septum. Failure occurs because the voxel grey level represents the average density of tissue within a local neighbourhood. Consequently, along tissue boundaries and where structures are thin, this average may no longer be representative of either bone or soft tissue. This effect is known as *partial voluming* and afflicts most imaging modalities. For example, the analogous artifact in ultrasound is *beam width*. Volume rendering algorithms such as Levoy's [Levoy88] attempt to overcome this problem by using both grey-level value and grey-level gradient to assign transparency.

Although transparency methods can sometimes compensate for partial voluming effects and generate smoother images [Hohne90], the assignment of opacities and related parameters is arbitrary and a broad range of visual impressions can be obtained for the same object. It is difficult to know which rendering is the most accurate and how precisely the rendering represents the actual surface. [Tiede90] have attempted to quantify image quality in their study of rendering methods. They conclude, with particular reference to rendering bone surfaces from CT data, that if exact surfaces can be determined then non-transparent rendering often yields the best perception.

Volume and surface rendering strategies can be complementary depending on the goal — whether it be qualitative or quantitative. Indeed, hybrid schemes have been proposed by [Udupa93] and [Levoy90]. Irrespective of image 'quality', fuzzy or transparent renderings are not suitable for making quantitative measurements or for fitting surfaces due to the ambiguous depiction of anatomy.

1.2 Ray-casting

Traditional surface rendering techniques such as marching cubes and the cuberille technique generate a fine polygon mesh by sequential tessellation of a boundary with triangles or rectangles. Ideally, cubic voxels are required. Medical volumetric data rarely exhibits isotropic sampling. CT data volumes consist of slices which are usually 2-3mm apart, while resolution within the slices is typically sub-millimetre. Linear or shape-based interpolation [Herma92] between adjacent 2D data slices is therefore used to generate intermediate slices so that an array of approximately cubic voxels is achieved. More recently, ray-casting techniques have been used to render opaque surfaces directly without forming an intermediate surface representation of polygons. Tuy *et al* [Tuy84] were among the first to apply ray-casting to the display of binary volumetric data. Subsequent implementations have varied primarily in their use of transparency, shading and in coding details which improve the computational efficiency of the method.

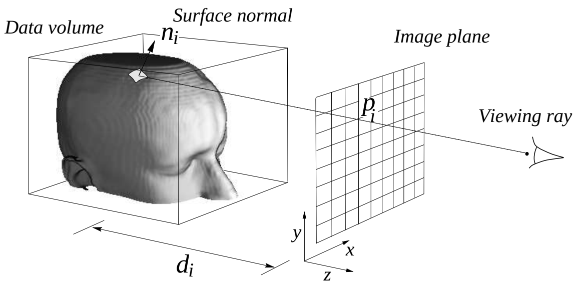

Figure 3: The ray-casting paradigm.

Fig. 3 illustrates the principle of the ray-casting method. Rays emanate from a viewpoint, pass through pixels in an image plane, and then intersect the data volume. For each pixel p_i in the image plane, a ray is tracked through the volume until either a surface is detected or the volume is exhausted. A surface is determined by thresholding the scalar intensity distribution. The distance to the surface, d_i, is recorded in a *z-buffer* along with the surface normal n_i. Intensity values are interpolated between voxel centres. Tri-linear interpolation, where an intermediate value is computed from the linear combination of the 8 nearest voxels, is commonly used. Within a boundary voxel the intensity is evaluated at fine increments along the ray until a threshold transition is detected or the voxel is exited. In this way a surface transition is determined with sub-voxel resolution and aliasing is avoided. Shading is fundamentally dependent on the determination of surface normals. Volume gradients are normally estimated at voxel centres from those voxels in a 3x3x3 neighbourhood [Hohne86]. Between voxels the gradients are interpolated and used to estimate surface normals. Consequently, an iso-valued surface is rendered. Fig. 3 suggests a perspective projection, but parallel rays are generally used in medical imaging to assist quantitative analysis.

For computational reasons an approximation to tri-linear interpolation is usually made which avoids incremental evaluation along a ray. This uses bi-linear interpolation to evaluate the scalar intensity at a ray's entry and exit points from a voxel neighbourhood. The intensity profile along the ray is then linearly interpolated between these two points. Using this method, a point in the data volume will generally be evaluated with slightly different values depending on the ray path taken through the point, i.e. for different views, the scalar distribution will appear to be multi-valued. However, if visual appearance is the sole concern, then this effect is not noticeable when voxels are comparable in size to pixels in the image plane. Both tri-linear interpolation and associated approximations result in surface artifacts (ripples) which, though sub-millimetre, are reproduced in titanium prostheses and models milled from the ray-cast surface. In Section 3 we model the scalar distribution piecewise with an approximating cubic spline. We argue that this reduces interpolation artifacts and provides a more faithful depiction of the true surface.

2 RELATED WORK

[Webbe90] has also tried to fit mathematical functions to volume data to improve the ray-casting algorithm. His method involves fitting a local bi-quadric surface as a function of two variables, $w = f(u, v)$. A 3x3x3 voxel neighbourhood determines coefficients of the polynomial f. In order to do this, a coordinate system u-v-w is chosen such that the rendered surface is locally a function of u and v. A surface which is vertical relative to the u-v plane cannot be represented. Choosing an appropriate u-v plane for each voxel is difficult and no such plane may exist in the general case. The method results in a series of local quadric surface patches which form a discontinuous surface. Webber demonstrates the approach by locally approximating the surface of a sphere and justifies the use of a discontinuous surface on the basis of the degree of precision commonly present in voxel data and the density of rays cast. Coordinate changes generate the greatest patch discontinuities and are to be avoided. However, the complexity of anatomical structures is such that many coordinate changes are likely. This method suffers from many difficulties because it is essentially a 2D approach to a 3D problem.

[Molle97] has investigated the use of cubic spline filters in volume rendering and applied them to MRI data. Their work builds on earlier research by Mitchell and Netravali [Mitch88] which discusses cubic interpolation in computer graphics from a signal processing point of view. They use a Fourier analysis of the approximation error and subjective tests based on 2D image interpolation to argue for a particular cubic spline filter. [Molle97] uses a Taylor series expansion to argue that the Catmull-Rom spline is the optimal interpolator. Interestingly, they choose a different filter, rather than the derivative of the interpolant, to determine gradients. In this paper we have found that, despite these previous analyses, a quasi-interpolant which does not exactly interpolate the data is useful when rendering structures known to be smooth and continuous. We believe this is because the theoretical analyses do not consider the presence of noise in the sampled data.

3 B-SPLINE APPROXIMATION

In this section we introduce a tensor-product cubic B-spline which quasi-interpolates a 3D scalar distribution sampled on a regular grid. Engineers are familiar with the use of parametric cubic B-splines to synthesise free-form surfaces in traditional computer graphics. In this paper we use cubic B-splines to approximate the continuous function which underlies the volumetric data. The spline control points are therefore derived from the points (voxel centres) where the function is known, rather than being specified by a designer.

Consider the 1D spline $f(x)$ specified over the interval $x_i \leq x \leq x_{i+1}$. $f(x)$ can be written as the weighted average of four points $P_{i-1}, P_i, P_{i+1}, P_{i+2}$, which are equally spaced along x. Without loss of generality, if the origin is translated to the point x_i and the interval $x_i \leq x \leq x_{i+1}$ scaled such that x varies over the range $0 \leq x \leq 1$, then a spline segment can be described by,

$$f(x) = TM \begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix}, \tag{1}$$

where

$$T = \begin{bmatrix} x^3 & x^2 & x & 1 \end{bmatrix}, \tag{2}$$

M is a 4x4 matrix of weights which are chosen to provide various degrees of continuity between adjacent segments at the endpoints $x = 0$ $x = 1$. In general, the spline does not pass through the control points $P_{i-1}, P_i, P_{i+1}, P_{i+2}$. Specifically, we consider the family of BC-splines where M is given by,

$$\frac{1}{6}\begin{bmatrix} \text{-B}-6C & \text{12-9B-6C} & -12+9B+6C & \text{B+6C} \\ \text{3B+12C} & -18+12B+6C & 18-15B-12C & \text{-6C} \\ \text{-3B}-6C & 0 & 3B+6C & 0 \\ B & 6-2B & B & 0 \end{bmatrix}.$$

This family is derived from the most general form for a symmetric cubic basis function. By requiring the value and first derivative to be continuous everywhere, the number of free parameters which determine the spline is reduced from eight to two (B, C). Note that the BC-spline family includes the cubic B-spline (1,0), the Cardinal cubic splines (0,C) and the Catmull-Rom spline (0,0.5). The Cardinal splines interpolate the control points. In the specific case of the B-spline (1,0), M becomes,

$$M = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}. \tag{3}$$

When the product in Eq. 1 is now evaluated, it can be seen that the B-spline is constrained to lie within the *convex hull* of the four control points since the coefficients of the four control points range in value between 0 and 1 and sum to 1 for $0 \leq x \leq 1$.

We now consider the analogous BC-spline in 3D where $\mathbf{x} = (x_1, x_2, x_3)$. In 3D the control points lie on a regular grid in R^3 and are denoted by $P_{i,j,k}$ where i, j, k are integer indices referring to grid locations where the data are known. The nodes of the grid are the voxel centres. The 3D spline is a weighted average of 64 control points in a 4x4x4 neighbourhood which

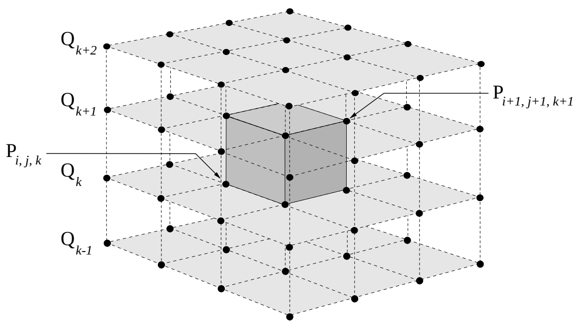

Figure 4: The geometry of the sixty-four control points in Eq. 5 which define the cubic spline in three dimensions.

describes the spatial distribution $f(\mathbf{x})$ within the volume bound by the eight central voxels (Fig. 4). If the locations of these voxels are scaled and shifted such that they lie within the unit cube, then the spline can be formulated as,

$$f(\mathbf{x}) = RM \begin{bmatrix} SM & Q_{k-1} & M^T T \\ SM & Q_k & M^T T \\ SM & Q_{k+1} & M^T T \\ SM & Q_{k+2} & M^T T \end{bmatrix}, \tag{4}$$

where

$$Q_k = \begin{bmatrix} P_{i-1,j-1,k} & P_{i,j-1,k} & P_{i+1,j-1,k} & P_{i+2,j-1,k} \\ P_{i-1,j,k} & P_{i,j,k} & P_{i+1,j,k} & P_{i+2,j,k} \\ P_{i-1,j+1,k} & P_{i,j+1,k} & P_{i+1,j+1,k} & P_{i+2,j+1,k} \\ P_{i-1,j+2,k} & P_{i,j+2,k} & P_{i+1,j+2,k} & P_{i+2,j+2,k} \end{bmatrix} \tag{5}$$

$$R = \begin{bmatrix} x_3^3 & x_3^2 & x_3 & 1 \end{bmatrix}, \tag{6}$$

$$S = \begin{bmatrix} x_2^3 & x_2^2 & x_2 & 1 \end{bmatrix}, \tag{7}$$

$$T = \begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \end{bmatrix}. \tag{8}$$

and $\quad 0 \leq x_1, x_2, x_3 \leq 1$

In the ray-casting paradigm, voxel values are used as control points which describe the scalar parameter being visualised in a piecewise fashion. The aggregate of BC-spline segments determines the scalar distribution over the entire data volume. The resulting spatial distribution is smooth and, in the case of the B-spline, exhibits second order continuity. The convex hull property of the B-spline means that the the fitted function is constrained to have values within the range of the voxel values forming the support. Other choices of B and C, such as the Catmull-Rom spline, do not exhibit this property. The B-spline is a *quasi-interpolant* of the voxel data since it approximates the voxel values at the interpolation nodes. It is not unreasonable to approximate the sampled data at voxel centres if the sampled data are noisy or do not represent the actual value of the parameter at the grid coordinates but are averages of the true parameter over the volume of the voxel. In such cases, the data can be viewed as having a noise component arising from the partial voluming effect.

In practice, the B-spline interpolant closely approximates the data except where sudden changes occur at a scale which is small relative to the size of the B-spline support. The piecewise nature of the spline interpolant avoids the propagation of spurious ripples due to sudden changes or discontinuities in the data.

4 APPLICATION

In the following examples, rays were cast into a data volume and an iso-value surface corresponding to bone was extracted. Threshold transitions along rays were not determined analytically but by searching along the viewing rays. The smallest step size was determined by the tolerance required by the application; $50\mu m$ was considered sufficient for machining prosthetic templates. The convex hull property of the approximating spline constrains it to have a value within the range of those voxels in the 4x4x4 neighbourhood which determine it. The spline is therefore only evaluated within those neighbourhoods where a voxel exceeds the surface threshold. Thus, evaluation at every voxel which a ray passes through is avoided. This is not the case for a general cubic spline since the spline may exhibit values outside the range of the local voxel values. When a ray terminates, the gradient $\nabla f(\mathbf{x})$, is computed analytically by differentiating Eq.4.

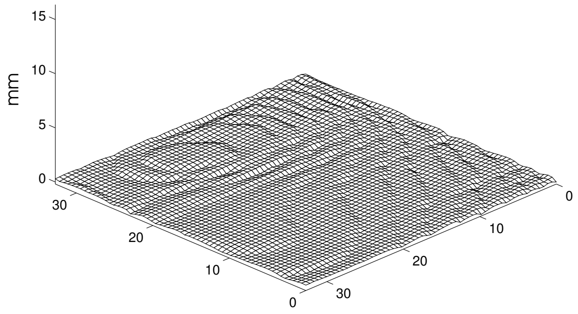

Figure 5: The difference between the tri-linear surface in Fig. 1(a) and the approximating spline surface in Fig. 1(b). The scales are in mm.

Fig. 1 illustrates part of a typical cranial surface extracted from CT data by ray-casting. The CT data consists of rectangular voxels of dimension 1x1x2.3mm. Fig. 1(a) was obtained using conventional tri-linear interpolation while Fig. 1(b) is the corresponding surface obtained using the spline approximation. The scales are in millimetres. Apparent in Fig. 1(a) is a step artifact occurring at voxel boundaries. This is largely absent from the appreciably smoother surface depicted in Fig. 1(b). In Fig. 5 the difference between the two surfaces is displayed. Surface differences range between -0.19mm and 0.45mm, the mean being 0.21mm with a standard deviation of 0.016mm. Significantly, the difference between the two surfaces is very small. Furthermore, Fig. 1(b) differs from Fig. 1(a) primarily in the absence of ripples.

In Fig. 1 it is not possible to directly determine which representation is the more accurate since the actual skull surface is not known, nor can it be easily measured. However, given that the difference between the two surfaces is very small (a fraction of a voxel), and that this is primarily due the ripples in the tri-linear surface, then it can be argued that the B-spline is the more likely representation since it is known that skulls do not contain such ripples.

Fig. 6 compares skull surfaces rendered using tri-linear and B-spline interpolation. The data set consists of 47 slices at 3mm spacings with 0.88mm resolution within the slices. Fig. 6(a) is the B-spline surface rendered using the B-spline gradient. Fig. 6(b) is the corresponding tri-linear surface rendered using a grey-level volume gradient. The same threshold has been applied in both cases. The B-spline surface gradient results in shading that is very similar to the grey-level volume gradient. In Fig. 6(c) the B-spline depth-map is shaded using z-buffer gradients determined by the algorithm described in Appendix A. Fig. 6(d) is the corresponding z-buffer shaded image for the tri-linear ray-cast surface. The smoother B-spline surface is particularly apparent in the z-buffer surface renderings, near the orbits of the eyes. The observation that the B-spline gradient produces surface shading very similar to that of the grey-level volume gradient is typical of the B-spline interpolant. Fig. 6(c) illustrates that a high quality image, similar to that achieved using volume shading, can be obtained with z-buffer shading if the underlying surface is smooth. The popular grey-level volume shading method (Fig. 6(b)) merely masks inadequacies in the extracted surface while z-buffer shading accurately depicts them (Fig. 6(d)).

5 DISCUSSION

The B-spline approximation developed in Section 3 uses a uniform parameterisation which means that the continuity properties are invariant under an affine transformation of the data points. Consequently, the technique is applicable when the data lie on a regular cubic grid or can be mapped on to a regular cubic grid by an appropriate linear transformation. If such a transformation cannot be found, then a parametric representation can be formulated where f, x_1, x_2, x_3 are each written as a function of three variables, say r, s, t, which vary between 0 and 1, i.e. $f(r,s,t), x_1(r,s,t), x_2(r,s,t), x_3(r,s,t)$. However, the difficulty with a parametric representation is that evaluating f at a point (x_1, x_2, x_3) requires finding the corresponding point in the parametric space. This involves solving third order polynomials in three variables. Consequently, multiple solutions are possible. Although $f(r,s,t)$ is a single valued function, $f(x_1, x_2, x_3)$ may be multi-valued and this has no physical meaning in the medical imaging context.

6 CONCLUSION

The B-spline approximation improves the rendering of surfaces from volume data known *a priori* to be locally smooth and continuous. It is less susceptible to noise in the sampled data and therefore produces smoother surfaces. Gradients can be determined analytically. In the vicinity of thin bone the B-spline interpolant does not perform as well as other splines such as the Catmull-Rom spline favoured by [Molle97]. This is due to the convex hull property

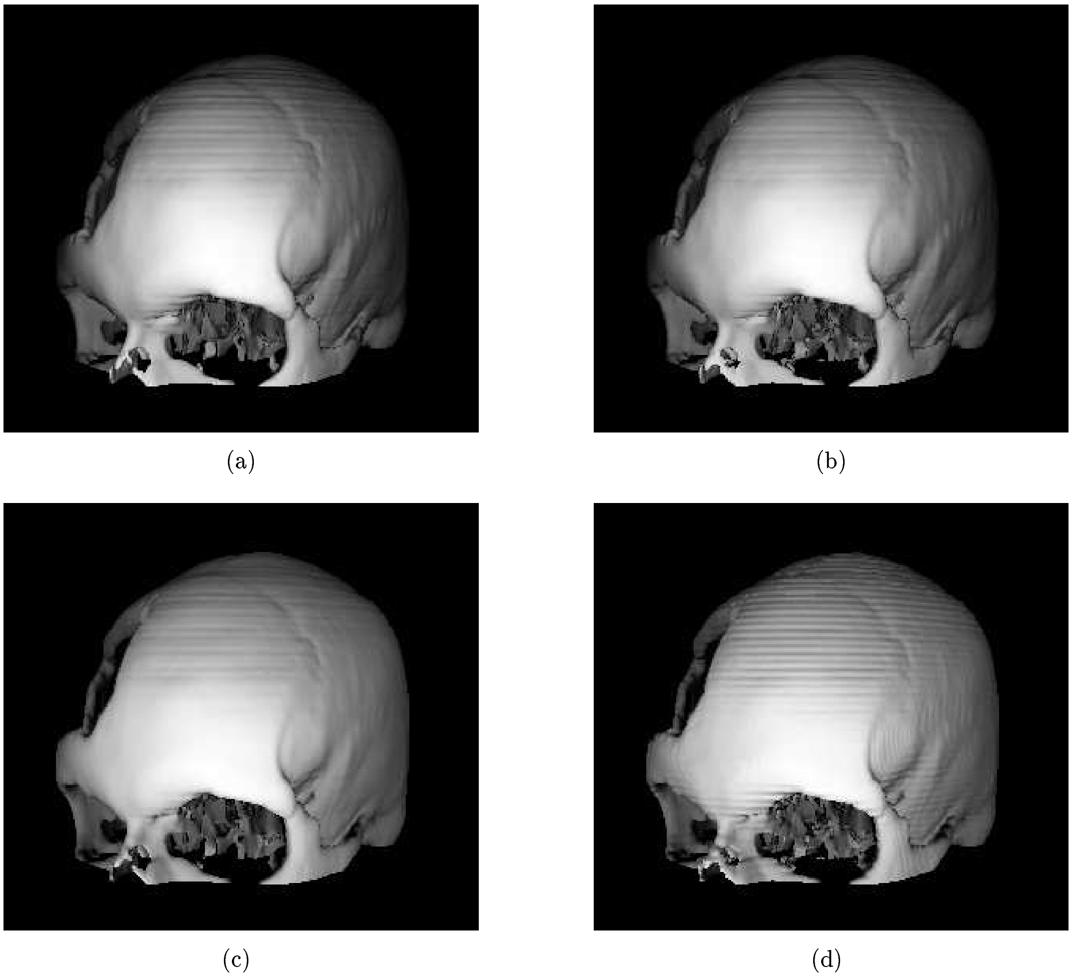

Figure 6: Comparison between tri-linear and B-spline surface renderings. (a) B-spline depth-map shaded using the B-spline gradient, (b) tri-linear depth-map shaded using a grey-level volume gradient, (c) B-spline depth-map shaded using the z-buffer gradient, (d) tri-linear depth-map shaded using the z-buffer gradient.

of the B-spline which means that no data value outside the range of the control points can be generated. The Catmull-Rom spline, for example, is not constrained by the convex hull property. In the application considered in this paper we are interested in fitting surfaces to regions where the skull surface is smooth and free of defect. The application is ideal in this respect because regions where discontinuities occur, which are more likely to result in rendering errors, are usually rejected. Surfaces are fitted only to those regions where the bone is well defined.

Shading using volume gradients can mask interpolation artifacts in a ray-cast surface. These become important when models are to be constructed from the rendered surface. Analytically determined gradients are preferable. However, accurate, high quality shading is possible when gradients are determined from the z-buffer, provided discontinuities in the z-buffer are considered (Appendix A).

Acknowledgements

This work was supported by the SOLUS-3D European project (contract no. BMH4-CT-95-0476).

7 REFERENCES

[Carr97] Carr, J. C., Fright, W. R., and Beatson, R. K., Interpolation with radial basis functions for medical imaging, *IEEE Trans. on Medical Imaging*, Vol.16, No.1, pp96–107, 1997.

[Chen85] Chen, G. T. Y., Herman, G. T., Reynolds, R. A., and Udupa, J. K., Surface shading in the cuberille environment, *IEEE Computer Graphics and Applications*, Vol.5, No.12, pp33–43, 1985.

[Cline88] Cline, H. E., Lorensen, W. E., Ludke, S., Crawford, C. R., and Teeter, B. C., Two algorithms for the three-dimensional reconstruction of tomograms, *Medical Physics*, Vol.15, No.3, pp320–327, 1988.

[Drebl88] Dreblin, R. A., Carpenter, L., and Hanrahan, P., Volume rendering, *Computer Graphics*, Vol.22, No.4, pp51–58, 1988.

[Foley90] Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F., *Computer Graphics Principles and Practice*, Addison-Wesley, Reading, Mass., 1990.

[Gordo85] Gordon, D., Image space shading of 3-Dimensional objects, *Computer Vision, Graphics, and Image Processing*, Vol.29, pp361–376, 1985.

[Herma92] Herman, G. T., Zheng, J., and Bucholtz, C. A., Shape-based interpolation, *IEEE Computer Graphics and Applications*, May, pp69–70, 1992.

[Hohne86] Hohne, K. H., and Bernstein, R., Shading 3D images from ct using gray-level gradients, *IEEE Trans. on Medical Imaging*, Vol.5, No.1, pp45–47, 1986.

[Hohne90] Hohne, K. H., Bomans, M., Pommert, A., Riemer, M., Schiers, C., Tiede, U., and Wiebecke, G., 3D visualization of tomographic volume data using the generalised voxel model, *The Visual Computer*,Vol.6, No.1, pp28–36, 1990.

[Levoy88] Levoy, M., Display of surfaces from volume data, *IEEE Computer Graphics and Applications*, May, pp29–37, 1988.

[Levoy90] Levoy, M., A hybrid ray tracer for rendering polygon and volume data, *IEEE Computer Graphics and Applications*, Mar., pp33–40, 1990.

[Loren87] Lorensen, W. E., and Cline, H. E., Marching cubes A high resolution 3D surface construction algorithm, *Computer Graphics*, Vol.21, No.4, pp163–169, 1987.

[Mitch88] Mitchell, D. and Netravali, A., Reconstruction Filters in Computer Graphics, *Computer Graphics*, Vol.22, No.4, pp221-228, 1988.

[Molle97] Moller, T. and Machiraju, R. and Mueller, K. and Yagel, R., Evaluation and Design of Filters Using a Taylor Series Expansion, *IEEE trans. Visualisation and Computer Graphics*, Vol.3, No.2, pp184-199, 1997.

[Nelso93] Nelson, T. R., and Elvins, T. T., Visualisation of 3D ultrasound data, *IEEE Computer Graphics and Applications*, Nov., pp50–57, 1993.

[PHIGS86] Shuey, D., Bailey, D., and Morrissey, T. P., PHIGS a standard, dynamic, interactive graphics interface, *IEEE Computer Graphics and Applications*, No.6, pp50–57, 1986.

[Tiede90] Tiede, U., Heinz, K., Bomans, M., Pommert, A., Riemer, M., and Wiebecke, G., Investigation of medical 3D-rendering algorithms, *IEEE Computer Graphics and Applications*, Mar., pp41–53, 1990.

[Tuy84] Tuy, H. K., and Tuy, L. T., Direct 2-D display of 3-D objects, *IEEE Computer Graphics and Applications*, Vol.4, No. 10, pp29–33, 1984.

[Udupa93] Udupa, J. K., and Odhner, D., Shell rendering. *IEEE Computer Graphics and Applications*, Nov., pp58–67, 1993.

[Webbe90] Webber, R. E., Ray tracing voxel based data via biquadratic local surface interpolation, *The Visual Computer*, Vol.6, No.1, pp8–15, 1990.

APPENDIX A: z-buffer shading

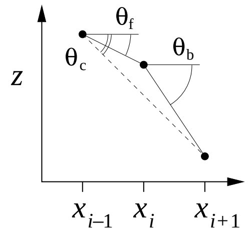

Figure 7: Geometry for estimating $\frac{\partial z}{\partial x}$ from the z-buffer.

Surface gradients can be estimated from the z-buffer and are required when an analytic expression or volume gradient are not available. The z-buffer is viewed as a function of two variables $z = z(x, y)$ and the surface normal $\mathbf{n}$ is obtained from the vector ∇z.

$$\mathbf{n} = (-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1). \tag{9}$$

Consider the z-buffer profile depicted in Fig. 7. To a first approximation, $\frac{\partial z}{\partial x}$ at x_i can be estimated from the central difference, $(z_{i+1} - z_{i-1})/(x_{i+1} - x_{i-1})$. Unfortunately the central difference will fail when a discontinuity occurs in the z-buffer due to one surface occluding another (Fig. 8). [Gordo85] has proposed a method which takes a weighted average of the forward $z_{i+1} - z_i$ and backward $z_i - z_{i-1}$ differences to provide a more reliable estimate of surface gradient from parallel projections. We use a modified version which attempts to explicitly identify discontinuities in the z-buffer corresponding to partially occluding surfaces.

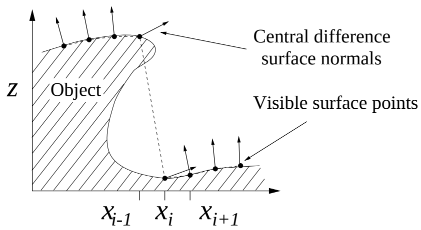

Figure 8: Failure to correctly estimate surface normals when a discontinuity occurs in the z-buffer.

Consider determining $\frac{\partial z}{\partial x}$ at the point x_i (Fig. 7) where the forward, backward and central differences are expressed by the angles, θ_f, θ_b and θ_c, respectively. Note that the x_i's need not necessarily be equally spaced. Fig. 10 summarises the algorithm used to estimate $\frac{\partial z}{\partial x}$ at the point x_i. Nine cases are sequentially considered. These are illustrated in Fig. 11. The first three cases deal with the possibility of one or both of the adjacent Z-values being *undefined*, which occurs when the corresponding ray does not intersect a surface. The following six cases consider the prospect of a discontinuity occurring in the intervals $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$.

(a) (b) (c)

Figure 9: Comparison between z-buffer shading methods (detail from Fig. 6(d)). (a) z-buffer shading using the central difference gradient estimate, (b) z-buffer shading using the method described in Fig. 10, (c) shading with the grey-level volume gradient.

Case	**Conditions**	$(\frac{\partial z}{\partial x})_i$
1	$z_{i-1} = \textit{undefined}$, $z_{i+1} = \textit{undefined}$	0
2	$z_{i-1} = \textit{undefined}$	$\tan\theta_b$
3	$z_{i+1} = \textit{undefined}$	$\tan\theta_f$
4	$\lvert\theta_b - \theta_f\rvert < \Delta\theta_{max}$	$\tan\theta_c$
5	$\lvert\theta_f\rvert > \theta_{max}$, $\lvert\theta_b\rvert > \theta_{max}$, $\theta_f\theta_b < 0$	0
6	$\lvert\theta_f\rvert > \theta_{max}$, $\lvert\theta_b\rvert > \theta_{max}$, $\theta_f\theta_b \geq 0$	$\tan\theta_c$
7	$\lvert\theta_f\rvert > \theta_{max}$	$\tan\theta_b$
8	$\lvert\theta_b\rvert > \theta_{max}$	$\tan\theta_f$
9	*default case*	$\tan\theta_c$

Figure 10: Algorithm for estimating $(\frac{\partial z}{\partial x})_i$ from the z-buffer (Fig. 7). The cases are considered in sequential order, where $\frac{-\pi}{2} < \theta_b, \theta_f < \frac{-\pi}{2}$. The constants $\Delta\theta_{max}$ and $\theta_{max} > 0$.

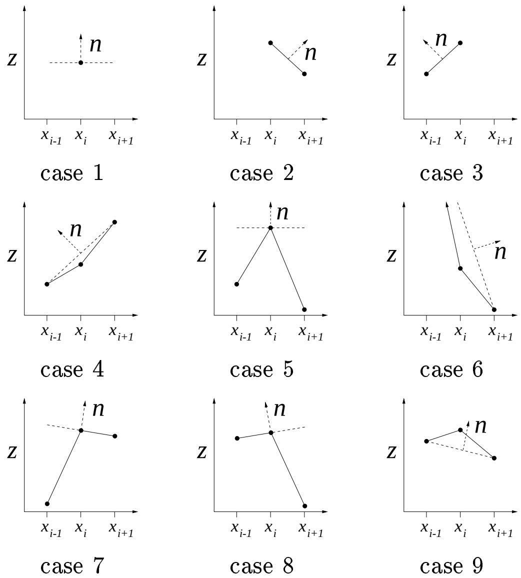

Figure 11: Illustration of the 9 z-buffer cases identified in Fig. 10.

A large change in depth between two adjacent points in the depth-map alone is not indicative of a discontinuity in the surface profile. A smooth continuous surface segment oriented obliquely to the viewing direction may also exhibit a steep profile. The change in gradient must also be considered when determining whether a discontinuity occurs. Gordon's weight allocation indirectly achieves this in certain cases.

As with Gordon's method two parameters, θ_{max} and $\Delta\theta_{max}$, are used to detect discontinuities. θ_{max} relates to the magnitude of the forward and backward differences while $\Delta\theta_{max}$ relates to the magnitude of the change in gradient between the forward and backward differences. $\Delta\theta_{max}$ determines the maximum permissible change in surface gradient. If the change in gradient, $|\theta_b - \theta_f|$, is less than $\Delta\theta_{max}$, then no discontinuity is deemed to have occurred in the vicinity of x_i, regardless of whether the gradients over the intervals $[x_{i-1}, x_i]$, and $[x_i, x_{i+1}]$ exceed the threshold θ_{max} (case 4).

If a significant change in gradient occurs at x_i, then the magnitude of the forward and backward differences are considered. In case 5 the magnitudes of θ_f and θ_b are compared to the threshold θ_{max}. If both exceed this threshold and are of opposite sign, then a 'spike' (or an inverted spike) is deemed to have occurred. Both intervals $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$ therefore contain a discontinuity. If both gradients exceed the threshold θ_{max}, but have the same sign, then no discontinuity is deemed to have occurred even though the change in slope exceeds $\Delta\theta_{max}$. Cases 7 and 8 consider the more usual occurrence where a single discontinuity occurs, either in the interval $[x_{i-1}, x_i]$ (case 7) or $[x_i, x_{i+1}]$ (case 8). Case 9, the default case, corresponds to the situation where a significant change in slope has occurred, $|\theta_b - \theta_f| > \Delta\theta_{max}$, but neither θ_f nor θ_b exceeds the threshold θ_{max}. In this case no discontinuity is deemed to have occurred.

From experience, the values of 65° for θ_{max} and 20° for $\Delta\theta_{max}$ have been found to produce good results. Fig. 9 examines the region of the zygomatic arch from Fig. 6(d) in detail. Fig. 9(a) is shaded from the z-buffer using the central difference method. Fig. 9(b) is shaded from the z-buffer using the method described in Fig. 10. For comparison, Fig. 9(c) is the volume gradient shaded image. The new algorithm has correctly assigned normals in most cases, eliminating the 'shadow' artifact apparent in Fig. 9(a) and producing a result similar to the volume shaded one. Fig. 9(b) differs from 9(c) primarily in the correct depiction of ripples resulting from tri-linear interpolation used in the ray-caster.