3D SHAPE RECONSTRUCTION USING VOLUME INTERSECTION TECHNIQUES

Jonathan C. Carr

CUED/F-INFENG/TR 300

September 1997

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

E-mail: `jccc@eng.cam.ac.uk`

3D Shape Reconstruction using Volume Intersection Techniques

Abstract

This paper presents a technique for reconstructing objects from noisy boundary data that are scattered, unorganised and incomplete. Volume intersection algorithms are used to reconstruct incomplete objects from their silhouettes. An imagined light source is moved about the data and the cumulative amount of 'light' seen at each point in space is interpreted as indicating the likelihood that the point is inside the object. The object data need not be uniformly distributed nor exclusively come from the surface of the object. Explicit identification of false object data and distinction between surface and interior data are avoided. A limitation of volume intersection algorithms is their inability to reconstruct concave surfaces. We show how the dependency of the visual hull of an object on the viewing region can be used to resolve concavities. The novel concept of a localised viewing region is introduced and shown to improve the ability of the method to reconstruct complex shapes in the presence of noise. Algorithms for 2D pixel and 3D voxel data are described and applied to 3D ultrasound data.

1 Introduction

Volume intersection methods are a popular way of obtaining shape from silhouette [2, 3, 4, 5, 6, 7]. Sources of silhouette data include laser range data, video and thermal imaging [10]. A silhouette can be thought of as a binary valued projection of an object space. A set of silhouettes specifies a bounding volume obtained by back-projecting each silhouette and finding the associated intersection volume. This volume approximates the object to a greater or lesser extent depending on the silhouette viewpoints and the object itself.

The two main aims of shape-from-silhouette research have been to recognise a known object from its silhouette and to estimate its position and orientation [11, 10, 13, 12] or, in the case of robot navigation and acquiring 3D models, to reconstruct an unknown object from a set of silhouettes which it has generated [14, 4]. In this paper, volume intersection methods are applied to a new type of problem: that of reconstructing the shape of *incomplete* objects. Projections of incomplete object data are used not only to reconstruct the given object data but also to interpolate across missing data.

The paper is organised as follows. Section 2 discusses related work and Section 3 introduces the principle of the reconstruction method. The constraints placed on a surface formed by this approach are not obvious and Section 4 attempts to shed light on these. In Section 5, we show how the method is realised by a modified volume intersection algorithm. In Section 6 the algorithm is extended to real-valued data and in Section 7 an iterative version which overcomes the major difficulties of the initial algorithm is introduced. Techniques for handling the problem of reconstructing cavities within a shape are discussed in Section 8. The reconstruction method is then applied to 3D voxel data in Section 9 and results are presented for a 3D ultrasound phantom.

2 Related work

2.1 Shape from silhouette

Two types of volume intersection algorithms are known for extracting shape from silhouette, those that employ a slowly changing viewpoint between image frames [4] and those based on a sparse set of distinct views [6]. In both approaches each image defines a cone of rays formed by back-projecting the silhouette. The object is described as the volume which lies within all the cones. If every possible viewpoint outside the convex hull of the object were used, the volume that lay within all the cones would be the object's visual hull [2], or more specifically, the *external* visual hull. The object $\mathbf{O}$, its visual hull $\mathbf{VH(O)}$ and its convex hull $\mathbf{CV(O)}$ are related as follows, $\mathbf{O} \leq \mathbf{VH(O)} \leq \mathbf{CV(O)}$. The object may only be exactly reconstructed where the object surface is coincident with the surface of the visual hull. Laurentini [1, 2] has considered the question as to how much of the shape of the unknown object can be inferred from the volume formed by the intersection of back-projected silhouettes and presents algorithms for computing the visual hull of polygonal sets, polyhedra and solids of revolution. While this work is of theoretical importance, it does not tackle the practical problem of reconstructing real objects from incomplete data in the presence of noise.

Although the volume intersection methods are inherently limited in their ability to reconstruct concave or self-occluding surfaces, researchers have studied the problem of identifying regions where concavity is likely [4, 10]. Michel and Nandhakumar [10] suggests that by detecting concavities in the silhouette of an object and partitioning the silhouettes it is possible to expose subparts that would otherwise be occluded. In the special case of reconstructing a rotating object from continuous varying views, Zheng [4] uses discontinuities in the silhouette profile and the rate of change in the silhouette with respect to the angle of rotation to identify regions which cannot be reconstructed. In Section 8 we show that it is possible to resolve concavities when reconstructing the shape of incomplete objects by introducing a localised viewing region.

Much research has focussed on efficient implementation and storage of reconstructions [5, 6, 7]. The problem of reconstruction from restricted viewing directions [5] has also been studied. In active vision systems researchers consider the problem of automatically choosing new viewing directions in order to efficiently constrain the reconstructed volume [14, 15]. Other work has focussed on approximating silhouettes with polygons [5] and reconstruction from polygonal silhouettes [6]. Volume intersection techniques can be further divided into those in which the reconstruction is object-centred [3] and those where the reconstruction is dependent on an external co-ordinate system [6]. In this paper we are concerned with object-independent representations.

2.2 Shape reconstruction from scattered object data

We are interested in the problem of fitting a surface to scattered, unorganised data associated with an isolated object. The data is unorganised in the sense that the adjacency relation between surface points is not known. Two points may be in close proximity in 3-space but not necessarily adjacent on a surface passing through them. When surface data is obtained from a laser surface scanner, the relationship between surface points is usually known by virtue of the methodical way in which an object is scanned. Consequently, many techniques for recovering surfaces from range data require the adjacency relationship to be known. Constructing surfaces from unorganised range data [8] is relevant to stereo vision, free-hand laser scanning and the extraction of surfaces from 3D edge data [16]. In this paper we consider the specific example of 3D ultrasound data acquired from a free-hand system [17, 18]. It is clinically desirable to be able to fit surfaces and measure organ volumes from ultrasound data. Ultrasound detects changes in the acoustic impedance of the media through which it propagates. These changes are often associated with the boundaries of objects of interest. Small changes in acoustic impedance within an object mean that return echoes are not always associated with the boundary of the object to be reconstructed. The nature of free-hand ultrasound data means that the raw data do not lie on a regular grid and some regions are more densely sampled than others. Consequently, we are interested in the problem where the object data have the following properties: (1) the data are not exclusively surface data, (2) the data are non uniformly distributed, (3) the data contains noise, (4) no *a priori* knowledge of the object's shape is available.

Other approaches to the problem of reconstructing the shape of incomplete objects do not address all these properties. For example, Hoppe *et al.* [8] have considered a the problem of fitting surfaces to unorganised range data, but the input data are assumed to be exclusively surface data and associated with a single surface. A uniform sampling density is also assumed. Cheng *et al.* [9] have studied the problem of boundary extraction from 3D ultrasound data, but they consider the specific case of extracting the surface of the fetus from a regular voxel array. Belohlavek *et al.* [19] propose a method for extracting volumes from 3D ultrasound data which involves the identification of regions of interest, contrast enhancement, thresholding, and morphological filtering. However, these processes require significant user interaction. Snyder *et al.* [16] have developed a method for closing gaps in two and three dimensional edge data, a problem with similar characteristics. Their method is applicable to binary data, where the presence or absence of an edge is specified, but does not generalise to continuous-valued data in an obvious manner. In the analysis of real-world data, including ultrasound data, it is advantageous to associate a confidence with the presence of an object boundary. This results in a continuous-valued distribution.

An alternative approach to this type of problem involves fitting deformable models such as superquadrics to the data [20]. These methods require prior knowledge of the size and shape of objects to be reconstructed. Performance depends on the specification of parameters which control the surface and determine the relative influence of data points. Determining these parameters from actual data to achieve optimum fit is difficult, especially in the presence of noise [21]. Model fitting procedures constrain the topology of objects which can be reconstructed. Superquadric surfaces, for example, are parameterised in terms of latitude and longitude, implying a spherical topology. Ideally, a general approach is sought that will impose few restrictions on reconstructible shapes. The novel approach presented here is an attempt to form surfaces with minimal additional information.

The shape reconstruction problem might also be viewed as an unsupervised clustering problem where points are classified as belonging or not belonging to an object. There are many such techniques but most do not explicitly define boundary surfaces which interpolate the data. Examples of clustering approaches include minimum spanning trees, Voronoi diagrams and the generalised Hough Transform [22]. Most of these techniques use some sort of Euclidean distance measure or a specific geometric feature as the metric for clustering points. In this paper the novel metric used to cluster points is the visibility of a boundary surface which encloses the data. This metric has interesting properties which enable a surface to be fitted to sparse, noisy data when the density of the data varies throughout the reconstruction domain.

3 The shape reconstruction problem

We begin by considering a single object isolated from its surroundings and sampled in an irregular manner to give a scattered distribution of points deemed to belong to the object. In three dimensions we wish to reconstruct a surface enclosing the object data.

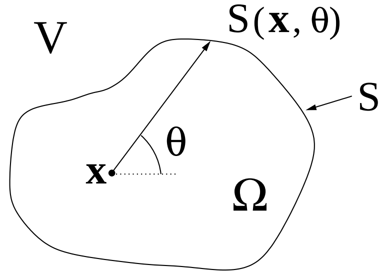


Figure 1: Definition of the reconstruction space, Ω, in two dimensions.

Before continuing further it is necessary to state the shape reconstruction problem formally. Consider a vector space $\mathbf{V}$, in which $\mathbf{x}$ defines a point. Let Ω be a subspace of $\mathbf{V}$ bounded by a simple closed surface S. Ω is termed the reconstruction space and must contain the object to be reconstructed. The region outside Ω is analogous to Laurentini's viewing region R [2]. S defines the boundary of the viewing region. $O(\mathbf{x})$ defines an attribute of the subspace. Initially we consider a binary value which defines whether or not a point belongs to the object of interest. Ω_{obj} denotes the region of the space occupied by the object of interest. $\hat{\Omega}_{obj}$ denotes a subset of Ω_{obj}, corresponding to the incomplete object data, $\hat{\Omega}_{obj} \subset \Omega_{obj} \subset \Omega \subset \mathbf{V}$. This is the ideal case. In practice $\hat{\Omega}_{obj}$ may contain noise; hence $\hat{\Omega}_{obj} \not\subset \Omega_{obj}$. The problem is to reconstruct Ω_{obj} given the partial object data, $\hat{\Omega}_{obj}$.

The approach developed here is also applicable to the analogous two dimensional problem of reconstructing a closed curve from partial edge data in a plane. Because the method proposed generalises straightforwardly from two to three dimensions, the two dimensional case is described initially.

The method involves determining an *occluded angle*, denoted by θ_{occ}, at each point in Ω. A ray emanating from a point $\mathbf{x}$, where $\mathbf{x} \in \Omega$, is occluded if it intersects object data prior to reaching the reconstruction boundary, S. In two dimensions (Fig. 1) a ray emanating from $\mathbf{x}$ can be defined by the point $\mathbf{x}$ and the ray's orientation relative to an arbitrary axis, θ, where $\theta \in [0, 2\pi]$. $S(\mathbf{x}, \theta)$ is the point where the ray *first* intersects the reconstruction space boundary. Note that since the definition of S is arbitrary, a ray may intersect S more than once. The occluded angle at $\mathbf{x}$, $\theta_{occ}(\mathbf{x})$, may be found by integration over all rays emanating from $\mathbf{x}$ and is expressed in Eq. (1).

$$\theta_{occ}(\mathbf{x}) = \int_0^{2\pi} F(\mathbf{x}, \theta) d\theta \tag{1}$$

where

$$F(\mathbf{x}, \theta) = \begin{cases} 1 & \text{if } S \text{ is occluded along the ray } \theta \text{ emanating from } \mathbf{x}, \\ 0 & \text{otherwise.} \end{cases}$$

From Eq. (1) it can be seen that θ_{occ} may range from 0 to 2π, where 0 corresponds to no occlusion and 2π implies that $\mathbf{x}$ is totally enclosed by object data. In Section 5 we show that the occluded angle is related to the intersection of back-projected silhouettes.

We can now calculate $\theta_{occ}(\mathbf{x})$ at each point in Ω and form the occluded angle distribution. Points inside the object may be expected to have a higher occluded angle than those outside the object. We therefore regard the occluded angle as a measure of the likelihood that a point is contained within the object. Initially we consider the simple case where points in the occluded angle distribution above a particular threshold are viewed as candidate solution regions. In this case, the choice of a threshold determines the shape of the reconstructed object.

(a) (b)

Figure 2: (a) Example of a shape which cannot be reconstructed, (b) alternative choice for S that allows the cavity to be reconstructed.

Shapes which can be reconstructed are restricted to those where every point along the perimeter can 'see' some part of the reconstruction space boundary, S. In other words, no point on the perimeter is occluded from all points on S. This limitation on shapes which can be reconstructed is dependent on S. Fig. 2(a) illustrates a shape for which the above condition is not met. The object occludes itself relative to S in the vicinity of the cavity. For the same shape this is no longer the case in Fig. 2(b) due to a different choice of S.

4 Analysis of occluded angle distribution

The occluded angle distribution, $\theta_{occ}(\mathbf{x})$, depends on S and $\hat{\Omega}_{obj}$. In general, $\theta_{occ}(\mathbf{x})$, evaluated at some point $\mathbf{x}$, $\mathbf{x} \in \Omega$, is a function of all the object data and not just that within the local neighbourhood of $\mathbf{x}$. This, and the observation that the $\theta_{occ}(\mathbf{x})$ distribution may contain discontinuities, and hence is only piecewise differentiable, makes an analysis difficult. However, some important general characteristics can be inferred by considering a few specific objects which might otherwise be considered of little practical significance. Table 1 lists properties of the occluded angle distribution for seven such cases. An infinite reconstruction space which encloses the convex hull of the object data is chosen so that the reconstruction dependency on S can be ignored.

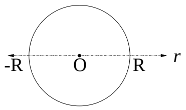
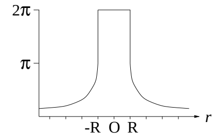
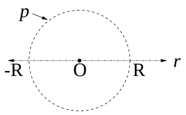
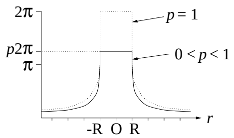
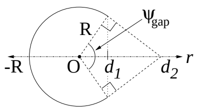
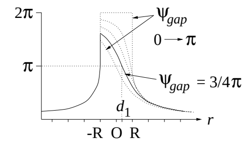
	Case	θ_{occ} profile
1.	Inside a closed contour.	$\theta_{occ} = 2\pi$
2.	On the boundary of a smooth, simple convex curve.	$\theta_{occ} = \pi$
3.	On the boundary of an arbitrary smooth curve.	$\theta_{occ} \geq \pi$
4.	Closed circle. 	 $\theta_{occ}(r) = 2\pi \quad 0 < r < R \quad (2)$ $\theta_{occ}(r) = 2\sin^{-1}(R/r) \quad \lvert r\rvert \geq R \quad (3)$
5.	Partial circle where p is the proportion of the circle present and the gaps are infinitely small and uniformly distributed along the circumference, $0 \leq p \leq 1$. 	 $\theta_{occ}(r) = p2\pi \quad \lvert r\rvert < R \quad (4)$ $\theta_{occ}(r) = 2(2p - p^2)\sin^{-1}(R/r) \quad \lvert r\rvert \geq R \quad (5)$
6.	Circle containing a single gap defined by the angle ψ_{gap}, where $0 \leq \psi_{gap} \leq \pi$. The profile considered is along the axis of symmetry. 	 $\theta_{occ}(r) = 2\sin^{-1}(R/r) \quad r \leq -R, r \geq d_2$ $\theta_{occ}(r) = \pi + 2\tan^{-1}\frac{d_1 - r}{R\sin(\frac{\psi_{gap}}{2})} \quad -R < r < d_2$ where $d_1 = R\cos(\frac{\psi_{gap}}{2})$, $d_2 = R/\cos(\frac{\psi_{gap}}{2})$.
7.	Inside a completely convex object O bound by a curve containing n discrete gaps.	$\theta_{occ}(x) = 2\pi - \sum_{i=1}^{n} \overline{\theta_{occ_i}(x)}$ where $\overline{\theta_{occ_i}} = 2\pi - \theta_{occ_i}$ and $x \in O$. $\quad (6)$

Table 1: Properties of the occluded angle distribution for some simple 2D curves. A reconstruction space Ω which encloses the convex hull of the objects is considered in these examples so that the distribution dependency on S may be ignored.

Inside any closed contour, θ_{occ} is 2π everywhere since the boundary, S, is occluded in all directions. Along the perimeter of any convex curve, θ_{occ} will be π, except when a discontinuity in curvature (i.e. a sharp corner) occurs, in which case θ_{occ} will be less than or equal to π at the discontinuity. For a smooth, arbitrary contour θ_{occ} will be $\geq \pi$.

In the case of a closed circle centred on the origin with radius R, θ_{occ} at a point outside the circle, $r \geq R$, can be described as a function of radial distance from the circle centre (Eq. 3). A discontinuity occurs between the interior of the circle, where θ_{occ} is 2π, and the boundary where θ_{occ} is π. If the circle now contains gaps such that only a fraction, p, of the curve is present, and the gaps are infinitely narrow and evenly distributed about the circumference, then in the limit, θ_{occ} "inside" the circle will form a uniform distribution with the value $2\pi p$ (Eq. 4). In this context "inside" means within the convex hull of the data points. The fraction of holes present in the circle is $1-p$, therefore outside the circle, $r \geq R$, θ_{occ} is similar to that in the closed case except that it is reduced by the coefficient $1-(1-p)(1-p) = (2p-p^2)$. The circle can be reconstructed by thresholding the θ_{occ} distribution. It can be shown that the range over which the threshold can be varied for a small reconstruction tolerance, $\delta \ll R$, decreases in an approximately squared relationship with decreasing p [18]. We might therefore expect that halving the number of points on a convex curve will lead to an approximate quartering of the range over which the θ_{occ} threshold may vary for a specified reconstruction tolerance.

We now consider the θ_{occ} profile along the axis of symmetry of a circle containing a single gap. The gap is defined by the angle which it subtends at the circle centre, ψ_{gap}. Along the circle perimeter θ_{occ} is π. It can be shown from the general expression for θ_{occ} inside the circle that constant values of θ_{occ} define a family of chords which pass through the gap end points [18]. Along the straight line between the gap end points θ_{occ} is π, independent of the gap width (this iso-surface corresponds to a chord with infinite radius). These observations are true for any convex curve containing a gap.

In the case of multiple discrete gaps in a circle the effect of each gap inside the circle can be considered separately. We define the complimentary (non-occluded) angle as $\overline{\theta_{occ}} = 2\pi - \theta_{occ}$. At each point inside the circle the occluded angle is the difference between 2π and the net non-occluded angle due to each individual gap (Eq. 6). Superposition can be used to determine θ_{occ} inside any convex curve, however, superposition is not generally applicable; applicability is guaranteed only inside the curve.

A superposition argument can also be used to consider the effect of noise data (data which are falsely identified as part of the object to be reconstructed). A noise element can be modelled by the smallest closed circle which circumscribes it. The value of θ_{occ} at a point in Ω may be increased due to the presence of a noise element by *at most* the amount $2\sin^{-1}(R/r)$, where r is the distance to the element and R is the radius of the circumscribing circle. A pseudo-superposition argument applies in the sense that points which are occluded from the noise element by intervening object data are not influenced by it.

5 A discrete algorithm for estimating $\theta_{occ}(x)$

A discrete 2D data space, represented as an array of pixels of finite size, is now considered. Following the notation introduced in Section 3, a subset of pixels forms the reconstruction space, Ω. $O(j)$ denotes the value of the j^{th} pixel in Ω, where j is an integer index. $O(j)$ is considered constant over the domain of each pixel. We want to calculate θ_{occ} at the centre of each pixel within Ω in order to form the discrete occluded angle distribution, $\theta_{occ}(j)$. In the case of binary object data,

$$O(j) = \begin{cases} 1 & \text{if } j \in \hat{\Omega}_{obj} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Here $\hat{\Omega}_{obj}$ is the subset of pixels identified as part of the object to be reconstructed.

We return briefly to the continuous domain in order to formulate an approximation for θ_{occ}. The term *projection* is introduced in this context to mean a line integral of the attribute $O(\mathbf{x})$ along a straight line path through Ω. Referring to Fig. 1, $\mathcal{P}(\mathbf{x}, \theta)$ denotes the projection along the ray emanating from the point $\mathbf{x}$ with orientation θ,

$$\mathcal{P}(\mathbf{x}, \theta) = \int_{\mathbf{x}}^{S(\mathbf{x},\theta)} O(u)du \tag{8}$$

where $S(\mathbf{x}, \theta)$ is the point on S where a ray initiated at $\mathbf{x}$ with orientation θ *first* leaves Ω. Straight line ray paths are assumed throughout this paper. The *binary* projection sample, $\mathcal{P}_{bin}(\mathbf{x}, \theta)$, is defined as,

$$\mathcal{P}_{bin}(\mathbf{x}, \theta) = \begin{cases} 0 & \text{if } \mathcal{P}(\mathbf{x}, \theta) = 0 \\ 1 & \text{otherwise} \end{cases} \tag{9}$$

$\mathcal{P}_{bin}(\mathbf{x}, \theta)$ is the silhouette of the object data at x in the viewing direction θ. Equation (1), which defines θ_{occ} at a point $\mathbf{x}$, may now be interpreted as the integral of binary projections (silhouettes) along all rays emanating from $\mathbf{x}$,

$$\theta_{occ}(\mathbf{x}) = \int_0^{2\pi} \mathcal{P}_{bin}(\mathbf{x}, \theta)d\theta \tag{10}$$

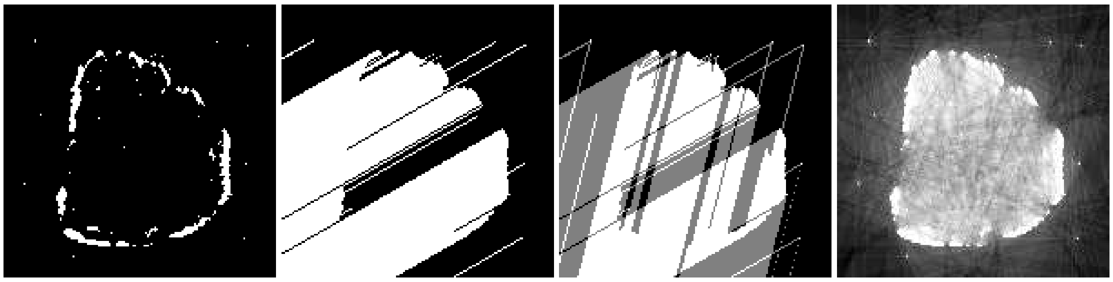
Equation (10) can be approximated by a finite sum of N projections taken at equally spaced orientations,

$$\theta_{occ}(\mathbf{x}) \approx \frac{2\pi}{N} \sum_{i=1}^{N} \mathcal{P}_{bin}(\mathbf{x}, \theta_i) \tag{11}$$

where θ_i is the angle specifying the orientation of the i^{th} ray, $\theta_i = i\frac{2\pi}{N}$. This equation says that the proportion of rays along which object data are "seen" to the total number of rays cast approximates the ratio of the occluded angle to 2π. In general, the larger N is, the better the approximation will be.

In the discrete space, a binary projection may be approximated with a pixel-based ray tracing algorithm [18]. The direct implementation of Eq. (11) suggests evaluating N projections at every pixel in Ω. However, the contribution to θ_{occ} at each pixel for a particular projection angle may be calculated simultaneously by casting a set of parallel rays through Ω, where the density of the rays cast is sufficient to guarantee that every pixel in Ω is considered. An *accumulator* array is formed where each cell corresponds to a pixel in Ω. The contributions from each projection angle at each pixel in Ω are summed in the accumulator. As each pixel along the path of a ray is considered, the content of the corresponding cell in the accumulator is set to one if object data has been encountered along the path of the ray, and set to zero otherwise.

This process is best illustrated by an example. Fig. 3(a) is a test data set of object



(a) (b) (c) (d)

Figure 3: Estimation of the discrete θ_{occ} distribution. (a) Object data, (b) a single binary projection, (c) summation of two projections, (d) summation of 25 projections.

pixels. The reconstruction space is the entire 150x150 pixel image. Fig. 3(b) depicts the contents of the corresponding accumulator array after a single set of parallel rays has been cast through Ω. The resulting image depicts the value of the corresponding binary projection evaluated at each pixel. The bright pixels represent pixels occluded by object data, i.e. the value of the projection is 1 at these pixels. As further sets of rays are traced through Ω over a range of orientations, the resulting binary projections for each orientation are summed in the accumulator array. The value in each accumulator cell indicates the number of projections occluded at the corresponding pixel in Ω. This is approximately proportional to θ_{occ}. Fig. 3(c) illustrates the summation of the projection in 3(b) with a second projection. The result is a tri-valued accumulator where pixels fall into three categories, those occluded by object data in both directions (bright pixels), those occluded by object data in only one of the two directions (mid-grey pixels), and those not occluded by object data in either direction (dark pixels). Fig. 3(d) is the accumulator corresponding to Fig. 3(a) after summing 25 projections taken at equally spaced angular intervals through Ω.

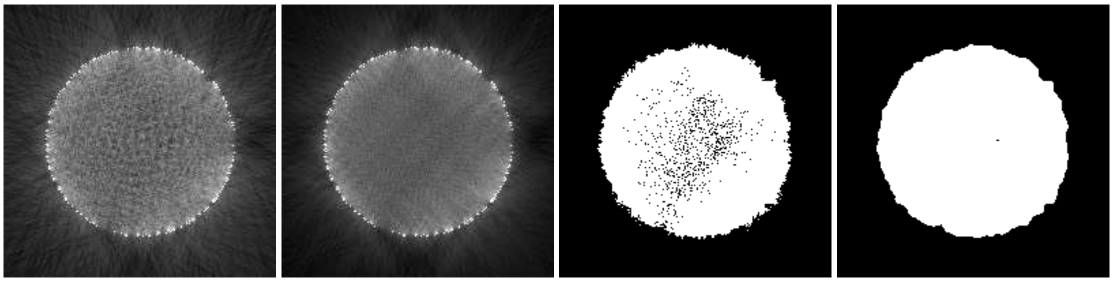
In this implementation, N projections are taken at equally spaced angular intervals of θ over the range $[0, 2\pi]$. Due to the finite number of projections, the occluded angle estimated at each pixel is discretised to integer increments of $\frac{2\pi}{N}$; this is the threshold resolution. The minimum number of projections required to reconstruct a given shape can be very small, but this number is shape dependent and prior knowledge of the shape and its orientation are required to take advantage of it.

In general Ω is a subset of pixels within a rectangular array. Only object data pixels within Ω are considered. Evaluation of the projection integral therefore commences when a ray enters Ω and halts when it leaves Ω. If a ray, when continued along its course, re-enters Ω then evaluation of the projection is recommenced afresh. This process is repeated until a ray leaves the domain of the image. Consequently, an arbitrary reconstruction space may be defined.

5.1 Thresholding

Having formed an estimate of the occluded angle distribution, it is possible to reconstruct simple objects such as those described in Table 1 by applying a threshold to the distribution.

Fig. 4 illustrates the reconstruction of an incomplete circle (case 5 in Table 1). The



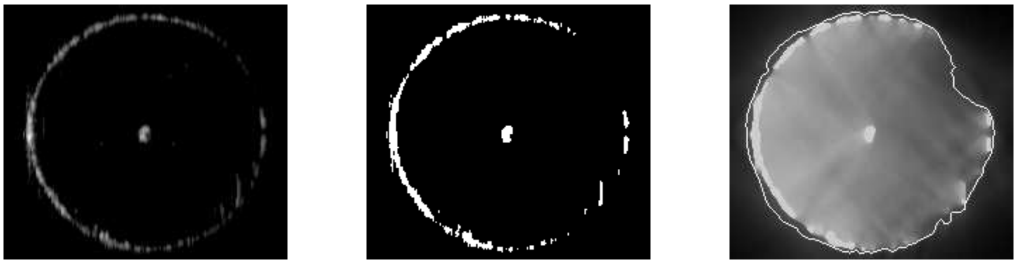
(a) (b) (c) (d)

Figure 4: Reconstruction of an incomplete circle with 25% of the perimeter present. (a) 50 projection accumulator, (b) 150 projection accumulator, (c) 150 projection accumulator thresholded for $\theta_{occ} \geq 0.5\pi$, (d) threshold applied after filtering with a 5x5 median filter.

original test data are a 200x200 pixel image of a circle with approximately three quarters of the circumference pixels removed, i.e. p, the proportion of the circle present, is 25%. Fig. 4(a) is the corresponding accumulator array after taking 50 binary projections through the data. In Fig. 4(b) the number of projections is trebled from 50 to 150; consequently the θ_{occ} distribution inside the circle is smoother due to the improved resolution ($2.4°$ *vs.* $7.2°$) and accuracy with which θ_{occ} is determined. The finite number of projections manifests itself as a crisscross or mottled pattern in the accumulator. The occluded angle distribution should be smooth and continuous throughout Ω except at object data, where discontinuities may arise.

The application of the threshold $\theta_{occ} \geq 0.5\pi$ to the 150 projection accumulator in Fig. 4(c) has resulted in holes. This is the critical value predicted by Eq. (4) in Table 1 which should reconstruct the circle in the ideal case. However, the discrete data approximately represents the theoretical case analysed in Table 1. The practical defects resulting from a finite number of projections and aliasing due to pixel-based ray tracing can be reduced by median filtering the accumulator. A median filter preserves low frequency transitions in the accumulator while removing high frequency speckle regions, the kind produced by a limited number of projections. Fig. 4(d) is the result of thresholding the 150 projection accumulator in Fig. 4(b) after applying a median filter with a 5x5 support. Filtering has smoothed out local variations in θ_{occ} while retaining the location of the circle boundary. Consequently, the threshold process has been able to reconstruct the interior of the circle uniformly.

Fig. 5(a) is a 2D ultrasound image of a rubber ball phantom suspended in jelly. The strong reflection in the middle of the scan comes from a wire support used to hold the ball in place. The stronger interface on the left side of the image indicates the direction of insonification. The raw scan data in Fig. 5(a) was thresholded to give Fig. 5(b), the required binary representation for the algorithm. The choice of this threshold is a compromise between a conservative one which ensures that weak interfaces likely to result from speckle are neglected, and a relaxed threshold which risks highlighting extraneous interface pixels. Fig. 5(c) is the corresponding filtered accumulator formed by taking 100 binary projections about 5(b); the contour $\theta_{occ} \geq 0.86\pi$ is superimposed upon the accumulator. This threshold choice resulted in the best reconstruction of the phantom. Although large gaps in the perimeter have been bridged, the threshold chosen does not follow the original data closely. This is expected since θ_{occ} is greater-than-or-equal to π along *any*



(a) (b) (c)

Figure 5: Reconstruction of an ultrasound phantom. (a) 2D ultrasound image, (b) thresholded ultrasound slice, (c) 100 projection accumulator with $\theta_{occ} \geq 0.86\pi$ contour shown.

smooth, opaque surface. A threshold less than π will therefore always dilate object data and will never construct a surface which coincides with the data. When gaps are large or non uniformly distributed, the application of a threshold to the θ_{occ} distribution is often inadequate to reconstruct the boundary of even a simple convex curve.

Two shortcomings of this initial approach become apparent from studying data from simple phantoms; the original binary classification of the ultrasound data is critical to the outcome and the application of a single threshold to the occluded angle distribution is rarely able to extrapolate across large gaps while closely adhering to interface data. These problems are addressed in the following two sections.

6 Continuous-Valued input data

A binary reconstruction space, where points are identified as either belonging or not belonging to the object under study, has been discussed. A reconstruction space where $O(\mathbf{x})$ is a continuous-valued distribution which reflects the likelihood that a point is part of the object to be reconstructed is now considered. This avoids a binary classification of the input data. Here rays cast through Ω are gradually attenuated by object data rather than simply being occluded. The data are treated as being translucent. The model for ray attenuation proposed is an exponential one based on monoenergetic X-ray attenuation. A ray undergoes exponential attenuation as it traverses the reconstruction space, where the linear attenuation coefficient is defined by $O(\mathbf{x})$. Referring to the two dimensional reconstruction space illustrated in Fig. 1, $\theta_{occ}(\mathbf{x}, \theta)$, the amount of attenuation experienced by a ray passing through Ω along the path defined by θ and $\mathbf{x}$, is given by,

$$\theta_{occ}(\mathbf{x}, \theta) = 1 - e^{-\int_{\mathbf{x}}^{S(\mathbf{x},\theta)} O(\mathbf{u})du} = 1 - e^{-\mathcal{P}(\mathbf{x},\theta)}, \tag{12}$$

where $S(\mathbf{x}, \theta)$ is the point on S where a ray initiated at $\mathbf{x}$ with orientation θ *first* leaves Ω. As before, the total amount of attenuation at $\mathbf{x}$, $\theta_{occ}(\mathbf{x})$, is found by integrating over all rays emanating from $\mathbf{x}$,

$$\theta_{occ}(\mathbf{x}) = \int_0^{2\pi} (1 - e^{-\mathcal{P}(\mathbf{x},\theta)})d\theta = 2\pi - \int_0^{2\pi} e^{-\mathcal{P}(\mathbf{x},\theta)} d\theta. \tag{13}$$

It is appropriate to refer to $\theta_{occ}(\mathbf{x})$ as an attenuation distribution. A consequence of the continuous projection model is that points originally specified as object data are not necessarily preserved by thresholding the attenuation distribution. In the case of binary projections, $\theta_{occ}(\mathbf{x})$ is 2π at any point in the object data.

6.1 Discrete implementation

Following the notation of Section 5, the discrete 2D attenuation distribution, $\theta_{occ}(j)$, is calculated at the j^{th} pixel by approximating the integral in Eq. (13) with a summation over a finite number of projections,

$$\theta_{occ}(j) \approx \frac{2\pi}{N} \sum_{i=1}^{N} (1 - e^{-\mathcal{P}(j,\theta_i)}) \tag{14}$$

where N is the number of projections taken and $\mathcal{P}(j,\theta_i)$ denotes the projection sample taken at the j^{th} pixel with the orientation θ_i. Because $\mathcal{P}(j,\theta_i)$ has a continuous value it is appropriate to calculate the contribution from each pixel according to the ray path length through the domain of the pixel. If $O(\mathbf{x})$ is considered constant over the domain of each pixel, then the contribution of the k^{th} pixel to the projection sample $\mathcal{P}(j,\theta_i)$ passing through it is $O(k)\delta_{i,j,k}$ where $\delta_{i,j,k}$ is the ray path length through the k^{th} pixel.

The direct implementation of Eq. (14) suggests taking N projections through Ω at each pixel. Although this guarantees that θ_{occ} is evaluated at pixel centres, it is computationally intensive. A significant reduction in computation can again be achieved if a single set of parallel projections is traced through Ω for each projection angle. As a ray is traced through Ω, the cumulative value of the line integral of the linear attenuation coefficient along the ray is recorded at each pixel boundary encountered. The amount of attenuation at the centre of the k^{th} pixel can be linearly interpolated from the projection rays either side of the pixel. For example, in Fig. 6 the value at the k^{th} pixel is interpolated from the projection samples $\mathcal{P}_{\eta,k_1}, \mathcal{P}_{\eta,k_2}$ and $\mathcal{P}_{\eta+1,k_3}, \mathcal{P}_{\eta+1,k_4}$. The attenuation at a pixel due to each projection angle is then summed in an accumulator array. In this way the computationally intensive task of ray-tracing is greatly reduced. Filtering is not required because aliasing artifacts from pixel-based ray tracing are avoided.

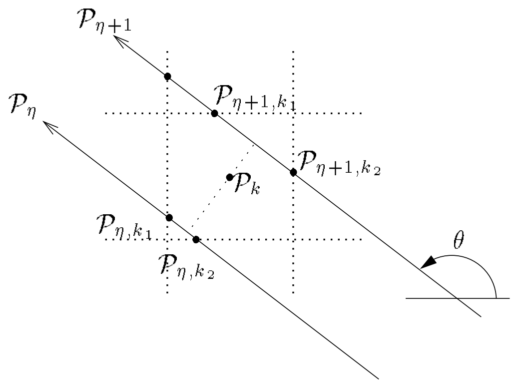


Figure 6: Bi-linear interpolation is used to estimate the attenuation at the k^{th} pixel, $\mathcal{P}_k$, for the projection angle θ.

7 Iterative algorithm

In Fig. 5 the choice of an appropriate threshold was critical to producing the final reconstructed shape. This choice depends on the object's shape and the object data. A high *conservative* threshold results in surfaces which closely adhere to object data. However, larger gaps are not bridged and separate regions tend to result. A lower, *relaxed* threshold is capable of bridging larger gaps in incomplete surface data. However, the resulting surface may no longer adhere closely to object boundary data and consequently the reconstructed surface will tend to be larger than the actual object, as demonstrated in Fig. 5(c). When the threshold is relaxed further, the surface fitted tends to circumscribe the object data and become more circular. We define a conservative threshold as one which is greater than or equal to π. This follows from the observation in Section 4 that θ_{occ} is π along any smooth section of a convex curve. A threshold less than π will define a contour larger than the object. In general, the application of a single threshold is capable of reconstructing only simple, mostly complete objects with uniformly distributed gaps and little noise.

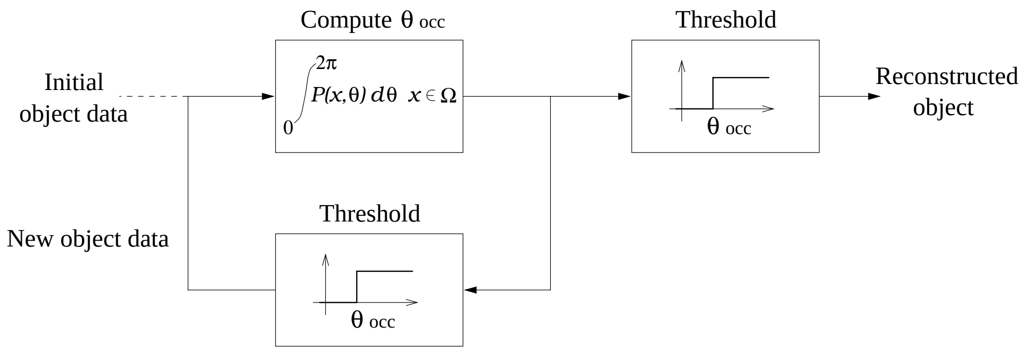
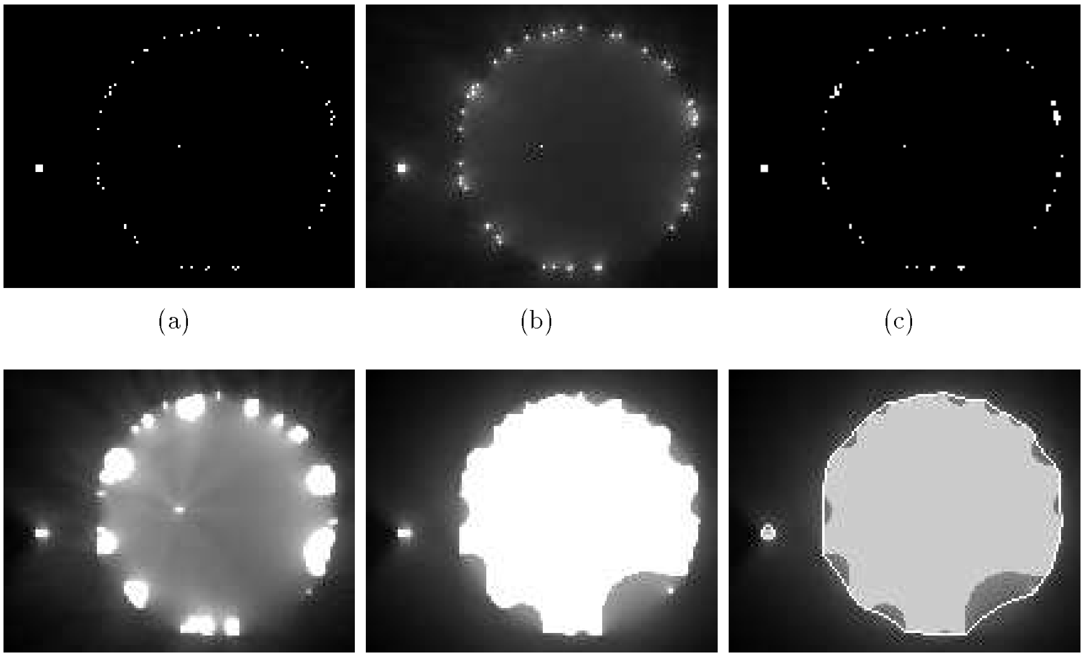


Figure 7: An iterative algorithm for object reconstruction.

An iterative algorithm (Fig. 7) is proposed to allow interpolation across large gaps while adhering closely to likely object boundaries. The attenuation distribution $\theta_{occ}(\mathbf{x})$ is formed as before and a conservative threshold is chosen which partially extrapolates the data. The regions extracted by thresholding are then employed as the object data for a second iteration of the algorithm. This process is repeated with the result that the object data grow to form a continuous region, interpolating across gaps, yet retaining accurate object boundaries.

Fig. 8 illustrates the iterative reconstruction of a circle from sparse binary edge data. The reconstruction space coincides with the domain of the image and contains a small noise region (Fig. 8(a)). The initial accumulator, after taking 100 projections about the object data, is displayed in 8(b). It is not possible by thresholding the accumulator after a single iteration to form a reconstruction which extrapolates across the gaps in the edge data and at the same time adheres closely to the perceived circle. Instead a conservative threshold, $\theta_{occ} \geq \pi$, is chosen to form the new object data (Fig. 8(c)) for the next iteration. The growth in the initial data is small, but this quickly increases with further iterations. Fig. 8(d) and Fig. 8(e) are accumulators for the fourth and sixth iterations. A contour corresponding to the threshold $\theta_{occ} \geq \pi$ is superimposed upon the sixth accumulator in 8(f); this contour interpolates the original data and represents a possible solution. In this example the conservative threshold was maintained at $\theta_{occ} \geq \pi$ between iterations;

new object data found upon thresholding were universally assigned a high attenuation coefficient.



(a) (b) (c) (d) (e) (f)

Figure 8: Iterative reconstruction of a circle. (a) Object data, (b) first iteration accumulator, (c) object data for 2nd iteration obtained by thresholding (b), (d) accumulator for 4th iteration (e) accumulator for 6th iteration, (f) 6th iteration accumulator with contour corresponding to $\theta_{occ} \geq \pi$ superimposed.

Fig. 9 illustrates the iterative reconstruction of an object from sparse data contaminated by noise. The initial object data, Fig. 9(a), is binary and the reconstruction space coincides with the domain of the image. Four iterations consisting of 100 projections are illustrated in Fig. 9(b-e). The thresholds applied at each iteration were $1.6\pi, 1.6\pi, 1.3\pi, 1.3\pi$, respectively. At each iteration the new object data derived by thresholding the attenuation distribution were assigned high attenuation coefficients. The binary image depicted in 9(f) is the result of thresholding Fig. 9(e) for $\theta_{occ} \geq 1.3\pi$.

In these examples a simple binary threshold, chosen interactively, has been applied at each iteration. The general requirement is to start with a high threshold and decrease it over subsequent iterations. Ideally, the initial threshold should not be so high that it eliminates genuine object data, or that it fails to augment the initial object data. High initial thresholds are chosen primarily to ensure that only the points most likely to be part of the object are added to the new derived set of data. If false object points are identified (points outside the actual object) then these may grow in subsequent iterations. In the case of a single iteration, the reconstructed object was determined by the threshold applied to the initial accumulator and was very sensitive to this choice. The threshold choice in the iterative scheme is not so critical. The process converges if a fixed threshold greater than π is maintained.

Key properties of the shape which the iterative scheme produces are determined by the final threshold applied. In the case of binary projections, if the final threshold applied is π,

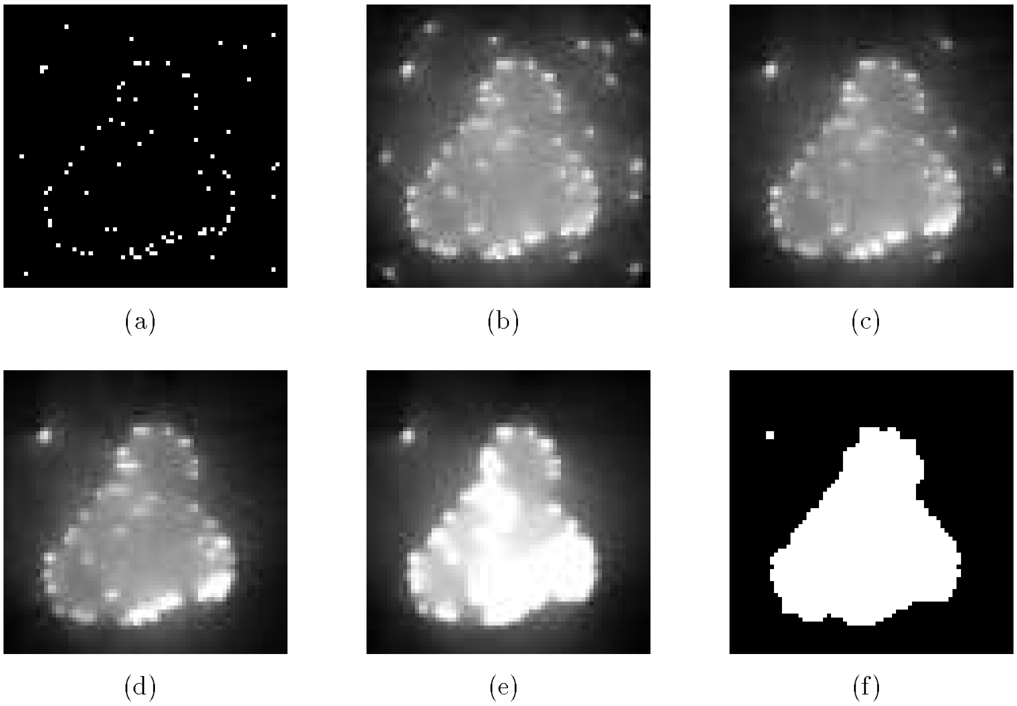


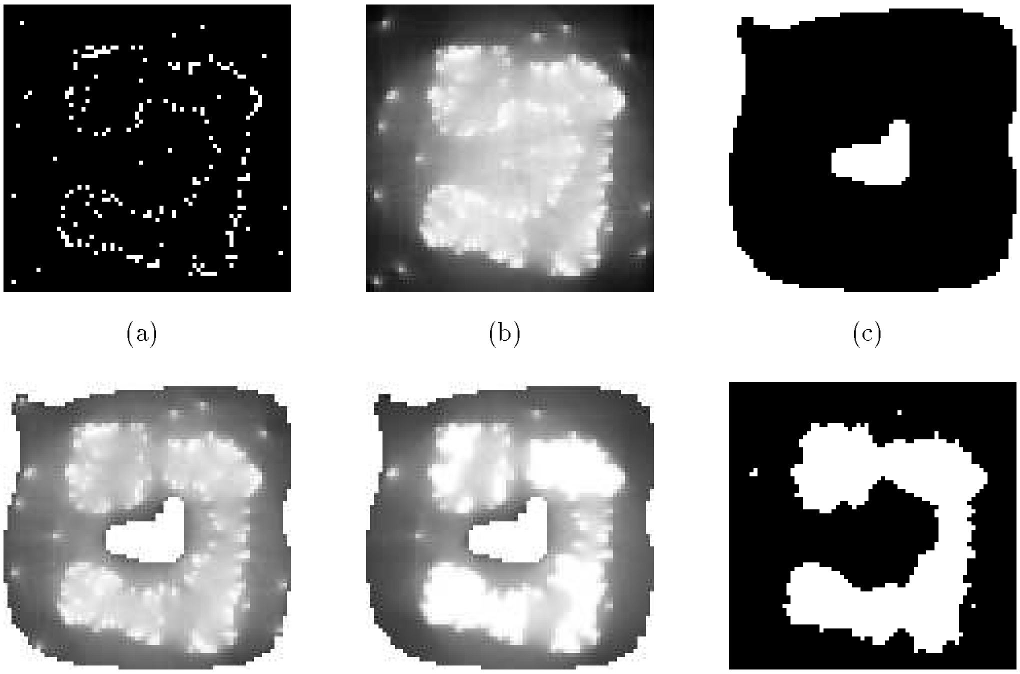
Figure 9: Iterative reconstruction of an object in the presence of noise. (a) Object data (b-e) 100 projection accumulator after 1st, 2nd, 3rd and 4th iterations (f) reconstructed shape obtained by thresholding (e) $\theta_{occ} \geq 1.3\pi$.

then the shape constructed will be completely convex. The shape constructed in Fig. 9(f) is not completely convex; here the final threshold choice of $\theta_{occ} \geq 1.3\pi$ has preserved the concavities. A lower choice would begin to fill these in. The final threshold determines the depth of concavities allowed in a shape's perimeter.

8 Reconstruction of cavities

The surfaces of an object which can be reconstructed from silhouettes are those which coincide with the object's visual hull. Laurentini shows that the visual hull for an object is uniquely defined for all viewing regions which enclose its convex hull [2]. When the viewing region enters the object's convex hull, the visual hull is dependent on the viewing region. This is illustrated in Fig. 2. The limitation which a global threshold places on shapes containing cavities can be overcome by taking advantage of the dependency of $\theta_{occ}(\mathbf{x})$ on the reconstruction (viewing) boundary S, when S lies within the convex hull of the object.

Fig. 10(a) illustrates the problem of reconstructing a shape with a significant cavity. If the domain of the image is employed as the reconstruction space, then the minimum threshold angle required to resolve the cavity is such that the object data will remain as a cluster of discrete regions. Gaps will not be bridged, but treated as cavities. The application of a global threshold cannot distinguish between a gap and an intended cavity of similar size in the perimeter of an object. Fig. 10(b) is the result of taking 100 projections where the reconstruction space coincides with the domain of the image. The bright pixels



(a) (b) (c) (d) (e) (f)

Figure 10: Iterative reconstruction of a cavity by defining a suitable reconstruction space. (a) Object data, (b) accumulator for 1st iteration where the reconstruction space coincides with the domain of the image (c) alternative reconstruction space, (d) accumulator for 1st iteration, (e) accumulator for 2nd iteration, (f) reconstructed shape after three iterations.

inside the cavity indicate high attenuation and hence these are seen as having a high probability of being part of the object. This is contrary to the desire to preserve the cavity. In Fig. 10(c) an alternative, 'doughnut' reconstruction space is defined. Fig. 10(d) is the result of taking 100 projections through this new reconstruction space; it can be contrasted with 10(b). Fig. 10(e) is the result of a second iteration of 100 projections through the data derived by thresholding Fig. 10(d) for $\theta_{occ} \geq 1.6\pi$. Fig. 10(f) is the result of thresholding the third and final iteration of the process such that $\theta_{occ} \geq 1.6\pi$. The definition of the new reconstruction space has allowed the cavity to be resolved, while still extrapolating across the gaps in the perimeter. Note that the only restriction on Ω is that it form a connected space; Ω may contain holes.

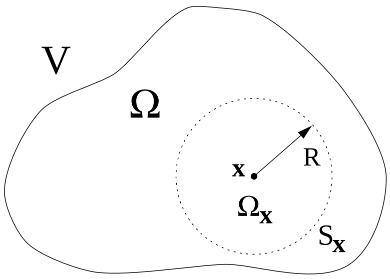


Figure 11: Definition of a circular reconstruction subspace centred on the point $\mathbf{x} \in V$.

User interaction can determine whether or not a cavity is resolved by supplying a suitable reconstruction space. However, this amounts to supplying an initial guess of the object's shape. Alternatively, by introducing a localised reconstruction space centred on each point $\mathbf{x}$, the ability to reconstruct cavities can be significantly improved without having to supply an initial estimate of the shape.

Consider the 2D case of a circular reconstruction space, Ω_x, centred on the point $\mathbf{x}$, where $\mathbf{x} \in V$ (Fig. 11). In 2D, with continuous-valued data, $\theta_{occ}(\mathbf{x})$ is now given by,

$$\theta_{occ}(\mathbf{x}) = 2\pi - \int_0^{2\pi} e^{-\int_0^R O(\mathbf{u},\theta)du} d\theta, \tag{15}$$

where R is the radius of the reconstruction space centred on $\mathbf{x}$. The amount of attenuation at each point $\mathbf{x} \in V$ is determined by the object data contained within the circle of radius R surrounding it. Clusters of points circumscribed by the reconstruction space will tend to form regions with successive iterations of the method. The size of R limits the distance over which two data points can be joined, and hence the size of regions formed. Thus, with a small reconstruction space, only data points in close proximity can be grouped together in a single iteration. As R increases, the influence of points further away is felt.

Localising the reconstruction space necessarily compromises the power of the original method to interpolate across large regions of sparse object data. However, in practice starting with a local reconstruction space and increasing R, together with the threshold applied at each iteration, produces better results than employing a reconstruction space that contains all the object data.

Implementation of the localised reconstruction space can be achieved using the discrete algorithm described in Section 6.1 with neglible additional computing overhead. A FIFO buffer is introduced to calculate the line integral of $O(\mathbf{x})$ along each projection ray. This stores the contribution to the projection integral from the last n pixels which a ray has passed through (Fig. 12). The size of this buffer, n, is a function of how many ray-pixel intersections occur over the distance R along the ray path. As the ray is traversed and the i^{th} pixel is encountered, the previous value of the projection, $\mathcal{P}_{i-1}$, is incremented by the contribution from the new pixel, $\Delta\mathcal{P}_i$, and decremented by the contribution from the oldest pixel in the buffer, $\Delta\mathcal{P}_{i-n-1}$. The new increment is then stored in the buffer in place of the oldest, and ray-tracing continues. As before, the attenuation at pixel centres is interpolated from the neighbouring projection samples. The additional computation required to realise the localised reconstruction space is therefore small and is independent of R.

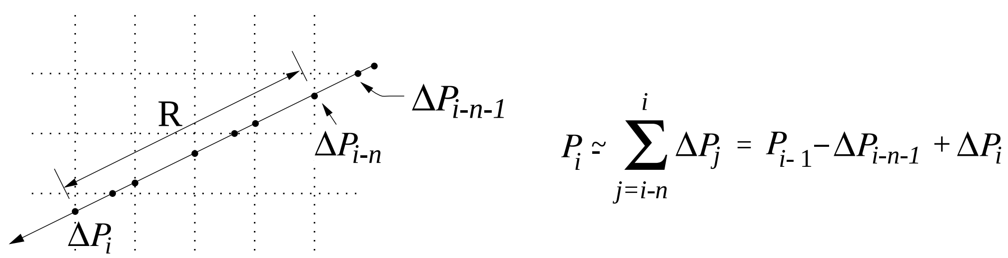
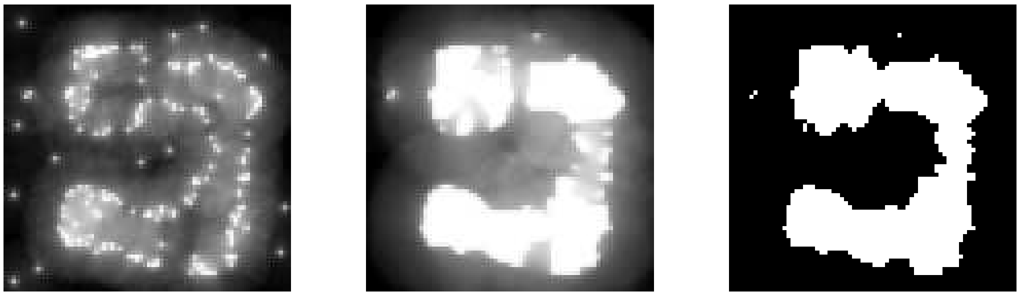


Figure 12: Computation of θ_{occ} along a ray on a 2D grid for a circular subspace with radius R.

In Fig. 13 the application of a local reconstruction space is illustrated with the data considered in Fig. 10. Fig. 13(a) is the accumulator after taking 100 projections. The attenuation at each pixel is calculated from object data within a 10 pixel radius. Fig. 13(b) is the accumulator after three iterations with a space 10 pixels in radius and a further iteration with a larger space 15 pixels in radius. A threshold of 1.4π was applied at each of the first three iterations. A more conservative threshold of 1.5π was then applied to the accumulator in Fig. 13(b). Fig. 13(c) is the result of taking a further iteration of projections with a space 15 pixels in radius and thresholding it for 1.5π. A reconstruction similar to that in Fig. 10(f) has been achieved without the definition of a shape-specific reconstruction space.



(a) (b) (c)

Figure 13: Iterative reconstruction of a cavity by defining a local reconstruction space. (a) Accumulator after first iteration with a circular space $R = 10$ pixels, (b) accumulator after two further iterations and a fourth with $R = 15$ pixels, (c) thresholded accumulator after a fifth iteration with $R = 15$ pixels.

9 3D shape reconstruction

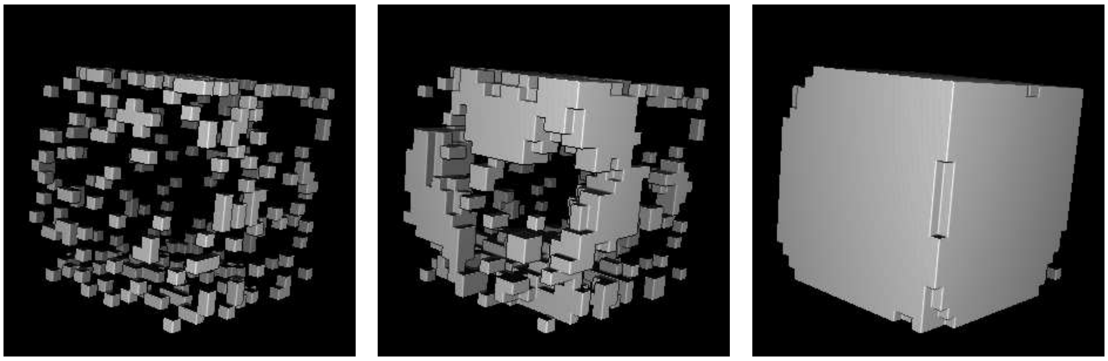
In three dimensions the reconstruction space becomes a volume and θ_{occ} is interpreted as the occluded *solid angle*. θ_{occ} is measured in steradians, hence $\theta_{occ} \in [0, 4\pi]$. The occluded solid angle at the point $\mathbf{x}$, $\theta_{occ}(\mathbf{x})$, is given by,

$$\theta_{occ}(\mathbf{x}) = 4\pi - \int_0^{2\pi} \int_{-\pi/2}^{\pi/2} e^{-\mathcal{P}(\mathbf{x},\theta,\phi)} \cos(\phi) d\phi d\theta \tag{16}$$

where $\mathcal{P}(\mathbf{x}, \theta, \phi)$ is the projection of $O(\mathbf{x})$ along the ray passing through Ω defined by the point $\mathbf{x}$ and the two angles, $\theta \in [0, 2\pi]$ (azimuth) and $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ (elevation).

Estimation of the discrete $\theta_{occ}(\mathbf{x})$ distribution in 3D is analogous to that in 2D. The accumulator becomes a 3D array of voxels. Planes of rays are traced through the reconstruction space and tri-linear interpolation between the four neighbouring projection rays is used to estimate θ_{occ} at each voxel centre.

In 2D, it is possible to take projections at equally spaced orientation angles for any specified number of projections. In 3D, orientations are chosen by projecting a uniform random distribution on to the solid angle space defined by θ and ϕ. A cylindrical equal area projection is used to do this. An approximately even coverage of orientations is then achieved for sufficiently large N.



(a) (b) (c)

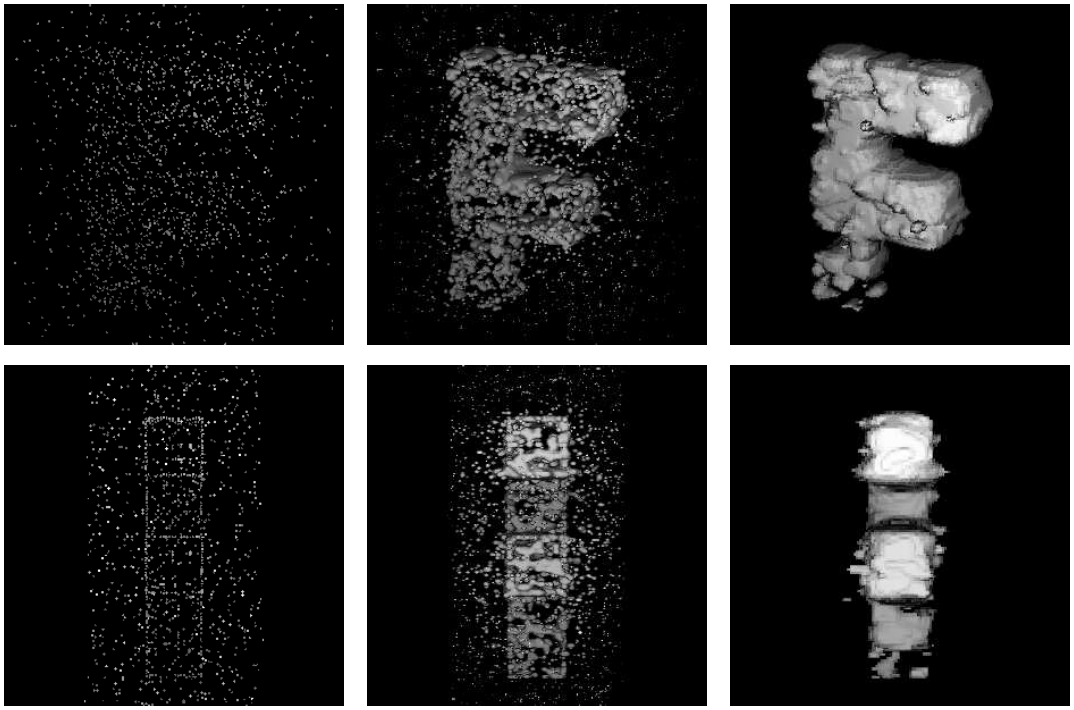
Figure 14: Iterative reconstruction of a cube. (a) Object data consisting of a cube with 90% of its surface voxels missing, (b) reconstructed object data after two iterations, (c) reconstructed object data after four iterations.

9.1 3D reconstructions

Figure 14 demonstrates the iterative reconstruction of a cube from an incomplete set of surface voxels. In Fig. 14(a) 90% of the surface voxels for a complete (20x20x20) cube have been removed. The object data form a binary distribution and have been assigned a high attenuation coefficient rendering them opaque. Fig. 14(b) is the reconstructed object data after two iterations of projections with a conservative threshold choice ($\Theta_{occ} \geq 2.5\pi$ steradians). The surface voxels have formed small clusters. Fig. 14(c) is the reconstructed object data after four iterations. Voxels occluded in at least half the projections taken are shown ($\Theta_{occ} \geq 2\pi$ steradians). 98.8% of the cube has been reconstructed with 0 extraneous voxels. The sharp edges and flat faces of the cube have been reconstructed. The $\Theta_{occ} \geq 2\pi$ threshold defines a convex surface in 3D, with respect to S. A lower choice would dilate the cube.

Figure 15 illustrates the reconstruction of a more complex object from noisy data. 95% of the surface voxels of the test object were removed and noise was added with a uniform random spatial distribution. The data set is binary; noise and object voxels carry equal weighting. Four iterations of projections were performed. In the final iteration a localised spherical reconstruction space was used to remove extraneous noise regions surrounding the main object. The threshold applied after the first iteration (1.2π) was increased to 2.2π and maintained for the following iterations. Fig. 15(a) is the raw data. Fig. 15(b) is the data resulting from thresholding the first iteration. Fig. 15(c) is the final iteration with the surface corresponding to the threshold 2.2π depicted. Without prior knowledge and recognition of the object, or any of its subparts, there is no reason to suppose that the object consists of flat surfaces nor that a surface should be fitted across the apparent gaps in the trunk of the 'F'. Variation of the threshold applied between iterations might produce an improved reconstruction, however, we have yet to determine a mechanism for making such a choice. Figure 16 illustrates the effect of different threshold choices on the second iteration of the reconstruction. A range of thresholds between 2π and 4π have similar effect. In this example, the final reconstructed object is not very sensitive to the threshold choice. This observation tends to be true of many other examples studied.

By considering the ideal case of opaque surfaces containing no missing data we can

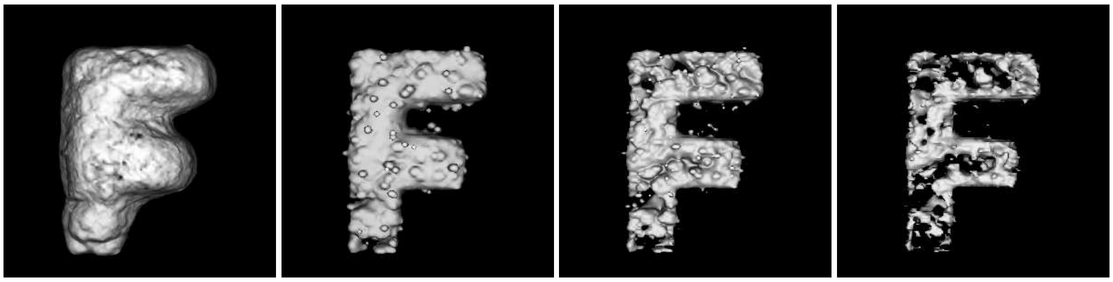


(a) (b) (c)

Figure 15: Iterative reconstruction of an object in the presence of noise. (a) Object data (b) reconstructed object data after the first iteration (c) reconstructed object data after four iterations

define limits on what shapes are reconstructible for a given threshold. The scale of reconstructible surface features is determined by the size of the reconstruction space. Consider a local subspace of radius R and the points on the opaque surfaces depicted in Fig. 17. If the surfaces are opaque and contain no holes then the visible angle at the point $\mathbf{x}$ in each case is 2π, π and $\frac{\pi}{2}$, respectively. This means that the occlusion thresholds which allow these surface points to be reconstructed are those greater than 2π, 3π and $\frac{7}{2}\pi$, respectively. For example, in Fig. 17(a) a threshold T less than 2π will construct a surface which lies a distance $R\cos\frac{T}{4}$ above the plane. A threshold less than 3π in Fig. 17(b) will construct a surface which fills-in the concave edge with a beading, the radius of which is a function of R and T. If a threshold close to 0 is chosen then the object data are uniformly dilated. In mathematical morphology this is equivalent to a dilation with a spherical structuring element of radius R [23]. A threshold of 4π is equivalent to an erosion with a sphere of radius R. For an object containing all three types of surface facets shown in Fig. 17, the threshold required to accurately reconstruct the shape is dictated by that of the most concave feature, in this case the corner shown in Fig. 17(c). A threshold between 3π and $\frac{7}{2}\pi$ will not accurately reconstruct such corners but may reconstruct planar and edge surfaces depicted in Figs. 17(a) and 17(b).

In general, the final threshold applied determines the cavities permitted within an object, irrespective of the data. The use of a localised reconstruction space means that the occluded solid angle is determined over a sub-region of the object's surface. The cavity constraint implied by the threshold is still valid, but only at a scale determined by the size



(a) (b) (c) (c)

Figure 16: The effect of varying the solid angle threshold (a) π (b) 2π (c) 3π (d) 3.8π.

of the local reconstruction space. If the object data are translucent (i.e. when continuous projections are used to compute $\theta_{occ}(\mathbf{x})$) then it becomes difficult to make such general statements since the amount of occlusion (attenuation) at a point depends on the thickness as well as the spatial distribution of the surrounding object data.

Figure 15 represents a particularly difficult example of a 3D reconstruction because of the binary nature of the input data. In many real-world data sources true object data and false (noise) data carry different weightings. For example, in the case of ultrasound data, false interfaces tend to be characterised by weaker reflections while genuine interfaces are much stronger. This information can significantly assist object reconstruction.

Figure 18 illustrates the reconstruction of an ultrasound phantom from a set of spatially registered 2D ultrasound scans [18]. Fig. 18(a) is the raw ultrasound data projected onto a regular voxel grid of 1mm^3 voxels. 2D ultrasound slice data were taken from arbitrary positions and orientations around the phantom and were mapped to the nearest voxel with multiple contributions to the same voxel being averaged. The majority of voxels contained no data due to the incomplete coverage of the phantom by the set of scans. This is the input data to the algorithm. Fig. 18(b) is the corresponding 3D accumulator after two iterations of projections. At each iteration 1000 projections were taken with rays spaced 1mm apart within each projection. The reconstruction space coincided with the domain of the voxel array; a locally defined reconstruction space was not used to reconstruct this object. The raw data were acquired with 8-bit resolution and an attenuation coefficient assigned which varied linearly with the ultrasound number over the range 0.0-5.0. Consequently, strong reflections were effectively treated as opaque and weaker reflections as translucent. The first iteration was thresholded for $\theta_{occ} \geq \frac{5}{2}\pi$ steradians. The resulting object data was then

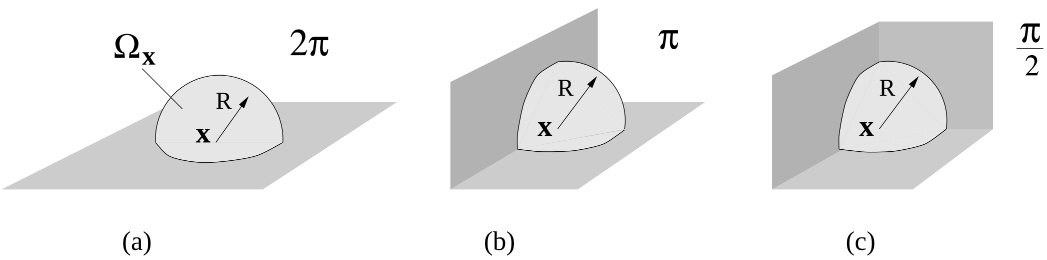


Figure 17: The local visible surface angle for three configurations of opaque planar surfaces.

(a) (b) (c)

Figure 18: Reconstruction from 3D ultrasound data. (a) 3D ultrasound data for a phantom, (b) 3D accumulator after two iterations of 1000 projections. The surface shown corresponds to a threshold of $\theta_{occ} \geq 2\pi$ steradians. (c) Rendering of phantom from CT data.

assigned a high attenuation coefficient, resulting in a binary data set. The accumulator formed by taking projections about this data was thresholded for $\theta_{occ} \geq 2\pi$ steradians. This is the surface displayed in Fig. 18(b). For comparison, the phantom was CT scanned with parallel slices taken at constant spacing so that a regular voxel array of data was acquired. The resolution was twice the ultrasound resolution within each slice. A view similar to Fig. 18(b) was produced in Fig. 18(c) via a standard ray-tracing algorithm. The main differences are due to incomplete coverage of the object by the ultrasound probe during scanning.

10 Conclusion

In this paper we have demonstrated a new application for shape-from-silhouette research: the reconstruction of object surfaces from scattered, unorganised data. The problem differs in that the 3D object data, albeit incomplete, are specified rather than being unknown. Projections taken from multiple viewpoints through the data have been used to reconstruct an incomplete object just as a human observer might do when inspecting a data set with a volume visualisation package. People have the ability to discern structure from semi-transparent visualisations of volume data when the data are viewed from multiple viewpoints in an animated sequence, even though the individual views may be ambiguous. The approach to object modelling proposed in this paper can be viewed as an attempt to model this apparent perceptual skill. The method used to cluster points in the object data is based on determining their visibility (occlusion). We defined the occluded angle distribution as the projection of the object data within the subspace Ω on to each point in Ω. We generalised from binary projections (silhouettes) to continuous-valued projections. This allowed points originally specified as object data to be removed by the thresholding process. An iterative algorithm was developed which enabled interpolation across regions of sparse and dense data alike. Although it is difficult to characterise this algorithm, due to the dependency on the initial object data, some general properties can be discerned in the case of binary projections. The most important is that the threshold applied at each iteration determines the degree of self-occlusion allowable in the reconstructed surface.

Reconstructing concavities is an inherent difficulty with volume intersection techniques. The visual hull determines how much can be understood about an object from its silhouettes. It becomes dependent on the viewing region when the viewing region enters the convex hull of the object. In Section 8 this dependency was used to reconstruct concave object surfaces. This solution requires some knowledge of the object to be reconstructed when specifying Ω. A more satisfying approach is the use of a localised reconstruction space $\Omega(\mathbf{x})$. This gives the method a morphological flavour whereby $\Omega(\mathbf{x})$ is analogous to a morphological structuring element. However, unlike traditional morphological closing filters [23] there is no explicit erosion operation and the dilation is a function of the local data within Ω. The amount of dilation varies throughout Ω, it is not uniform except in the case of an extreme threshold choice.

Throughout this paper we have considered projections taken at equi-angular spacing in order to compute the attenuation or occlusion at each point in Ω. Depending on the data distribution and the object, this is not strictly necessary. Some active vision systems choose projection directions based on the data [15, 14]. Such techniques may be applicable in our problem and may reduce computation.

Any solution to the ill-posed problem of constructing surfaces from sparse, unorganised data in the absence of *a priori* information is a compromise between opposing goals. Distinguishing between a gap to be bridged and a cavity to be preserved is arbitrary in the absence of further information. In the method described, user interaction has been used to resolve ambiguities in how a surface may be fitted to the data through the specification of a threshold and the domain of Ω. These parameters are the subject of further investigation. The approach presented here is inherently biased towards filling-in cavities when they are not reconstructible. Similarly, a poor choice of thresholds results in cavities being filled-in. However, we have shown that this novel approach can produce interesting results. Furthermore, in the presence of noise, the ability to robustly fit a simpler surface is likely to be useful in many applications.

References

[1] A. Laurentini. "How far 3D shapes can be understood from 2D silhouettes," *IEEE Transactions on PAMI*, 17(2):188–195, February 1995.

[2] A. Laurentini. "Visual hull concept for silhouette-based image understanding," *IEEE Transactions on PAMI*, 16(2):150–162, February 1994.

[3] M. Jones and J. Oakley. "New shape representation for fast 3D reconstruction from multiple 2D perspective images," In *IEE Computing and Control Division Colloquium on New Developments in 3D Image Capture and Application*, pages 4/1–4/3. IEE, Stevenage, England, May 1995.

[4] J. Zheng. "Acquiring 3D models from sequences of contours," *IEEE Transactions on PAMI*, 16(2):163–177, February 1994.

[5] N. Ahuja and J. Veenstra. "Generating octrees from object silhouettes in orthographic views," *IEEE Transactions on PAMI*, 11:137–149, February 1989.

[6] H. Noborio, S. Fukuda, and S. Arimoto. "Construction of the octree approximating three-dimensional objects by using multiple views," *IEEE Transactions on PAMI*, 10(6):769–781, November 1988.

[7] S. Senbel and M. Ismail. "Octree generation from object silhouettes in orthographic views," In *Visual Data Exploration and Analysis III*, volume 2656, pages 379–389. Proceedings of SPIE, February 1996.

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

[9] X. Y. Cheng, A. Ohya, M. Natori, and M. Nakajima. "Boundary extraction method for three dimensional ultrasonic echo imaging using fuzzy reasoning and relaxation techniques," In *Proceedings of the Nuclear Sciences Symposium and Medical Imaging Conference, San Francisco*, 1993.

[10] J. Michel and N. Nandhakumar. "Unified 3D models for multisensor image synthesis," *Graphical Models and Image Processing*, 57(4):283–302, July 1995.

[11] S. Lavallee and R. Szeliski. "Recovering the position and orientation of free-form objects from image contours using 3D distance maps," *IEEE Transactions on PAMI*, 17(4):378–390, April 1995.

[12] R. Murch. *Inverse Scattering and Shape Reconstruction.* PhD thesis, Dept. Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, April 1990.

[13] C. Chien and J. Aggarwal. "Model construction and shape recognition from occluding contours," *IEEE Transactions on PAMI*, 11(4):372–389, April 1989.

[14] K. Yoshida, H. Tanaka, J. Ohya, and F. Kishino. "Active 3D modelling by recursive viewpoint selection based on symmetry," *Proceedings of SPIE*, volume 2588, pages 326–336, October 1995.

[15] X. Yuan. "A mechanism of automatic 3D object modelling," *IEEE Transactions on PAMI*, 17(3):307–311, March 1995.

[16] W. E. Snyder, R. Groshong, M. Hsiao, K. L. Boone, and T. Hudacko. "Closing gaps in edges and surfaces," *Image and Vision Computing*, 10(8):523–531, October 1992.

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

[18] J. C. Carr. *Surface Reconstruction In 3D Medical Imaging.* PhD thesis, Dept. Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, February 1996.

[19] M. Belohlavek, J. F. Greenleaf, D. A. Foley, and J. B. Seward. "Utility of image enhancement methods in three-dimensional ultrasound reconstruction," In *IEEE 1991 Ultrasonics Symposium Proceedings*, volume 2, pages 1219–22. New York: 1991.

[20] T. McInerney and D. Terzopoulos. "Deformable models in medical image analysis: A survey," *Medical Image Analysis*, 1(2):91–108, June 1996.

[21] N. S. Raja and A. K. Jain. "Recognizing geons from superquadrics fitted to range data," *Image and Vision Computing*, 10(3):179–190, April 1992.

[22] R. O. Duda and P. E. Hart. *Pattern classification and scene analysis.* Wiley, 1973.

[23] R. C. Gonzalez and R. E. R. Woods. *Digital image processing.* Addison-Wesley, 1992.