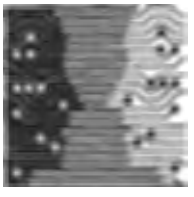
PERCEPTUAL DEPTH CUES IN SUPPORT OF MEDICAL DATA VISUALISATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE,
FACULTY OF SCIENCE
AT THE UNIVERSITY OF CAPE TOWN
IN FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Caleb A. Lyness
June 2004

Supervised by
Edwin Blake &
Patrick Marais



Abstract

This work investigates methods to provide clinically useful visualisations of the data produced by an X-ray/CT scanner. Specifically, it examines the use of perceptual depth cues (PDCs) and perceptual depth cue theory to create effective visualisations.

Two visualisation systems are explored: one to display X-ray data and the other to display volumetric data. The systems are enhanced using stereoscopic and motion PDCs. The presented analyses show that these are the only possible enhancements common to both systems. The theoretical and practical aspects of implementing these enhancements are presented.

Volume rendering techniques are explored to find an approach which gracefully handles poorly sampled data and provides the interactive rendering needed for motion cues. A low cost real time volume rendering system is developed and a novel stereo volume rendering technique is presented. The developed system uses commodity graphics hardware and Open-GL.

To evaluate the visualisation systems a task-based user test is designed and implemented. The test requires the subjects to be observed while they complete a 3D diagnostic task using each system. The speed and accuracy with which the task is performed are used as metrics. The experimental results are used to compare the effectiveness of the augmented perceptual depth cues and to cross-compare the systems.

The experiments show that the user performance in the visualisation systems are statistically equivalent. This suggests that the enhanced X-ray visualisation can be used in place of CT data for some tasks. The benefits of this are two fold: a decrease in the patient's exposure to radiation and a reduction in the data acquisition time.

Acknowledgments

I would like to thank African Medical Imaging (AMI) and LODOX for sponsoring this research. My thanks to my supervisors Edwin Blake and Patrick Marais. I would like to extend my gratitude to Patrick, Dave Nunez and my sister, Karynsa, for having the patience to completely read my thesis, providing suggestions and corrections. My thanks also goes to Mattieu de Villiers for providing the LODOX LACT and SACT data, the associated raw X-ray data, and the machine specifications. I would like to thank Klaus Engel from the University of Stuttgart for his assistance regarding the implementation and my colleagues in the CVC lab for thoughtful, exciting and sometimes somewhat distracting ideas and discussions. A special note to Otto-Carl Marte, as a fellow LODOX project student, who had similar experiences and endured countless progress meetings.

Finally I would like to thank my parents, Malcolm and Gill, and my girlfriend, Maria Bösing, for providing me with moral and emotional support during the lows, when I had no more to give. Without them I would never have completed this dissertation. Its been a long, winding road. Cheers.

Contents

List of Tables

List of Listings

List of Figures

Chapter 1

Introduction

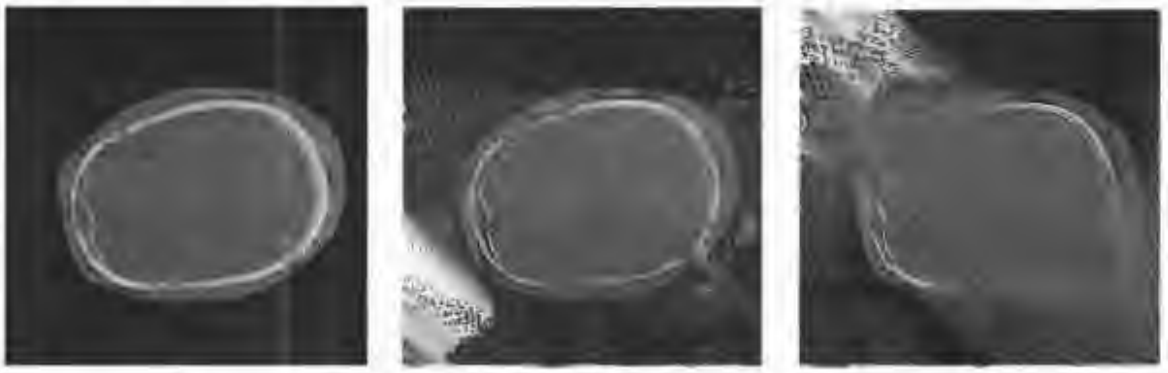
Medical imaging is growing in importance. Non-invasive diagnostic techniques provide clinicians with insight into a patient's pathology. Accurate and rapid diagnoses are made possible in this way. Interactive digital medical imaging places the clinician in control and it exposes a wider range of data.

Unfortunately, non-invasive techniques expose the subject to harmful radiation. Techniques to reduce the radiation dosage are constantly being sought. Limited angle and sparse angle CT are two such techniques.

This dissertation explores the use of perceptual depth cues and perceptual depth cue theory to create effective visualisation of X-ray/CT data. It is hypothesised that effective visualisation can be used to reduce the impact of limited and sparse angle CT reconstruction artifacts. In addition, it is believed that X-rays can be used for diagnostic tasks for which 3D reconstructions have been traditionally reserved. To test these assertions, several prototypes will be developed and studies performed to test the effectiveness of the perceptual depth cue enhanced systems.

1.1 Computed tomography

Computed Tomography (CT) describes the process of reconstructing an object's cross section from projections of the internal structures taken at various angles. The projections are created through various techniques, for example, the transmission of X-rays or charged particle deflections[47, 23]. Several modalities are commonly used, in particular, Magnetic Resonance Imaging (MRI), X-ray Computed Tomography (X-ray/CT), Positron Emission Tomography (PET) and Photon Emission Computed Tomography (PECT). Each of the modalities reveals different properties of the imaged object. Through these techniques the



(a) (b) (c)

Figure 1. Single cross sections from CT reconstructions. a) A full reconstruction provides detailed information about the internal density structures of the subject. b) A sparse angle CT reconstruction results in a cross section with streak artifacts. c) A limited angle CT reconstruction results in a cross section with missing edge information.

[nternal structure and nature of objects can be studied in a non invasive manner.

Traditional X-ray/CT requires a large number of X-ray projections over a full angular range (180°). While the required high sampling rates produce high resolution CT reconstructions, capturing and processing of the projection data is time and resource consuming. Additionally, for each projection, the subject is exposed to ionising radiation. For live subjects, minimising the harmful radiation dosage is of great importance. Limited angle CT (LACT) and sparse angle CT (SACT) alleviate these problems by requiring fewer projections for a reconstruction.

SACT uses larger angles between the projection positions. Current reconstruction techniques applied to sparse angle data sets produce images containing streak artifacts — see Figure 1b. LACT, on the other hand, uses a limited angular range, for example 90°. The limited angular range tends to be the result of either physical restrictions placed on the hardware or signal quality concerns within the range of exposed angles. Reconstructions from limited angle data sets result in distorted images with edge information missing in the regions corresponding to the angular range used — see Figure 1c [23]. While current work [24] is doing much to improve the reconstruction and reduce the artifacts for both LACT and SACT, the artifacts still remain a problem.

1.2 X-ray/CT visualisation

It is believed that effective visualisation may be an alternative solution to the reconstruction artifact problem. Rather than trying to remove artifacts, the available data may be presented to the viewer "as is". The visualisation must provide the viewer with the insight required to use the data to its full potential, while ensuring that he/she is aware of the reconstruction artifacts. In X-ray/CT there are two forms of data: digital X-rays and volumetric data. The X-ray data is the "raw" form, coming directly from the machine. The volumetric data is built by stacking several CT cross sections which are reconstructed from the X-rays.

Thus, a pertinent question is: "How does one *effectively* visualise the data?" In many respects, scientific visualisation and more generally computer graphics is about "tricking the brain" into perceiving a three dimensional environment. Relatively simple techniques are often applied to enhance the "realism" of an image. Enhancements may alter the viewers perception and thus the view of the data in a visualisation. While the ability to apply such enhancements to a visualisation is valuable, there is a possibility of introducing false perceptions which could compound errors and artifacts further. This possibility is a genuine concern in medical visualisation as a diagnosis is based on what is displayed and false diagnoses may have dire consequences.

The human visual system (HVS) is a powerful analysis tool if properly assisted. Therefore, the previous question can be rephrased as: "How does one assist the HVS in a visualisation?" *Perceptual depth cue theory* provides a possible solution. Perceptual depth cue theory attempts to explain how humans perceive depth. Depth perception is a crucial stage in analysing and understanding images. A perceptual depth cue enhanced visualisation attempts to maximise the use of the large portions of the brain involved in the subconscious processing of natural visual stimuli [137, 6], while minimising the introduction of visual artifacts. A large amount of visual processing occurs preconceptually in specialised brain centres, so we can effectively present more detailed information without much impact on the cognition required.

Examining the scientific and medical visualisation literature reveals the ubiquitous use of perceptual deph cues (PDCs), at least implicitly. It is important to examine visualisation from a PDC theory perspective to learn what enhancements are possible and determine which PDCs are the most cost-effective. This last point is particularly salient as PDCs are frequently technically challenging and expensive to implement.

1.3 Research aims and methodology

This work aims to discover effective visualisations of the data available in X-ray/CT using perceptual depth cues. To this end, the development of two prototype systems will be investigated. One system to visualise the raw data (X-ray images) and the other to visualise the processed data (volumetric data).

Various volume visualisation techniques will be explored in order to determine which is the most effective at displaying LACT and SACT data. The best method will be analysed to determine the "natural" depth cues and which cues can be used to enhance the visualisation. A similar analysis will be applied to X-ray images. The visualisation systems will then be enhanced using selected PDCs. The same PDCs will be used to enhance both the X-ray and volume visualisation systems.

The enhanced visualisations will be tested using a "diagnostic" user task to determine whether the depth cues and their combinations are effective. In addition, we wish to show that the enhanced X-ray visualisation is comparable to the volume visualisation with respect to the three dimensional diagnostic task.

Two prototype systems will be developed for use on low end commodity hardware. It is believed that desktop computers will be the platform for future diagnostic software. The ever increasing power of low end consumer hardware has made this possible. The data will be acquired using a prototype low dosage, digital X-ray scanner (LODOX), which has successfully been used to produce LACT and SACT reconstructions [117, 118, 120, 23]. The software which will be developed is destined to be incorporated into this system.

The prototypes will act as a proof-of-concept and include the functionality necessary to evaluate the effectiveness of the PDC enhancements. The enhancements will be measured using task based evaluations [60, 99]. The task will involve locating feature points, positioning and marking these points with indicators. The accuracy and time taken to complete the task will be used as a measure of the effectiveness. In addition, during the experiments the users will be observed to determine the manner in which the task is performed.

The results will be used to compare each of the PDC enhancements and their combinations. The PDC enhancements will be ranked and the results compared for each system. This will allow for the determination of whether the enhanced X-ray visualisation is comparable to the volume visualisation for a 3D diagnostic task. Furthermore, the ranking will be used to evaluate the PDC implementation in each of the visualisations.

1.4 Outline of this dissertation

Chapter 2 – Volume visualisation: In Chapter 2 volume visualisation techniques are introduced. Both surface-based and voxel-based rendering techniques and concepts are presented. The emphasis lies on the voxel-based techniques.

Chapter 3 – Perceptual depth cues and visualisation: In this chapter PDC theory is described. Voxel-based volume visualisation and X-ray visualisation with respect to PDCs are discussed, as well. This discussion includes the analysis of PDCs found in typical images presented in each system.

Chapter 4 – Stereoscopic depth: In Chapter 4 the concepts of stereopsis, stereogram composition techniques, presentation methods and software frameworks for stereoscopic systems are discussed.

Chapter 5 – Prototype systems: In Chapter 5 the visualisation prototypes and the techniques developed are described in detail. The chapter is introduced with a discussion of the work leading up to the final prototypes. This includes a mention of the real time isosurface browsing system developed, the comparison of volume visualisation techniques with respect to LACT and SACT data and the developed stereoscopic test application. A novel approach to stereo volume rendering is presented.

Chapter 6 – Effectiveness study: In Chapter 6 the experiments used for testing the effectiveness of PDC enhanced visualisations is presented, along with the results and findings.

Chapter 7 – Conclusion: Chapter 7 concludes the dissertation with a summary of interesting outcomes and discusses the avenues for future work.

Chapter 2

Volume visualisation

Volume visualisation techniques have been investigated for nearly two decades. The rendering algorithms have evolved from those producing coarse block-like images to those producing near-photo realistic images in real time.

In this chapter we present an overview of the volume visualisation techniques. We begin in Section 2.1 by discussing volumetric data. In Sections 2.2–2.4 we describe various visualisation techniques. Finally Section 2.5 provides a summary.

2.1 Volumetric data

Volumetric data comes in various forms, either: rectilinear, structured, unstructured or hybrid — illustrated in Figure 2. When volumetric data was introduced in Chapter 1, it was defined as a stack of CT reconstructions. This form is classified as rectilinear and it is the most common in medical imaging. When rectilinear data has an equal resolution in each dimension, it is classified as uniform. Uniform data seldom occurs in CT data because of differences in slices resolution and slice separation.

The data points constituting a volume are know as *voxels* (*vo*lume *el*ements). Voxels are the equivalent of pixels extended into a third dimension. Voxels may be interpreted in two different ways. Either as point samples with no dimension; or small cubes whose values are the average value of the region they represent. Regardless of the interpretation, voxels sample a continuous field in three dimensional space. Sampling theory applies to volumetric data, as it does to images. To faithfully represent an object, the voxels need to sample the object's space at more than twice the highest spatial frequency (as per the Nyquist Theorem)[33]. Even in full data acquisition and reconstruction methods, the subject is seldom sampled at rates high enough to form an accurate representation. Consequently,

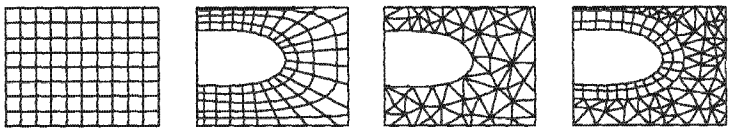


Figure 2: Example cross sections of volume formats. From left to right: rectilinear, structured, unstructured and hybrid formats

aliasing artifacts are often visible in the data prior to visualisation.

Volumetric data sets are usually large. A moderately sized volume with dimensions $256 \times 256 \times 256$ consists of more than a sixteen million voxels. Improvements to the scanning technologies and increases in computational power are allowing larger volumetric data sets to be constructed ($\geq 1024^3$). Consequently, efficiency is a concern.

Many of the volume visualisation techniques sub-sample the data to improve the resulting image quality. The volume is resampled using interpolation functions. With rectilinear data, one dimensional interpolation functions can be extended to three dimensions by considering each dimension separately. Data sampled according to Nyquist's Theory, can be interpolated perfectly using the sinc $= \frac{\sin(\pi x)}{\pi x}$ function — see Figure 3. However, the sinc function has an infinite extent in the spatial domain and, therefore, is computationally infeasible. Consequently, non-ideal interpolation functions with a finite extent are used, for example nearest neighbour, linear, cubic, spline and truncated sinc. The higher order functions provide better interpolation results at higher computation costs. Linear interpolation provides the best trade off between speed and quality[137]. Bilinear and trilinear interpolation refer to the use of linear interpolation in a progressive manner to interpolate in two and three dimensions respectively.

2.2 Simple volume visualisation

The seminal work in volume visualisation used relatively simple techniques [137]. One of the first techniques displayed the volume data as images or "slices". The slices were

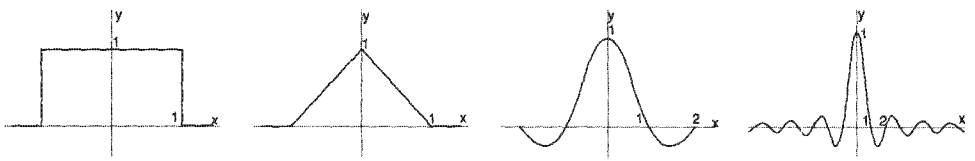


Figure 3: Graphs of common interpolation functions. From left to right: nearest neighbour, linear, truncated sinc and sinc

viewed primarily from three view points, aligned with each of the axes of the volumetric data. In medical terms, these views are referred to as the axial, coronal and sagittal views. Arbitrarily oriented planes were once less common, but have become standard. Medically, the extraction of such planes is referred to as multi-planar reformatting (MPR).

Planar representations do not reflect the three dimensional nature of the volumetric data. Consequently, complex structures and their spatial relationships are not easily interpreted [46]. The earliest recorded 3D technique, called Binary Voxels, represents voxels as solid cubes. The voxels are assigned a value indicating whether they are transparent or opaque. The opaque cubes are rendered directly. While this technique produces blocky solids, it introduces the fewest possible visualisation-related artifacts; no smoothing and no quantisation errors.

The *binary* decision as to whether a voxel should be *classified* as opaque is done via a boundary tracking algorithm or threshold-based segmentation. Segmentation is the process of determining to which objects a voxel belongs based on its position and value relative to the surrounding voxels. Techniques using binary classification are prone to creating or filling holes in the displayed objects. This problem is a result of the *partial volume effect*, where a voxel may represent a mixture of objects. Many of the ideas and problems of these original techniques are still relevant. Modern techniques can be categorised as either surface-based or voxel-based.

2.3 Surface-based volume visualisation

Surface-based techniques extract the boundaries of homogenous regions (*isosurfaces*) using binary classification of the voxels. The most popular techniques extract surfaces as polygonal meshes or through ray casting. These techniques reduce the blocky appearance of extracted objects by interpolating between the voxels during extraction.

2.3.1 Polygonal meshing

The isosurfaces are extracted in a preprocess. The resulting polygonal mesh is displayed using traditional polygonal rendering techniques. The extraction process is complex and computationally expensive. One common approach builds contours of the objects in each slice using advanced segmentation techniques. The contours are then stitched together to form a triangle mesh in a process known as *tiling* [123] — See Figure 4. Tiling techniques struggle with contours which branch between slices (topological changes). Further processing is required to handle branches. However, the resulting semi-regular meshes are smoother

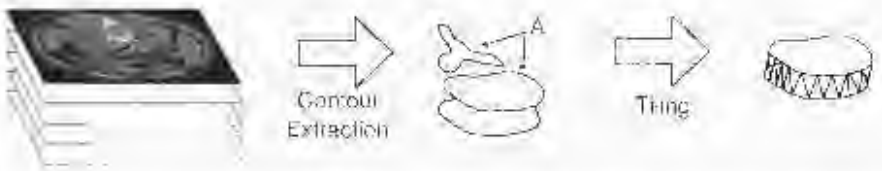


Figure 4: Tiling of extracted contours between two adjacent slices. Contours are extracted through segmentation. The contours are then stitched together using tiling techniques. Objects which branch between slices (indicated by A) are problematic.

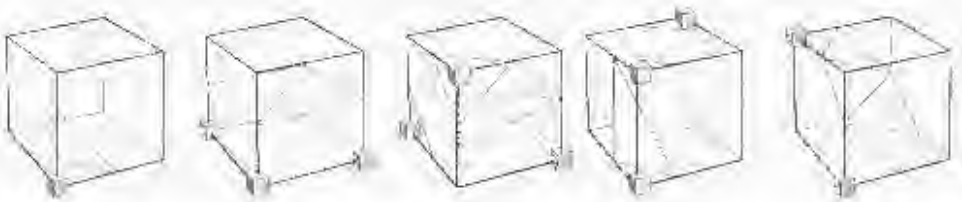


Figure 5: Five of the possible configurations of the voxels and the resulting isosurfaces as described by the marching cubes algorithm. The vertices marked with voxels are considered inside the isosurface.

and more suitable for rendering.

Another approach is that of the marching cubes algorithm [86]. The marching cubes algorithm visits each voxel in the volume. Eight neighbouring voxels are processed at a time with each voxel classified as lying inside or outside the isosurface depending on its value. The eight voxels can produce $2^8 (= 256)$ configurations of inside and outside vertices. The isosurface shape is defined by the configuration — see Figure 5. The set of unique configurations (ignoring symmetry) are recorded. The configuration of the voxels is used to create an index into the table of recorded configurations. The table entries describe the triangles making up the surface. (The exact vertex position is found by interpolating the voxel data.)

The original marching cubes algorithm has several downfalls, which have been addressed by various extensions. For example, the original algorithm used a table of only 14 configurations. Several ambiguities were discovered within these configurations, which resulted in erroneous isosurface definitions. The ambiguities have since been resolved by extending the list of configurations and subdividing the cube to eliminate the ambiguous conditions [95, 138]. The marching cubes algorithm is prone to producing meshes consisting of many thin triangles. Thin triangle meshes are inefficient to render. However, the thin triangles can be merged and removed by further mesh processing [12].

While mesh extraction preprocesses are time consuming, the resulting meshes can be displayed in real time using current graphics hardware. Consequently, viewpoint manipulation can be done in real time to support region of interest discovery.

2.3.2 Ray casting

An alternative to polygonal mesh extraction uses ideas developed in ray casting. For each pixel in the image, rays are fired from the viewpoint through the volume. The volume is sampled at uniformly distributed points along each ray. The samples are classified as inside or outside the desired isosurface. The visible isosurface is found between the first two samples which are classified as outside-inside. Figure 6 illustrates this process for a single ray. As soon as the surface is found the ray casting stops.

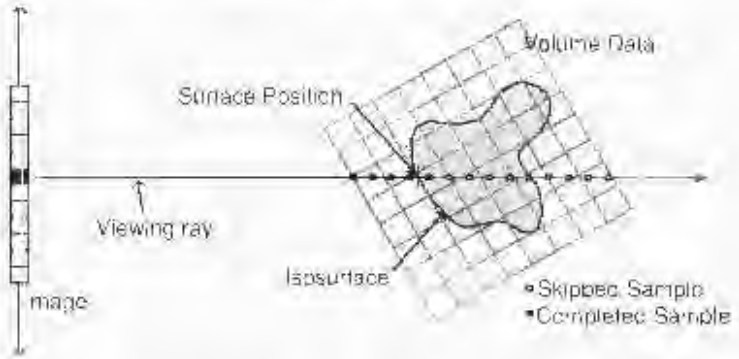


Figure 6: Surface extraction through ray casting. A ray is "fired" for each pixel. The volume is sampled at regular intervals along the ray until the isosurface is intersected.

Techniques for generating isosurfaces directly from the voxels are closely related to voxel-based volume rendering techniques. Recent work by Engel et al. [29] uses the concepts developed to accelerate voxel based rendering to accelerate isosurface rendering.

2.4 Voxel-based volume visualisation

In voxel-based volume visualisation, voxels are treated as semi-transparent particles. Images are generated by modeling how light rays interact with the voxels. Each voxel discolours and attenuates the light according to a pair of coefficients (colour and extinction). Images illustrating different internal structures and compositions are generated using different coefficients.

Through careful selection of the coefficients, this method can be used to to illustrate interior structures and compositions.

2.4.1 The principles of voxel volume rendering

The origin of volume rendering can be traced back to the light transfer equations used in physics, specifically light transport theory [75] and Boltzmann's equation. Boltzmann's equation was adapted by Kajiya [62] to develop "The Rendering Equation" of computer graphics. The rendering equation is computationally expensive, consequently several simplifications were made to make it feasible for use in volume rendering. The simplifications are: (1) Only primary light rays are considered (no reflections), (2) interactions with the regions outside the data set are ignored, (3) voxels are treated as having isotropic absorption, and (4) the only light entering the volume comes from a limited number of point light sources [77, 26]. The resulting equation, aptly named the *volume rendering equation* (VRE), is given as:

$$I = \int_{d_0}^{d_n} \kappa(x(\lambda)) e^{-\int_0^\lambda \tau(x(\lambda'))d\lambda'} \, d\lambda \tag{1}$$

This equation describes the colour of a ray of light, I, after having passed through the volume. The ray has been parameterised as $x(\lambda)$, where λ is the distance from the viewer. Only the regions inside the volume are considered, consequently d_0 and d_n represent the volume entry and exit distances. $\kappa(\rho)$ and $\tau(\rho)$ are functions, returning the colour and extinction coefficients for any given point, ρ, in the volume data. These are defined as $\kappa(\rho) : \mathbb{R} \to \mathbb{R}^3$ and $\tau(\rho) : \mathbb{R} \to \mathbb{R}$. The final colour, I, is the sum of all the colours along the ray after having been attenuated by the extinction coefficients between the viewer and the point of emission.

An analytical solution to the VRE does not exist and it must be evaluated numerically. A common approach uses Rieman Sums (Rectangle Rule) for n equal ray segments of length $\ell = \frac{d_n - d_0}{n}$ [75, 112, 29]. Giving the following solution:

$$\begin{aligned} I &= \sum_{i=0}^{n-1} \kappa_i \ell \cdot e^{-\sum_{j=0}^{i-1} \tau_j \ell} \\ &= \sum_{i=0}^{n-1} \kappa_i \ell \prod_{j=0}^{i-1} e^{-\tau_j \ell} \end{aligned}$$

Where

$$\kappa_i = \kappa(x(d_0 + i\ell))$$

$$c_i = \epsilon(x(d_i + i\ell))$$

If we let

$$\begin{aligned} \alpha_i &= 1 - e^{\kappa_i \ell} \quad \text{opacity at sample } i \\ c'_i &= \frac{\kappa_i}{\alpha_i}\ell \quad \text{colour at sample } i \\ c_i &= c'_i \cdot \alpha_i \quad \text{premultiplied colour and opacity} \end{aligned}$$

We find

$$\begin{aligned} I &= \sum_{i=0}^{n-1} c_i \cdot \prod_{j=0}^{i-1} 1 - \alpha_j \\ &= c_0 - c_1(1-\alpha_0) - c_2(1-\alpha_0)(1-\alpha_1) + \dots \\ &\quad + c_{n-1}(1-\alpha_0)(1-\alpha_1)\dots(1-\alpha_{n-2}) \\ &= c_0 \text{ over } c_1 \text{ over } c_2 \dots \text{ over } c_{n-1} \end{aligned} \tag{2}$$

The terms *opacity* (α), its inverse, *transparency* ($1-\alpha$) and the *over* operator are common in digital image compositing [102] and provide a useful shorthand which has been adopted throughout volume rendering literature. Equation 2 is known as the volume compositing equation. It may be written recursively as the blending operation:

$$I_{i+1} = I_i(1-\alpha_i) + c'_i\alpha_i$$

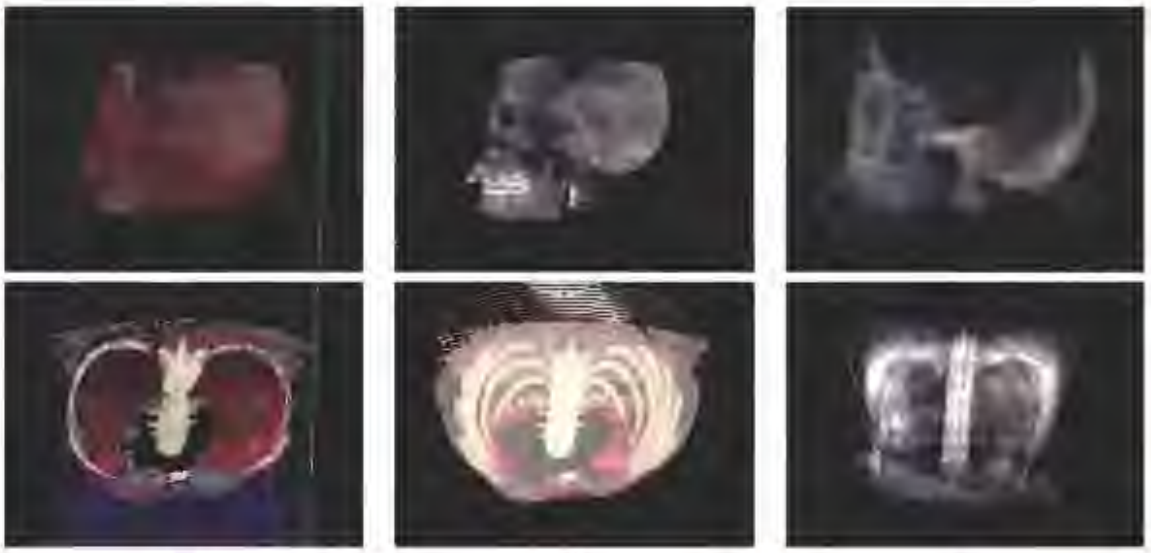
for back to front compositing. I_i is the accumulated colour prior to the ray passing through the voxel and I_{i+1} is the resulting colour, which is closer to the viewer. Reversing the direction of compositing changes the recursive definition and requires the accumulation of both opacity and colour. The alternative blending functions are:

$$I_{i-1} = I_i + (1-A_i) + c'_i\alpha_i \tag{3}$$

$$A_{i-1} = A_i + \alpha_i(1-A_i) \tag{4}$$

A_i represents the accumulated alpha values. I_0 is the colour of the background, and is opaque.

While the volume compositing equation is the most common, it is not the only one [112]. Other approaches include the maximum or minimum intensity projection (MIP) and the weighted additive projection (WAP). MIP uses the maximum or minimum value along the ray as the final colour. WAP blends all the opacity values along the ray together with white, modeling the characteristics of an x-ray. Figure 7 shows image generated using these three compositing techniques. MIP is useful when a contrast medium is present in the data set,



(a) (b) (c)

Figure 7: Images generated using different volume compositing techniques. a) Standard volume composit, b) Maximum Intensity Projection (MIP), c) Weighted Additive Projection (WAP). The top row was rendered from the LODOX SACT reconstruction data set, Elliot. The bottom row was rendered from a standard CT reconstruction of an abdomen.

while weighted additive projection is favoured by clinicians who prefer viewing X-rays and wish to have none of the internal structures obscured.

2.4.2 Classification

Classification in voxel-based rendering refers to the process of mapping colour and extinction coefficients to voxel values. The mapping function is known as a *transfer function* and it is traditionally represented as a look up table. In a formal sense, classification refers to the definition of the two functions $\kappa(\rho)$ and $c(\rho)$ in Equation 1. The power of direct volume rendering lies in the transfer function specification. It is very difficult to specify a transfer function which produces an informative visualisation.

Voxel data occasionally includes tag and colour information, for example the Visible Human data set [97]. Tags label groups of voxels identified as belonging to specific objects in the volume. The tags are calculated using advanced segmentation techniques and may be used along with any available colour information to produce transfer functions local to the objects [129, 116, 14, 65]. Specifying these local transfer functions is a relatively easy task with a few degrees of freedom.

For standard data sets, the transfer functions must be defined for the entire data set.

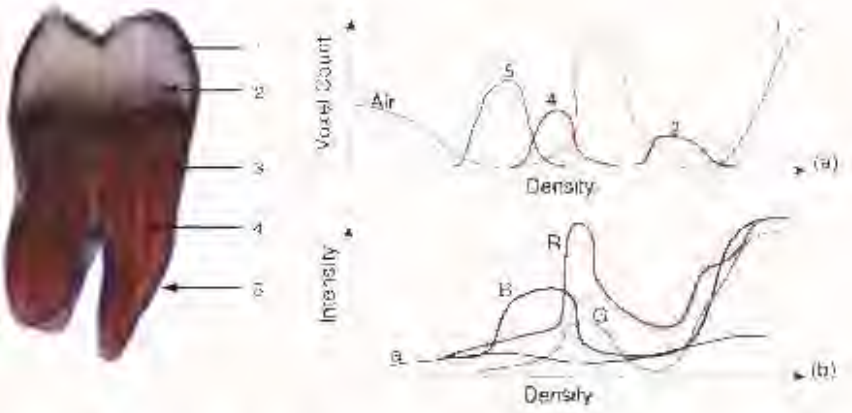


Figure 8: Traditional transfer function specification. The tooth is made up of various parts (1–5). Each part is associated with a range of densities which overlap with other parts. A graph of the object-density associations is given in a). In an arbitrary volume these relationships are unknown and are discovered through trial and error. The relationship is useful to know when trying to assign colour and opacity values. In b) the transfer function is given for the rendered tooth. The mapping was specified by setting the individual red, green, blue and alpha components using a set of splines.

Traditionally, transfer functions were specified manually by manipulating the colour and opacity values in the look up table. Figure 8 illustrates a transfer function specified using 4 curves: a curve for the opacity, red, green and blue components. These curves are manipulated as splines or a sequence of line segments. At best, this approach contains several hundred degrees of freedom which a viewer must explore via trial and error to find a quality transfer function. Immediate visual feedback allows the effects of changes to the transfer function to be tracked. However, this may not be possible due to processing demands.

Research [101] is underway with the aim to automate this process. Thus far, two categories of automation have emerged. The categories are defined by: (1) transfer function specification based on rendered images, and (2) transfer function specification based on the underlying volume data properties.

Image centric approaches [45] sample the parameter space of the transfer function and generate thumbnail renderings. Through user selection, in a query type process, the parameter space is narrowed until a desirable transfer function is found. Unfortunately, thumbnail generation is time consuming and the process relies on user evaluation and experience to determine the best rendering, which may not be the best transfer function.

Data centric approaches extract signature features from the data, which are used to

guide transfer function parameter selection. A model of the underlying data is often used in combination with the extracted features to generate a transfer function [32, 68, 34, 69]. These models assume the viewer is interested in the boundaries between objects in the data. For example, Kindlmann and Durkin [68] use the voxel value, the first and the second directional derivative to build a *3D* histogram which is searched for a characteristic boundary signatures. The voxels contributing to those signatures are assigned appropriate opacity values (colour assignment is left to user). Fujishiro, et al. [34], on the other hand, use Hyper Reeb graphs to build transfer functions. The skeletal surface topologies described by the Hyper Reeb graphs are used to find "critical isosurfaces". The transfer functions are then built to accentuate these surfaces. Some data-centric approaches are not based on a data model, and produce a set of metrics which the viewer uses to construct the transfer function. The data-centric approaches are sensitive to noise and coarse sampling of the data, which is likely to make them ill-suited for use with SACT and LACT data sets.

Recently, *multi-dimensional transfer functions* (MDTFs) have become popular [69]. MDTFs use multiple voxel attributes when mapping to the colour and extinction coefficients. They offer spatially localised classification which allows the transfer function specification to be localised to regions in the data. Multi-Dimensional transfer functions are more complicated to design and require careful user interface considerations. MDTFs are limited by current hardware which only provide support for one dimensional transfer functions. Only a limited number of MDTFs, which can be mapped to a single dimension, are used.

2.4.3 Rendering algorithms

Several rendering algorithms exist to evaluate the VRE (Equation 2) and are categorised according to the order in which the voxels are traversed. The categories are: image order, object order and hybrid. The categories are an indication of the properties and forms of coherence which may be exploited to accelerate rendering.

Image order algorithms

Image order algorithms generate a viewing ray for each pixel in the image. The rays are intersected with the volume, traversing the voxels front-to-back, in a process known as *back projection* — see Figure 9a. The voxel data is accessed randomly with respect to the internal representation. Ray tracing[82] and ray casting[12] are two methods which fall into this category. In ray casting the voxels are traversed using discreet steps, for example, the grayed blocks in Figure 9a. Ray tracing calculates the exact voxel-ray intersections and uses trilinear or higher-order interpolation functions to determine the sample values. Ray

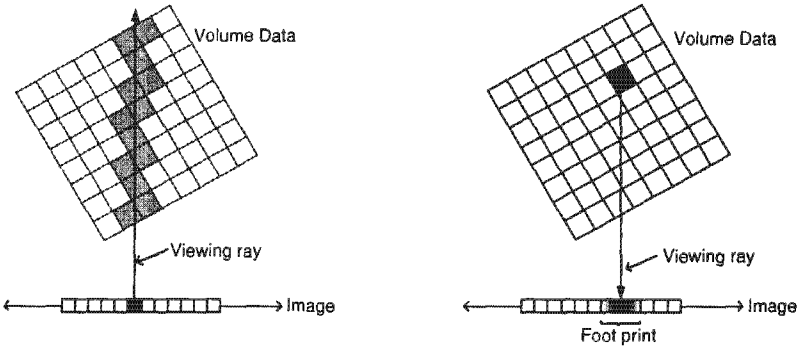


Figure 9: Rendering algorithms can be categorised according to the way in which the voxels are traversed: a) Image order rendering, and b) object order rendering, and hybrids (not shown).

casting requires fewer calculations and is faster. However, the coarse sampling produces lower quality images.

Several techniques may be applied to accelerate this class of rendering algorithm. When parallel viewing is used, a single template ray may be used to find all the intersections for all the rays (exploiting ray coherence) [141].

Early ray termination is possible in front-to-back traversal. The ray-volume traversal stops as soon as the accumulated alpha is "opaque" [82]. *Space Leaping* [104], may be used to skip over empty regions and sample uniform regions once (exploiting object space coherence). Space Leaping has traditionally been implemented by imposing a hierarchical structure on the volume data (usually octrees or kd-trees). The structure subdivides the volume into regions of constant value. Traversing these structures during ray-volume intersection calculations may be prohibitively expensive (depending on the spatial frequencies in the data). An alternative method embeds proximity information into the voxel data. The proximity information is represented as an integer value, indicating the minimum distance in voxel steps to the next region.

Further acceleration is possible by trading image quality for speed via incremental rendering [78]. A rendering is generated by sparsely sampling the image space and traversing the volume with a smaller number of rays. The resulting small image is resized, effectively interpolating the samples. Later passes across the image space recursively sub-sample the image, traverse the volume and fill in the pixels which were skipped. Rapid changes in colour are used as an indication of complex regions which should be concentrated on, while pixels of similar colour which are close together are used as an indication of uniform regions. These regions can be approximated with interpolation [140].

Object order algorithms

Object order algorithms traverse the voxels sequentially. This makes the effective use of memory caches possible — Figure 9b. Volume traversal may begin at either the front or the back of the volume. Traversing front-to-back allows early traversal termination (similar to early ray termination). However, back-to-front methods can use the blending hardware found on graphics cards. Voxels are projected into the image for blending (*front projection*) and may affect several pixels. The pattern of pixels effected is the *footprint*.

Splatting [57], *cell projection* [108] and texture-based rendering [15, 91, 131] fall into this category. In splatting, voxels are represented as spheres which are projected into the image as ellipsoids. In parallel viewing, the footprint calculations are pre-computed and stored in a look up table. In perspective viewing the projection calculations are simplified by using an ellipse to represent the voxels. In turn, the ellipses are represented as small shaded polygons which can be efficiently transformed, projected and blended into the image using graphics hardware. The initial size of the ellipses is a system parameter selected to ensure neighbouring voxels overlap when projected into the image. Space leaping may be used to skip empty regions of the volume.

Cell projection extends splatting by subdividing the volume using a hierarchical structure. This allows voxels of constant value to be grouped into regions (known as cells). The cells are then "splatted". The larger regions require fewer calculations and this reduces the traversal time. The hierarchical structure may be used to support incremental rendering. This is achieved by recursively moving down the hierarchy and rendering the approximate voxel groupings at each level.

Hybrid algorithms

Hybrid techniques [75, 131, 80] combine the advantages of both image and object order rendering. The best known hybrid is the *shear-warp* algorithm [75]. The shear-warp algorithm exploits coherence in voxel traversal (object space), pixel traversal (pixel space) and between rays by decomposing the rendering process into several stages, as illustrated in Figure 10.

The volume is represented as a stack of slices with the slices aligned to the volume axes. In the first stage, the slices are sheared and scaled resulting in viewing rays which are perpendicular to the slices. Stage two composites the transformed slices, using blending Equations 3 and 4, into the intermediate image. As the transformed volume data and the intermediate image are aligned, they can be traversed efficiently in scan line order. Run length encoding is used within each slice to allow homogenous and empty regions in the

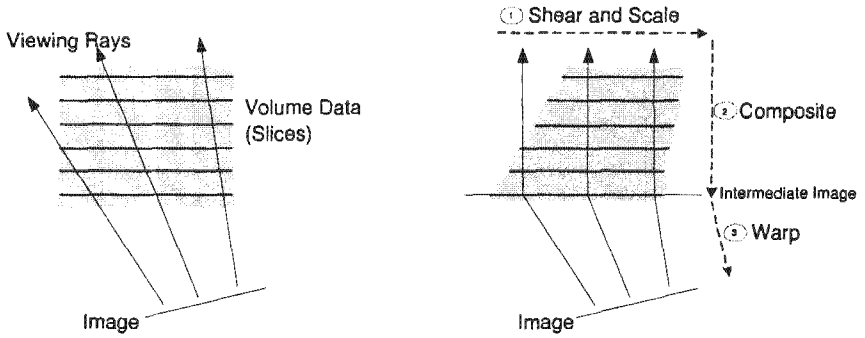


Figure 10: Shear warp volume rendering. The volume rendering process is broken down into several stages as indicated. By breaking the rendering process up, coherence in the rendering can be maximised.

volume to be skipped during compositing. Similarly, run length encoding may be used in the image to skip over rays that have been early terminated. In the final stage the intermediate image is warped back to the image plane. Perspective and parallel viewing are handled by the same algorithm. Parallel rendering differs from perspective rendering in that a unit scale factor is used during stage one.

The main disadvantages of this technique are the memory requirements and occasionally visible under-sampling. The three possible slicing axes implies that three copies of the volume must be stored in memory, ordered according to the slice direction. The slice direction is chosen to minimise the angle between the slice direction vector and the view vector. Consequently, it is possible that all three volume stacks will be needed to render a single image. Sampling artifacts occur as the angle between the stacks and rays enlarge. The larger angles require a larger shear which noticeably pulls the slices apart. The sampling artifact is a consequence of replacing the trilinear interpolation with bilinear interpolation and is a fair trade off between image quality and speed. The original shear-warp algorithm was implemented in software, and at the time, provided the highest rendering rates. Several hardware-based variations of this algorithm have since emerged.

Hardware support

Volume rendering can be accelerated considerably through the use of hardware. Two forms of hardware acceleration exist: dedicated hardware and general purpose graphics accelerators.

Dedicated volume rendering hardware solutions implement one of the algorithms mentioned previously. Several hardware implementations exist. The more popular systems are:

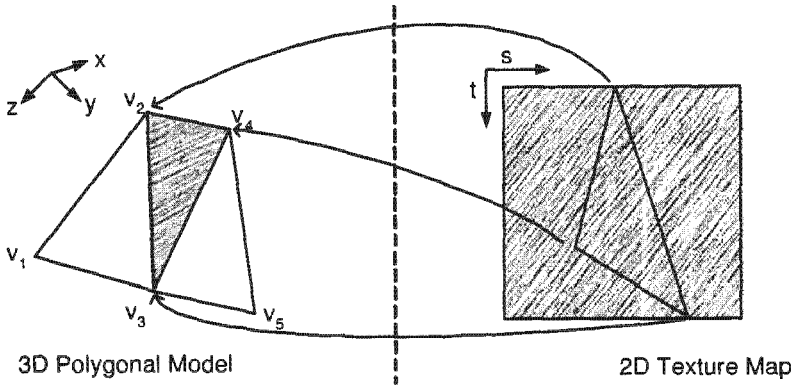


Figure 11: 2D texture mapping. The texture coordinates, (s, t), are specified for each vertex, v_i, of a polygonal mesh. The hardware maps the texture onto the polygons.

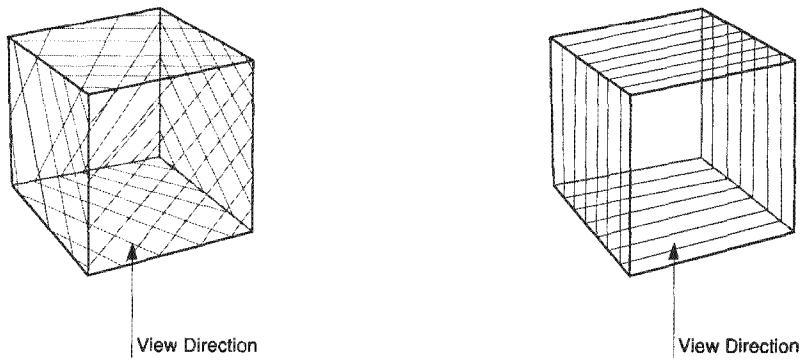
RACE II[105], VIRIM[41], VIZARD II[92], VG-Engine[133] and VolumePro [100]. VolumePro is the most commercialised system. The hardware implements a ray casting algorithm based on the shear-warp volume decomposition on a single chip. This solution is available as a stand-alone chip for custom applications, or as an add-on card for workstations.

In general, dedicated hardware is not widely available and relatively expensive compared to general-purpose graphics accelerators. Many of the implementation problems faced when using general-purpose hardware are solved by the dedicated hardware. For example; dedicated hardware can provide larger precision for opacity and colour accumulation buffers and native support for 12 or 14 bit voxels (common data precisions in medicine). Furthermore, the hardware can be built to perform more accurate gradient estimations (used for lighting and shading calculations) and perform higher order data interpolation (for example tricubic or spline) at little extra cost.

On the other hand, general purpose graphics accelerator cards are becoming cheaper, more powerful and widely available. The texturing hardware available on these cards may be used to perform the blending operations.

Traditionally, textures are images used to represent surface details in polygonal models. The texturing hardware maps images onto the polygons using a set of texture coordinates for each vertex in the polygon — Figure 11. Texture elements contain colour information(RGB) and optionally a transparency value (α) of the polygonal surface. Textures can be one, two or three dimensional.

If three dimensional (3D) texturing hardware is available, the volumetric data can be formatted and uploaded to the graphics card as a 3D texture [15, 131]. The 3D texture is applied to a series of generated polygons. The polygons cut through the region represented



(a) (b)

Figure 12: Texture-based direct volume rendering. a) View direction aligned polygons are used when 3D textures are available, otherwise b) Object aligned polygons are used with 2D texturing.

by the volume. They are aligned to the view direction so that they are parallel to the screen and clipped against the volume boundaries — Figure 12a. The polygons are rendered and blended together back-to-front.

If only 2D texturing hardware is available, the volume may be represented using a series of 2D textures, aligned with one of the volume axes. The polygon slices are aligned similarly — Figure 12b. The 2D texture approach can be likened to the shear-warp technique. It requires three copies of the volume data ordered according to the slice direction. The slice orientation and the volume ordering used during rendering depends on the view direction. And only bilinear interpolation of the voxels within each slice is used (which results in the same sampling artifacts). The sampling artifacts are the worst when the angle between the view vector and the slicing direction vector is at its greatest (45°) — see Figure 13.

Engel, et al. [28] have developed a technique to reduce the sampling artifacts using the graphics hardware to do linear interpolation between the polygons, effectively achieving trilinear interpolation. The 3D texture method does not suffer from these problems as the trilinear interpolation is implemented by the graphics card. In addition, a single copy of the volume is required.

2.5 Summary

In this chapter volume visualisation techniques were introduced. The basic visualisation approaches were presented, being: simple slice viewing, surface-based 3D rendering and voxel-based 3D rendering.

an example see Figure 17b. The button appears depressed or extruded depending on the vertical direction of the shading (relative to the viewers head orientation). If only horizontal shading is used, the perceived depth will be ambiguous, changing randomly.

Atmospheric perspective

Atmospheric perspective occurs in large panoramic views in which atmospheric conditions effect the light reaching the viewer — Figure 17c. The atmosphere (fog, smoke, mist, etc.) reduces the light reaching the viewer, effectively lowering the contrast of parts of the scene proportional to their distance from the viewer. This effect is also known as aerial perspective or depth cueing.

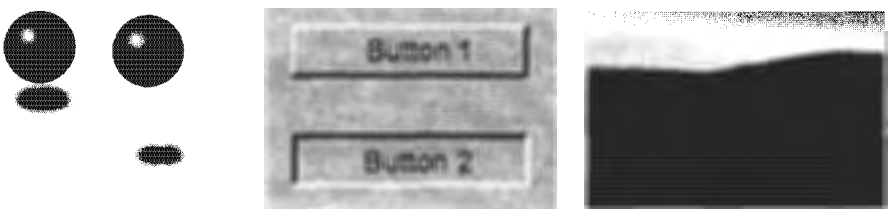
3.2.2 Physiological depth cues

Physiological depth cues are extra-retinal[103] and relate to physical changes of the eye and/or the viewing environment.

Motion cues

The amount and direction of motion of objects in an image depends on the relative position and motion of the physical objects (*motion parallax*). Distant objects move relatively slower than closer ones. Figure 18 illustrates radial and linear motion parallax.

The human mind automatically integrates motion spatially over time. This process, known as *Kinetic Depth Effect* (KDE), allows the underlying geometry, including depth, to be estimated. KDE is based on the assumption that objects under consideration are



(a) (b) (c)

Figure 17: a)Shadows: The position and size of the shadow indicates depth in the scene and height, b) Shading: Depending on the direction of shading objects may appear protruded or depressed, c) Atmospheric perspective: Parts of the scene become hazy over distance.

(a) (b) (c)

Figure 13: Visible aliasing artifacts in 2D texture based volume rendering. (a) An overview of the foot data set, with a zoom region indicated. (b) When the angle between the slice direction vector and the view vector is small the sampling errors are only slight, however (c) as the angle grows, the sampling errors become clearly visible.

Surface-based visualisation was covered briefly - with the two main techniques (mesh extraction and ray tracing) highlighted. Voxel-based visualisation was introduced in more detail with an overview of its origin and the basic idea. Included in the discussion was the definition of classification. Classification was described and it was emphasised that this process represents the true power and difficulty behind voxel based rendering. The automatic and semi-automatic techniques for classifications were sketched. Volume rendering algorithms and acceleration techniques in both software and hardware were described and categorised.

Chapter 3

Perceptual depth cues and visualisation

In this chapter perceptual depth cue theory is introduced and its application described in volumetric and X-ray visualisation systems. In Section 3.1 we introduce perceptual depth cue theory. A general description of perceptual depth cues and more detailed information on individual depth cues is given in Section 3.2. This is followed by a discussion on the combinatorial models used in perceptual depth cue theory in Section 3.3. Section 3.4 revisits the voxel-based volume visualisation from a perceptual depth cue perspective. An analysis of the "naturally" occurring perceptual depth cues is given, followed by a review of what cues have been implemented in past systems. Section 3.5 explains how X-rays are generated. This leads to a discussion of what PDCs are "naturally" present, followed by a discussion of what PDC can be added, and what systems have been implemented in the past. In Section 3.6 we account for limiting the study to just two perceptual depth cues and finally Section 3.7 provides a summary.

3.1 Perceptual depth cues theory

Humans have the ability to perceive $3D$ environments from two $2D$ images. One of the theories proposed to explain how this is achieved is that of Perceptual Depth Cues (PDCs) [114]. This theory illustrates that humans rely on contextual psychological information provided within the retinal images, along with physiological properties of the eyes, to locate and position objects relative to one another. The perceived relative distances are combined with prior experience to conject absolute distances [88].

Depth cue theory consists of individual depth cues and models describing how the cues

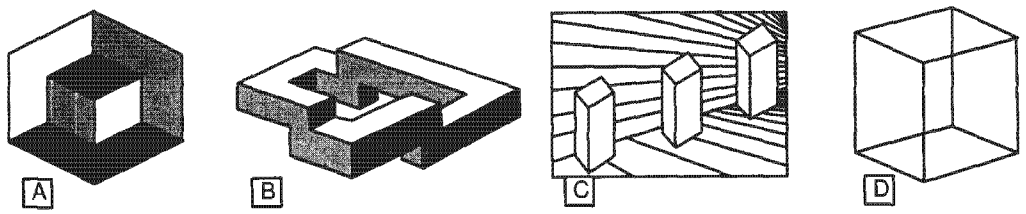


Figure 14: Optical illusions as a result of insufficient depth information: a) A cube in a "room" or a cube with a corner missing, b) Continuous stairs, c) Pillars of equal height do not appear so, d) A cube with two possible front edges.

combine and interact. Due to the nature of PDCs and their interactions, a visualisation study targeting selected PDCs should not neglect the effect of other PDCs which may or may not be present. While simplistic models support an additive combination of depth cues, the reality is that one depth cue can outweigh the importance of another in specific situations. If the interactions of depth cues are ignored, ambiguities are bound to occur. These ambiguities manifest as illusions. For examples refer to Figure 14.

3.2 Perceptual depth cues

PDCs are the "elements" used to perceive depth. They can be described using one of several taxonomies common in perception literature such as [55, 43, 119, 99]. Here, Gilliam's [36] approach is followed, and the PDCs are grouped according to their psychological or physiological nature.

3.2.1 Psychological depth cues

Psychological depth cues are monoscopic, meaning they can be viewed with one eye only[103]. The depth perception resulting from these cues may not be related to actual depth of objects [55]. Consequently, they are frequently the source of optical illusions [136]. However, when presented in the correct context they can provide powerful and effective depth perception.

Perspective

Perspective cues are a consequence of the projective geometry of the eye in which objects are scaled according to their distance from the viewer. Perspective cues can be loosely subdivided into texture gradient, size and size constancy and linear perspective based cues.

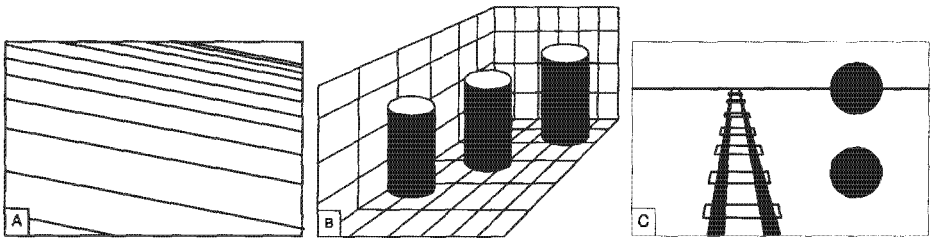


Figure 15: Perspective cues: a)Texture gradient – the texture in the top right corner has a higher spatial frequency and appears to be further than the front left, b)Ambiguities as a result of size constancy failure. The cylinders are either the same size at different heights, or different heights at different depths, c) Linear perspective – lines converge to vanishing points and disappear over horizons. Objects of equal size appear to be at different depths depending on the position in the field of view — the bottom circle appears closer than the top.

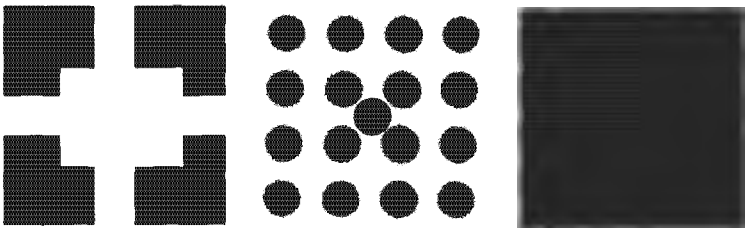
Texture gradient refers to the rate of change of a regular pattern across an image — Figure 15a. The spatial frequency of the texture increases as it extends into the distance. Texture gradient, combined with texture shape changes, provide a strong sense of depth and conveys the "surface" shape, including curvature [59].

Size constancy is stated by *Emmert's Law*, which says that the perceived size of an object is proportional to the retinal size and the perceived depth. Size constancy fails when too few depth cues are present. This results in perceptual ambiguities in depth versus size — refer to Figure 15b. Changes in perceived distances are more readily accepted than changes in perceived size. Consequently, a reduced copy of an object is usually perceived as more distant than the original.

Linear perspective occurs when objects in an image extend into the distance towards a vanishing point or horizon. Parallel lines converge as they approach the vanishing points. In addition, the horizon provides a "perceptual reference plane" - objects below the horizon appear closer than the same object displayed higher in the field of view — Figure 15c.

Occlusion

Occlusion occurs when closer objects partially block out more distant objects and the viewer is able to order the objects spatially. Occlusion information may provide strong depth perception or completely destroy it if incorrectly presented [98]. Humans have an uncanny ability to complete partial objects by inferring much of the missing information. Transparency and false or illusory contours are side effects of this process. Figure 16a illustrates false contours.



(a) (b) (c)

Figure 16: a)Occlusion – False contours: a white square is perceived as covering the 4 black squares. b) Depths of focus: the blurry objects appear deeper (part of the background). c) Colour: Red appears closer than blue.

Depth of focus

Depth of focus describes how in-focus objects are sharply defined, while nearer or further objects are blurred — Figure 16b. This effectively separates the background and foreground regions of an image. This cue is difficult to reproduce dynamically as it requires both gaze and focus tracking. Currently, depth of focus is implemented by bluring/filtering parts of the image. However, this is computationally expensive and is used predominantly for artistic effect in static scenes. Gaze tracking combined with level-of-detail rendering is being used to emulate this cue crudely [106, 79, 10].

Colour

Similar objects with different colours are perceived to have different depths [136] — for example, the red and blue objects in Figure 16c. The use of colour for depth discrimination is limited by colour constancy constraints. Colour constancy describes how regions of constant colour or low colour variation are grouped into single regions.

Shadows

The position and size of shadows allows the relative depth of objects to be estimated from a reference plane — Figure 17a. Objects with self shadows highlight their surface details. Shadows need not be realistic [99], but they should have soft edges as this is how they are distinguished from objects [136].

Shading

The HVS is "coded" to handle objects lit by a single overhead light source (the sun). The play of light across surfaces (direction of shading) provides depth information. For

rigid [25] and explains how a point cloud, which appears as a flat random scattering of points when static, becomes a structured object with depth under motion.

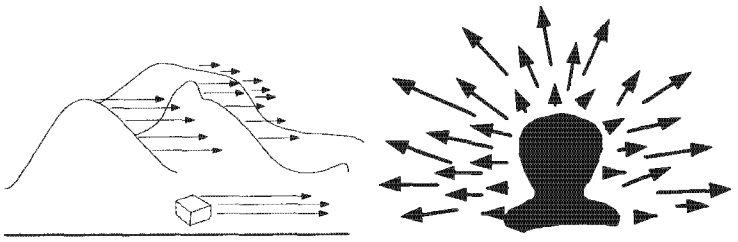


Figure 18: Linear and radial motion parallax. The velocity vector's magnitude is determined by the distance from the viewer while the direction is determined by the view and motion directions. Linear parallax occurs when the motion and view direction are orthogonal. Radial parallax occurs when the directions are parallel.

Binocular disparity

Each eye has a different view point, consequently each eye receives a slightly different image. The differences between the two images are referred to as *binocular disparity.* When the brain merges the two images, the disparity is used to estimate depth. *Random dot stereograms* (RDS) [61] illustrate the effectiveness of binocular disparity. The collection of spatially random dots used in an RDS has no information other than disparity. Yet patterns and shapes are visible at different depths when an RDS is viewed stereoscopically.

Eye convergence and accommodation

Eye convergence and accommodation are oculomotor, requiring muscle movement. Accommodation refers to the flexing of the ciliary muscle which changes the shape of the lens, altering the effective focal length. Convergence refers to the rotation of the eyes, allowing for objects to remain focused as they move further or closer to the viewer. The angle formed by the intersection of the central lines of sight from each eye is the *Vergence Angle.*

There is a common misconception that the Vergence Angle is used in combination with binocular disparity to triangulate and calculate a "true" 3D position. Studies [88, 18] have revealed that muscle control is not used as part of the HVS and that accommodation and convergence are not physically linked. Accommodation does result in the blurring of regions which are out of focus which is, in turn, used for depth perception [136] (refer to Depth of focus, page 26).

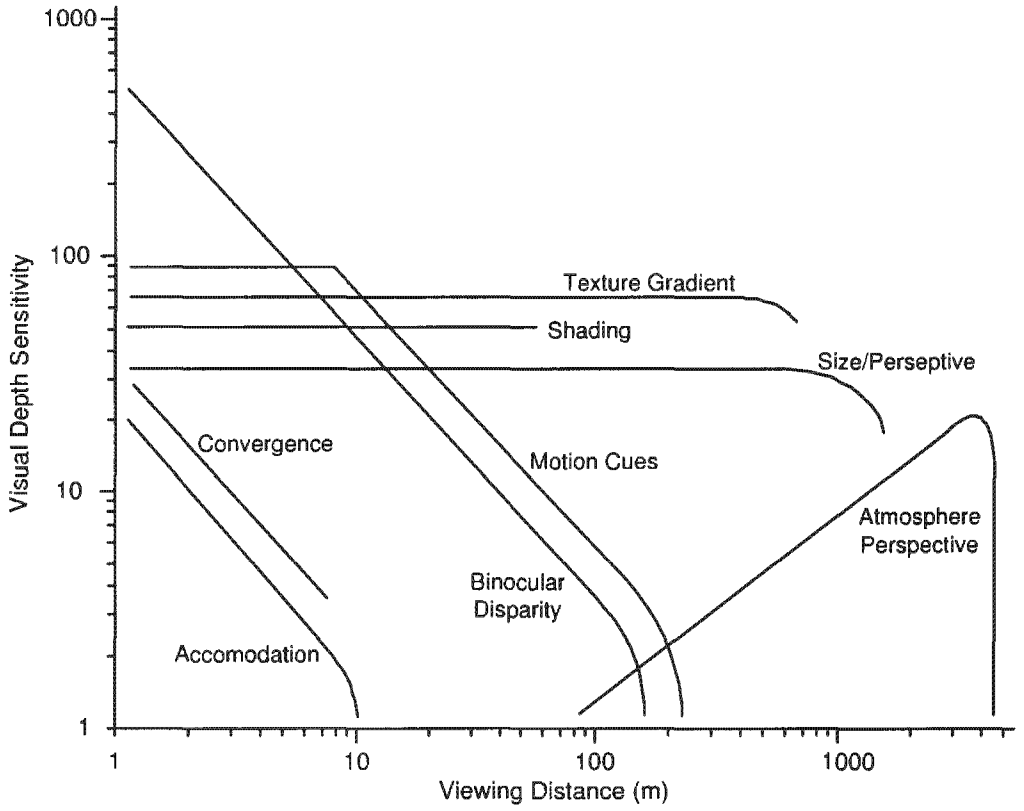


Figure 19: Graph representing the most effective cues over distance. Reproduced from [99].

3.3 Combining perceptual depth cues

Ignoring the affects and interaction between present and missing PDC may result in ambiguities. Beyond this, it is costly and impractical to implement every PDC. In fact, some PDCs cannot be implemented correctly, for example, depth of focus. In such cases it may be beneficial to omit the cue entirely [98]. There are usually several dominant PDCs for a specific task and environment [38, 99]. Only the dominant cues and dominant cue relationships need to be implemented, as these support accurate depth judgements. The human visual system is fairly resilient. Consequently, minor depth cues may provide conflicting depth information without much ill affect provided that the dominant cues are correctly supported.

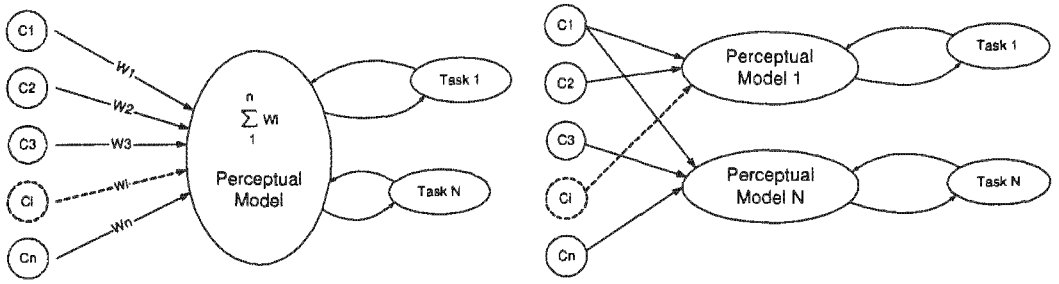


Figure 20: Combination models for perceptual depths cues. The circles represent depth cues. The ellipse on the right represent the tasks at hand. Both cues and feedback from the tasks allow the viewer to build a perceptual model(middle ellipse), which in turn, is used for the task. a) The earlier unified model. The perceptual model for all tasks is based on a weighted sum of all the input cues. b) A newer, more flexible model. Cues are combined according to their reliability and need. Each task is associated with a separate perception.

3.3.1 The unified model

Prior to the formulation of a combinatorial model, a set of ad-hoc rules and guidelines were used to determine dominant cues. Experimental results were used for this purpose. The "unified model of PDC theory" [13, 25] sought to provide a structured explanation of the combinatorial relationships. In this model a perceptual model is built by the viewer for all tasks. The perception is based on the weighted sum of the PDC inputs — see Figure 20a. A later variant of this model uses dynamic weighting of the individual cues according to their reliability for the tasks and environment [142]. The weights are determined experimentally. For example, Figure 19 illustrates how the effectiveness of various cues change over distance. Note that the perspective cues combined with binocular disparity provide the strongest impact over a wide range of viewing conditions [124].

3.3.2 The adaptive model

The unified model cannot effectively account for binary cues (for example occlusion); unreliable cues, which introduce ambiguities if considered at all; or the non-additive nature of cues [136]. Several complex relationships have been identified experimentally. For example, when motion cues and binocular disparity are presented together they are more effective than the sum of the individual cue's effect. Motion cues and binocular disparity are of equal importance in applications requiring the exploration of spatial relationships [109, 119].

Binocular disparity disambiguates and strengthens any psychological PDC [99], while occlusion cues will override both stereoscopic and motion PDCs in most situations [98].

The adaptive model was proposed to rectify some of the problems with the unified model. This model is more flexible and powerful. It involves the evaluation and optional combination of PDC as part of the viewers perceptual model input. In addition, a unique perceptual model is built for each task — see Figure 20b. A consequence of using this model is that each and every task must be analysed to determine the dominant cues. Fortunately, many tasks can be grouped according to a set of elementary tasks, some of which have already been analysed. Several pertinent examples are[136, 110, 7]:

- Tracking 3D Paths.

  Head tracked motion parallax is the strongest depth cue followed closely by KDE and stereopsis. Occlusion, when used with path colouring (the process of assigning colours to individual paths), helps. Interestingly, perspective (rather than orthogonal) viewing does not provide additional information. Neither shading nor shadows are useful.

- Judging surface shape and target surface detection.

  The most powerful depth cues for these tasks are shape from shading and texture gradient. Stereoscopic and motion cues together are very strong. Experiments have revealed that these two cues combine in a complex manner with the individual relative strength of each depending on the exact task.

- Finding patterns in 3D point clouds.

  Motion cues, specifically KDE, are the most important. Stereoscopic depth and colour are advantageous. Size provides a weak depth cue. Generally, the points have no orientation, so shape from shading cues are not available. If the points are represented by planar geometry, the cues may be useful and may be added artificially. Perspective, occlusion and shadows do not provide much useful depth information, especially for a large number of points.

- Estimating relative positions.

  This task is one the most complex to analyse as the dominant cues are dependent on the task and perceived viewer distance. In general, tasks close to the viewer are dominated by stereoscopic depth cues with fine position tasks being augmented by occlusion. Motion cues are *not* useful for fine work, but for coarse work and a perception of overall layout, motion cues, linear perspective, shadows and texture gradient all contribute to depth perception. Drop lines and shadows used with a

reference plane improve depth perception from perspective, size and size constancy cues.

3.4 Perceptual depth cues and volume visualisation

In this section, voxel-based volume visualisation is revisited from a PDC theory perspective. We describe our own analysis of what cues are "naturally" present and which cues can be added. This is followed by a survey of what PDCs have been used in past systems.

3.4.1 PDCs in volume rendering

Our analyses of plain volume renderings (refer back to Figure 7 for examples) reveals the following:

- Texture gradient and colour provide an indication of depth as they allow regions to be coalesced. Neither of these cues are reliable, because they depend on the detail of the data and user specified transfer functions.
- Occlusion and transparency effects provide the strongest depth effect. Objects which appear further back tend to be occluded or partially obscured.
- The compositing techniques which do not provide consistent occlusion information (WAP and MIP) appear "flat" and provide little depth information. An early WAP rendering algorithm [81] using the Fourier Projection-Slice theorem was modified [128] specifically to support shading after the authors realised the disadvantages that arise from a lack of occlusion information.
- Size, linear perspective and shape cues provide limited depth information if the viewer is familiar with the object(s). In the case of WAP and MIP compositing, these are the only depth cues other than motion. Parallel viewing removes the size and linear perspective depth cues completely.

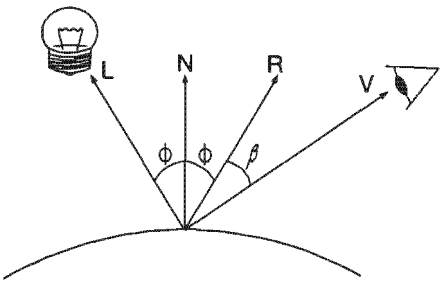
3.4.2 PDC enhanced volume rendering

A review of the volume rendering techniques and enhancements in the literature reveals that most of the PDCs, which can be implemented, have been implemented. Colour depth cues and depth of focus are the two PDCs which have not been used. Colour is used as an attribute of the volume data and it is specified by the viewer during classification. Depth of focus is not possible with current hardware and rendering techniques.

It is ironic that volume rendering has to be "enhanced" to produce additional PDCs. "The Rendering Equation" from which the volume rendering equation originated supported many cues "naturally". However, in the process of simplifying the equation to make it cheaper to compute, the PDCs were inadvertently removed.

Shading

Shading is usually described in combination with classification. Shading effects are created by modeling how the light from point light sources interact with the voxels. The standard approach applies the concepts of Phong's illumination model[33]. The model is used to modify the colour coefficients according to the viewer's position, the light sources positions and the surface normal — see Figure 21.



$$\kappa' = \kappa[k_a + k_d \cos\phi + k_s \cos^n \beta]$$

κ	:	Colour coefficient
k_a	:	Ambient reflection coefficient
k_d	:	Diffusion reflection coefficient
k_s	:	Specular reflection coefficient

Figure 21: The Phong lighting model. L, N, R and V are the light, normals, reflection and view vectors respectively. The colour coefficient, κ is modified according to the equations on the right. The meaning of each constant is given.

Strictly speaking, normals are not defined for volumes. However, they are approximated using volume gradients which are perpendicular to the local curvature. The gradients are typically calculated using central, forward or backward differences[112]. Central difference is given by the equation

$$n_{x,y,z} = \left(\frac{v_{x+1,y,z} - v_{x-1,y,z}}{2}, \frac{v_{x,y+1,z} - v_{x,y-1,z}}{2}, \frac{v_{x,y,z+1} - v_{x,y,z-1}}{2} \right)$$

$n_{x,y,z}$ is the normal of the voxel at the position given by (x, y, z). $v_{x,y,z}$ returns the value of the voxel. Forward and backward differences are computed similarly with the voxel indexing changing accordingly. A comparison of normal approximations is given in [93].

Van Gelder and Kim [131] illustrated shading in texture-based rendering by pre-classifying and shading the data. Meißner, et al. [91] extended this work to support post classification and shading using 3D textures, taking advantage of the features available in later hardware. More recently Kindlmann et al. [69] used shading to "dramatically increase the quality of

rendering". Their implementation was based on features found in current commodity hardware.

Shadows

Shadows are generated by combining the illumination reaching each voxel from each light source. One of the first techniques [63] to produce shadows in volume rendering, accumulated the light contribution in a second volume called a "shadow buffer". Grant [40] presented a modification of the algorithm which used a single 2D shadow buffer to construct a shadow volume. This could compute the effects of all the lights in a single pass. Lacroute [75] implemented the 2D shadow buffer algorithm for use with the shear-warp algorithm, and Behrens and Ratering [8] have shown how the technique can be used in combination with 2D or 3D texture based volume rendering — see Figure 22.

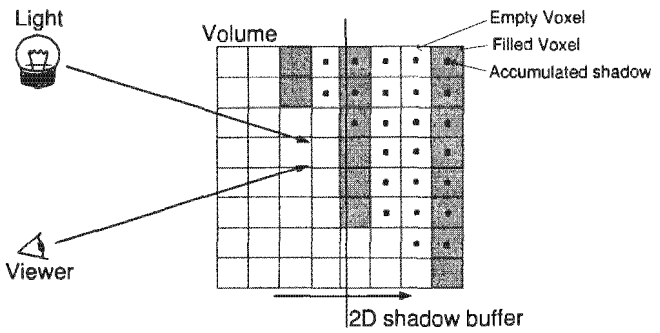


Figure 22: Shadowed volume rendering. The shadow buffer slides through the volume and accumulates the illumination for the current slice. The resulting illumination values are used to modify the colour intensity. The direction of movement depends on the position of the light relative to the viewer. Here it is front to back.

Shadowed texture-based volume rendering is reported to halve the rendering time. The shadow volume is only computed when the light sources change position. Shadow buffer techniques generally restrict the light position to be within 45° of the primary volume axis. In addition, a slice cannot be self shadowed, otherwise severe aliasing occurs.

Behrens and Ratering chose shadows as the means to "efficiently communicate the orientation" of the volumetric data. They reported that shadowed volume rendering was particularly effective when combined with interactive rendering and highlighted the details in the volume. It is, however, conceivable that the motion cues are producing the dominant depth effect in this case.

Atmospheric effect

Atmospheric depth effect, or depth cueing, has not been implemented in texture-based rendering. It has been implemented as a shear-warp extension [75]. Given that the shear-warp is related to texture-based techniques, it is worth mentioning. The effect was implemented by attenuating the voxel's colour coefficients (κ) according to the distance from the viewer. Foreground objects appear lighter than background objects. The attenuation is described as

$$\kappa' = \kappa T e^{-pd}$$

T and p are user defined constants for modifying the strength of the attenuation and d is the distance between the voxel and the viewer.

Motion cues

In many cases motion cues are very important. For example, in standard MIP and WAP composite volume rendering, depth can only be perceived when the volume is in motion. Rendering algorithms which claim to support motion cues are classified as either real time ($\geq 30 frames/second$) or interactive ($> 1 frame/second$). The effectiveness of motion cues at the lower frame rates is questionable. Real time rendering has been a long term goal in volume rendering [67, 66]. KDE is predominantly supported through user viewpoint selection in real time. Motion parallax has been implemented using head tracking. The speed demands for motion parallax are even greater as real time rendering rates are required to prevent noticeable delays and motion sickness [98].

Regardless of how quickly the rendering rates improve, it will always be possible to generate volumetric data which will surpass the hardware resources. In such cases, motion cues can still be provided by using iterative refinement or image-based rendering.

Iterative refinement renders a coarse representation of the volume and refines it (time permitting). During an interactive rendering session, it is possible that the detailed rendering is never seen. In a worst case scenario, only the bounding box of the volume is displayed, in which case no motion cues are presented.

Image-based rendering uses previous renderings to generate a new image. These techniques exploit frame-to-frame coherence. In other words, they exploit the fact that small localised changes occur in the image for small changes in the viewpoint. Typically, a volume is divided into *slabs*. The slabs are rendered, generating a set of billboards ("layered depth images" – LDI). The billboards are then composited for a given viewpoint. Standard LDIs

can be used to generate views within 6° of the original viewpoint without significant artifacts occurring. Mueller, et al. [94] modified the standard LDI approach to build "oriented" billboards. This allows a larger angular range (16°). This technique's effectiveness depends on the transfer function, the size of the slabs and the viewers distance. Figure 23 illustrates the basic ideas of LDIs and illustrates how they may be oriented.

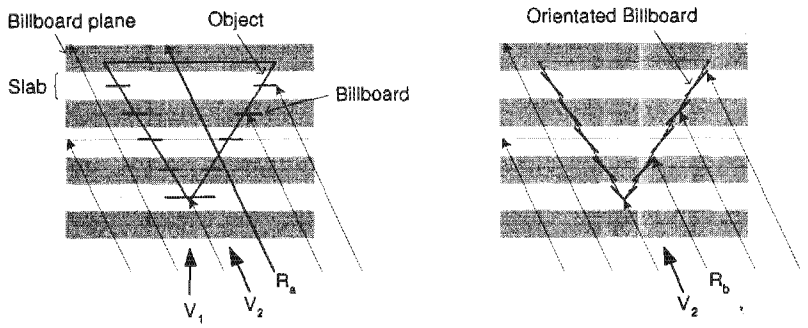


Figure 23: Image based rendering of volume data. The volume is divided into slabs. The slabs are represented using billboards, which are in turn used for rendering other viewpoints. Left: When the angular range between view points v_1 and v_2 becomes too large, artifacts occur as rays (R_a) pass between the billboards. Right: Oriented billboards reduce the errors (R_b is blocked). Note: the "object" as indicated does not actually exist as a surface, but is rather the surface implied by high opacity values of the classified volume.

This approach has been used to support interactive visualisations of very large volumes using widely distributed rendering [9, 54]. The volume was bricked and rendering prioritised such that the regions closest to the viewer were updated first.

Stereopsis

Stereopsis is seen as the PDC to provide unambiguous depth information. Stereo images are generated by rendering an image for each eye. In texture-based rendering, two separate images are generated, effectively halving the rendering rate. Techniques have been designed which use image-based rendering solutions to render the image pairs. The typical approaches are reprojection [3] and template based [71]. Reprojection uses either one image to generate the other or a between image to generate both. The template approach interleaves the rendering of each image, using the shared computations to reduce the overall cost of computing the two images. All the stereo image acceleration techniques have been implemented in software and only support parallel viewing. Perspective viewing is described as "too hard" [3, 71].

3.4.3 Discussion

All reproducible PDC enhancements have been implemented. However, it is seldom that multiple enhancement have been implemented in a single system. In light of the recent advances in graphics cards and their improved flexibility (programmability), many of the issues faced by the earlier implementers are disappearing. Ultimately, it will be possible to produce all the cues in one implementation. The cost of producing the cues will remain significant. In fact, the cost will increase as volume sizes increase. Thus the question shifts from how to produce an individual PDC to studying how the PDCs combine and which are important for specific tasks. Much of the volume rendering literature makes broad statements as to the effectiveness of particular PDC enhancements. However, at this time no systematic approach has been applied to test these statements.

3.5 Perceptual depth cues and X-ray visualisation

In this section we introduce X-rays and discuss X-ray visualisation from a PDC perspective.

3.5.1 X-ray generation

An X-ray source consists of a cathode and an anode with a high voltage between them. A heating element excites electrons on the cathode causing them to move towards the anode. As the electrons strike the anode, they release energy as heat (predominantly) and photons. The photons scatter in all directions – Figure 24. The X-ray source is surrounded by a Lead or Tungsten shield which prevents the photons from scattering into the operational environment. A narrow slit in the shielding creates a controlled beam of photons, commonly referred to as the X-ray beam.

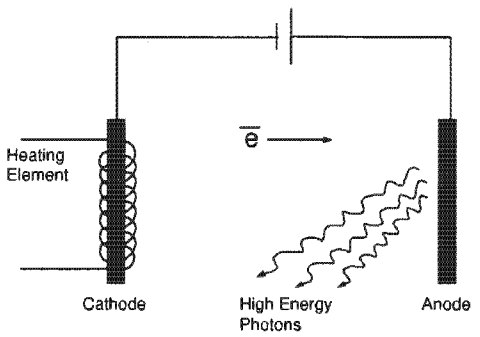


Figure 24: A schematic diagram of the internals of an X-ray source. Excited electrons pass from the cathode to the anode. On striking the cathode, high energy photons are released.

When the X-rays strike particles in foreign objects, they excite the particles. As the excited particles return to their initial, lower energy state they emit the gained energy as heat and as scattered, lower energy photons. The more dense the foreign object, the more likely the X-ray will be scattered by a particle and the energy absorbed. The amount of scattering can be used as an indication of the density of the object, which in turn characterises the materials of which the object is made. The X-rays which do not collide with any foreign objects finally strike an array of scintillating crystals, which converts the X-rays energy into light (either detectable or visible light). A light sensitive film captures the final image or, as in more modern systems, the light is converted to an electrical signal using an array of CCDs (Charge Couple Devices) and the signal processed digitally.

3.5.2 Perceptual depth cues in X-rays

Standard X-ray images are known as *shadow graphs*. Shadow graphs are difficult to interpret. The difficulties arise due to the projection of complex $3D$ structures onto a $2D$ image plane. Many of the PDCs used for normal image interpretation are not present. Those that are present are fully exploited and combined with any experience of the viewer.

Analyses of shadow graphs reveals several interesting facts concerning their interpretation:

- Shading has a small role in indicating the $3D$ nature of an objects's features, but provides no depth information.
- While occlusion does occur (objects of higher density obscure other objects) it provides little to no depth information.
- Relative object sizes and linear perspective provide limited depth information. However, the viewer needs to be familiar with the objects under inspection.
- The loss in sharpness of objects with distance from the detector creates an effect which can be likened to the atmospheric depth cue. This effect is unreliable as a depth cue as it is not consistent between machines and even scans on the same machine.

The two example X-rays shown in Figure 25 illustrate these points. The hand X-ray provides a relatively low complexity example, while the chest and shoulder X-ray illustrate how highly complex X-rays can be. The chest cavity contains multiple organs, each with their own local structure. When projected, the chest cavity appears as a maze of fine structures which requires a trained, experienced eye to interpret.

(a) (b)

Figure 25: Examples of X-ray images: a) A human hand. b) A shoulder joint and part of the chest cavity. The images have been inverted for printing.

3.5.3 Perceptual depth cue enhanced X-rays

It is impossible to add psychological PDCs to an X-ray without a reconstruction of the 3D structures. Reconstruction is done via CT and produces volumetric data. The only PDCs which can be used to enhance a shadow graph directly are motion cues, typically KDE, and stereopsis.

Stereoscopic radiography has been around for decades. In 1898, Davidson [21] is documented to have used stereoscopic X-rays for various studies. However, with the advent of CT and MRI, the use of stereoscopic X-rays faded. In more recent years radiography has turned to digital computers for capturing, storing and displaying the X-rays. Several hi-tech stereoscopic radiographic systems have since been reported in the literature [39, 53, 17, 143, 30, 31]. The newer systems offer a huge advantage over the old stereoscopic systems in the form of image enhancements. These enhancements may be applied to images interactively, allowing a larger range of the image data to become readably visible to the radiologists.

The studies that followed the first developments asked questions along lines of: "Is stereo viewing really better than conventional 2D Viewing?" [51, 53, 17]. The applications targeted mammography systems. Hsu et al. [53] used stereo imaging in mammography exams with the aim of improving the detection of malignant tissues. They compared standard

biplanar angiography scans (2 perpendicular view X-ray images) to a stereoscopic system and attempted to use the X-ray images to make direct quantitative measurements and reconstructions. A similar system was proposed by Chelberg, et al. [17] with the aim of measuring and locating tumours within breasts.

The general consensus is that stereoscopic displays are no worse than standard displays. In addition, stereopsis permits a better definition of "abnormal masses from similar surrounding normal tissues" and allowed "the observer to look through the clutter and focus on a point of interest"[31]. They found that stereopsis helped with identification of clusters and groups of tissues. Stereopsis is reported to be the most stable and reliable form of depth perception in shadow graphs — disambiguating and reinforcing the weaker PDCs. Stereopsis has been heralded in the past: "A stereo pair is especially advantageous where superposed structures form an intricate image"[115].

KDE and stereopsis have been used in radiography applications, unrelated to medicine, specifically the scanning of luggage and the inspection of electronic parts. These applications were examined by the 3D Image Research Group at Nottingham Trent University [120]. They implemented the KDE by capturing X-ray images from multiple viewpoints and displaying them as a rotating sequence. Their system is reported to improve the global depth perception of the inspected objects and, consequently, the online image. However, measurement and fine structure interpretation requires local depth perception and KDE may or may not hinder the process depending on the specifics of the task.

The multiple scans required to support KDE are time consuming, making the use of KDE ill suited in time critical environments, for example a trauma unit. The alternative of capturing multiple view points simultaneously is expensive to implement, requiring multiple sources and detectors. Another concern is the increase in the patients' radiation dosage. Advances in scanning technology are alleviating this problem by lowering the dosage required during a scan. Also, if the patient is having a LACT scan, several of the initial scans can be presented as an animation prior and during CT reconstruction, especially as the process is time consuming. The visualisation would give the viewer the opportunity to construct a mental model of the data and make him/herself aware of any reconstruction artifacts.

3.6 Studying selected perceptual depth cues

Given the limited scope of this work and the desire to compare the PDC enhancements across the visualisation systems, it was decided only the cues which could be implemented in both systems would be studied. Consequently, this study continues with the focus on

the use of stereopsis, motion cues (specifically KDE) and their combination.

The fact that specific cues cannot be easily isolated in a complex visualisation system implies that there will be interaction between the natural cues and the enhancements. The theory presented in this chapter will be needed to understand the results of the experiments presented in Chapter 6.

3.7 Summary

This chapter began with a review of the various depth cues and the models used to combine depth cues. Although cues are simple when considered individually, when combined they form complex interactions. The interactions are described effectively by the "unified model" of PDCs. Individual tasks should be analysed to ensure only dominant depth cues are implemented in a scientific visualisation. This facilitates accurate depth judgements, while minimising implementation cost. Many tasks may be grouped according to a set of elementary tasks. However, some tasks are extremely sensitive to the viewer's perception and consequently cause changes in the dominance of the cues.

The visualisation systems were then discussed from a PDC perspective. Voxel-based volume rendering was revisited. What PDCs are present in a basic volume rendering and what PDCs can be added were discussed. This was followed by a survey of existing work enhancing volume rendering with PDCs. Next, a description of how X-rays are formed was given. The information presented was used to assist in determining what PDCs already exist in X-rays and how the images can be enhanced using PDCs. On closer inspection, it was found that only stereopsis and motion cues are both applicable as enhancements. The use of these PDCs in past systems was then discussed.

Finally, the PDCs common to both visualisation system (stereopsis and KDE) were selected for further use in the remainder of this dissertation. This will allow for the comparison of both the volume and X-ray visualisation systems. In Chapter 6 the PDCs and their combinations will be evaluated and the systems will be compared. The findings will be explained using the presented theory.

Chapter 4

Stereoscopic depth

In the last chapter we came to the conclusion that KDE and stereopsis are the two PDCs that will be used in our visualisations. While supporting motion cues (through interactive manipulation and/or animation) requires the development of algorithms specific to a visualisation, stereopsis may be implemented relatively independently. Given the technical difficulties associated with producing an effective stereoscopic system, it is understandable that much attention has been paid to this topic here and in the literature. Stereoscopic display design consists of two tasks: composition, which describes how stereograms are constructed; and presentation, which ensures the left and right eye receives only the corresponding image.

This chapter covers: the concepts of stereopsis (Section 4.1), stereogram composition (Section 4.2), presentation solutions (Section 4.3) and software frameworks (Section 4.4). These techniques are used in the development of the visualisation systems presented in Chapter 5.

4.1 Concepts of stereopsis

The word stereopsis stems from the Greek word meaning *solid sight*[18]. It describes how humans see depth using disparities between the images formed on the retina of each eye. The two images are fused mentally to provide a single image with depth information. The perceived depth of a point depends on the position of the corresponding (*homologous*) points in each eye. The difference in the position is known as *disparity*.

Measuring the difference between retinal images is difficult. Consequently, when stereograms are discussed, the separation between the displayed points, known as *parallax*, is used as a metric. The parallax value characterises the depth of the perceived 3D point and

where it lies in the perceived viewing space. Table 1 summarises the four possible parallax cases. The vertical difference in the HVS is zero[1].

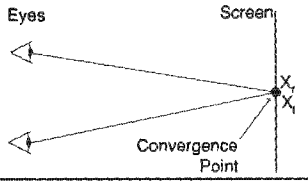
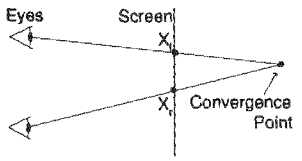
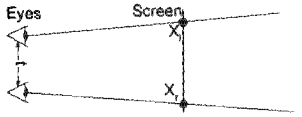
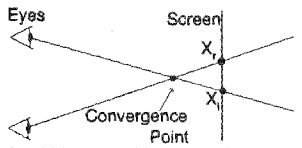
Parallax	Illustration	Equation	Description
Zero		$X_r - X_l = 0$	The eyes are converged, and the perceived point lies on the display surface.
Positive (uncrossed)		$X_r - X_l > 0$	The point lies behind display. The eyes tend towards parallel as the parallax increases. If the parallax is too great the display becomes uncomfortable to view.
Divergent		$X_r - X_l > t$	The parallax is greater than the eye separation. This is completely unnatural configuration and causes eye strain and headaches as the brain tries to converge the points.
Negative (crossed)		$X_r - X_l < 0$	The eyes are crossed. The point appears to lie in front of the display.

Table 1: Different parallax cases in stereo viewing. X_l and X_r describe the horizontal components of left and right homologous points. t is the distance between the viewers eyes

The amount of disparity that humans can cope with is limited. The limits are defined by a non-uniform region known as *Panum's Fusion Area* — illustrated in Figure 26. Beyond these limits the homologous points are no longer fused, but perceived as two distinct points. If a stereoscopic image contains disparities which cannot be fused, the stereoscopic effect fails and the viewer will see a double image (known as Diploplia).

[1]This can be verified by looking at a ruler held horizontally and then vertically, while closing alternatively the left and right eye — the ruler will only appear to move horizontally.

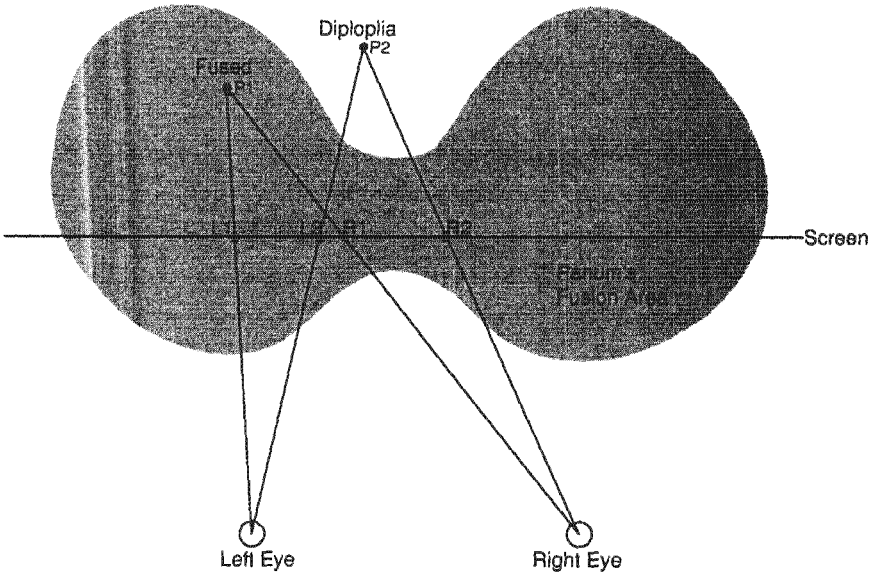


Figure 26: Panum's Fusion Area (grey) defines the limits at which the human visual system can fuse homologous points. As an example consider the x components of two homologous points: R_1,L_1 and R_2,L_2. R_1 and L_1 will be fused as P_1, while R_2 and L_2 cannot be fused and are perceived as two points rather than the desired point P_2.

The exact size and shape of the Panum's fusion area varies depending on: the display time, the size of the objects in the image and the distance between objects in the image. Panum's fusion area is generally very shallow, highlighting the short range of distances over which stereopsis is used by the HVS [99].

The maximum disparity with which the HVS can cope is proportional to the distance between the viewer and the display surface. As a result, parallax is usually stated in stereographics literature as a *visual angle*. The visual angle, β, is distance independent. Given a parallax value, P, and the associated viewer to screen distance, d, the visual angle is calculated as:

$$\beta = 2 \arctan \frac{P}{2d} \tag{5}$$

The HVS combines various PDCs to provide the final depth perception. In the HVS binocular disparity, accommodation and convergence are tightly coupled. In stereoscopic displays, a mismatch between accommodation and convergence occurs. When viewing a stereogram the eyes will change as if parts of the image are at different depths, while the screen and the entire image remains in focus. The vergence and disparity together convince

the mind the image contains depth. The focus, however, is incorrect, with the exception of points of zero parallax. Points of zero parallax have a disparity which places them on the display surface.

The set of all points with zero parallax forms a curve known as the *horopter*. As the convergence, accommodation and disparity are correct at the horopter, objects of interest should be centred on this curve. In addition, the HVS expects the depth to vary in a planar fashion [49]. Consequently, if the display is planar, the horopter and other curves of constant parallax should also be planar and parallel to the display surface. The limits of the HVS, the horopter shape and the positioning of the scene in the display space help select the stereoscopic parameters and indicate which system models are most effective.

The HVS is unaccustomed to accommodation and convergence operating out of synchronisation, resulting in fatigue which is experienced as eye-strain and headaches[27]. The conflict becomes more severe as the maximum parallax in a stereogram increases. The HVS can be taught to ignore the conflict through practice. Viewing a stereogram for 10–15 minutes a day for a week is reported to result in a dramatic improvement in depth acuity[58, 90]. The more the parallax varies, the stronger the depth effect. In an ideal system the maximum parallax should be chosen according the viewers "skill level". A warning needs to made: "if a system is used for prolonged periods, the eyes loose the ability refocus quickly"[135].

The maximum acceptable horizontal parallax values have been determined experimentally for an average user by Valyus [130] as $[-4.93°, 1.75°]$, however the range $[-1.5°, 1.5°]$ is the more common in stereographics literature [18, 48] and is sufficient for the purposes of this dissertation.

4.2 Stereogram composition

It is difficult to create a stereogram which is both visually correct and comfortable to view. The difficulty arises from the large number of variables in a stereoscopic system. A badly constructed stereogram will cause the viewer considerable eye strain, headaches and in extreme cases, nausea. Viewer discomfort will eventually lead to stereoscopic display system rejection [48]. Clear understanding of the system models and parameters are essential to avoid such results.

Stereoscopic systems are modeled using a dual camera model. There are two primary stereoscopic camera configurations: Cameras positioned by rotation and cameras positioned by horizontal translation.

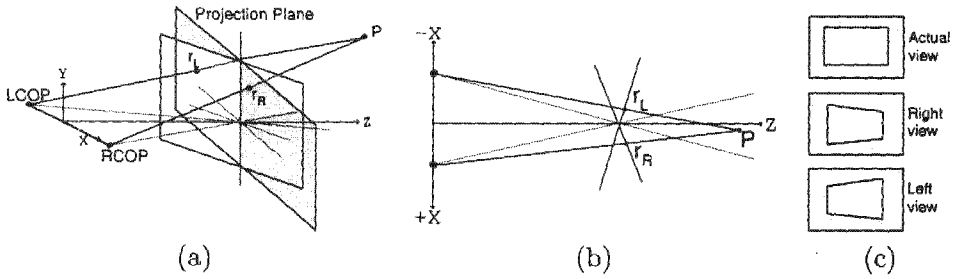


Figure 27: Toe-in stereo model: a) 3D representation of the camera configuration, b) Top view of the configuration, c) Key-Stone distortions in both left and right views.

4.2.1 Stereopsis by rotation

Stereopsis by rotation is commonly referred to as the "cross eyed projection method" or the *toe-in* method. It involves rotating the cameras around a reference point — illustrated in Figures 27a and 27b. This approach is fairly common as it is easily implemented and it has been used successfully in stereoscopic cinematography and photography.

The toe-in method produces poor stereograms under most circumstances. The problems are attributed directly to the technique's use of dual viewing axes. The dual rotated axes produce images which suffer from *Key Stone distortions*, a spatial distortion in which parts of the image are skewed — illustrated in Figure 27c.

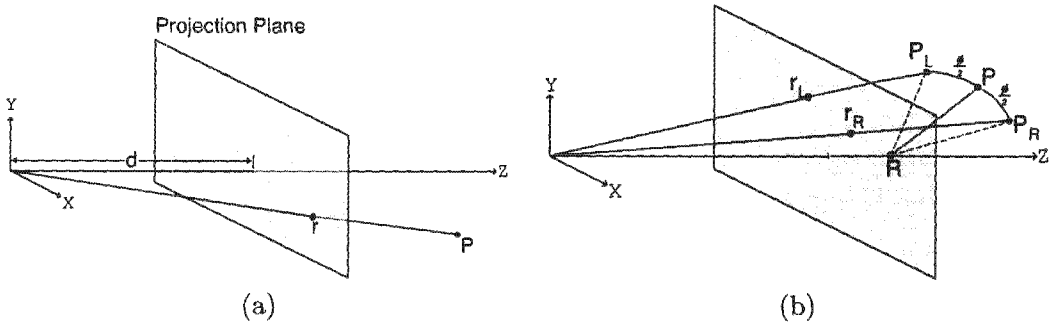


Figure 28: Projection models. a) Traditional perspective projection model, and b) Toe-in projection model. Rather than rotating the camera, the points have been rotated. The two camera planes lie coincident.

Each camera of the toe-in model uses a traditional perspective projection model [33] — illustrated in Figure 28a. This model describes how a $3D$ point $P = (x, y, z)$ is projected onto a viewing plane positioned d units from the camera with coordinates $r = (x, y)$. Mathematically, the projection is written as:

$$r_x = \frac{P_x d}{d + P_z}, \tag{6}$$

$$r_y = \frac{P_y d}{d + P_z} \tag{7}$$

In the toe-in model, the two cameras are rotated by ϕ around an axis parallel to the y axis, passing through the reference point R lying on the z axis. Equivalently the $3D$ points may be rotated in the opposite direction with respect to the same reference point, R, while the camera remains static. This is illustrated in Figure 28b, with the two planes coincident and a $3D$ point, P rotated to give the points P_L and P_R for the left and right cameras respectively. From this geometry the horizontal and vertical components of a 2D point, $r = (x, y)$, for a given camera are calculated by projecting the $3D$ point, P, as follows:

$$r_x = \frac{d(P_x \cos\frac{\phi}{2} - \alpha(P_z - R_z)\sin\frac{\phi}{2})}{(P_z - R_z)\cos\frac{\phi}{2} + \alpha P_x \sin\frac{\phi}{2} + R_z}, \tag{8}$$

$$r_y = \frac{dP_y}{(P_z - R_z)\cos\frac{\phi}{2} + \alpha P_x \sin\frac{\phi}{2} + R_z}, \tag{9}$$

where $\alpha = -1$ for the left projection, giving the projected point r_L, and $\alpha = 1$ for the right projection, giving the point r_R.

The rotation introduces subtle side effects, which can be seen when examining the stereo projections mathematically. Both the parallax in the image and the equation describing the horopter are affected. Equations of the parallax and the horopter are derived below, modified from [48]:

The parallax, D_p, is calculated as the difference between the components of the left and right homologous image points:

$$D_p = r_R - r_L \tag{10}$$

Substituting Equations 8 and 9 separately for the left and right views into Equation 10 results in the following equation for horizontal, H_p and vertical parallax, V_p:

$$H_p = \frac{-2d(P_x^2 \sin\frac{\phi}{2}\cos\frac{\phi}{2} + A(P_z - R_z)\sin\frac{\phi}{2})}{A^2 - P_x^2 \sin(\frac{\phi}{2}^2)}, \tag{11}$$

$$V_p = \frac{2dP_x P_y \sin\frac{\phi}{2}}{A^2 - P_x^2 \sin(\frac{\phi}{2})^2}, \tag{12}$$

$$\text{Where } A = (P_z - R_z)\cos\frac{\phi}{2} + R_z$$

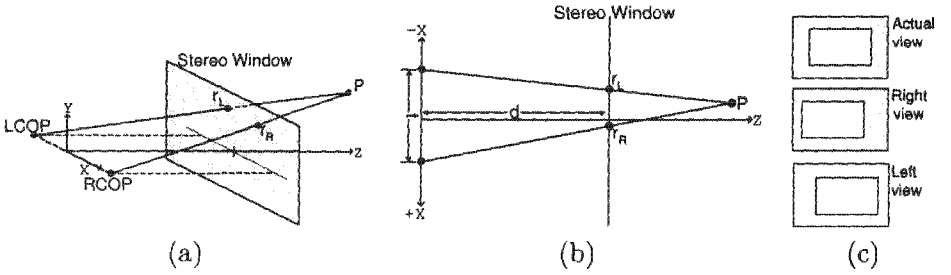


Figure 29: Horizontal Translation configuration. a) Diagram of the configuration, b) top down view of the configuration, c) projection of the scene from each camera position

While horizontal parallax is desirable, the introduction of vertical parallax ($V_p \neq 0$) is not. Vertical parallax is unnatural and results in images which are difficult to fuse into a single image.

Provided the amount of rotation is kept small and the vertical parallax produced is kept below 0.1°, the distortion becomes negligible[48]. This explains why the toe-in method has been used successfully in stereoscopic cinematography, where the separation between the cameras is relatively small compared to the scene being captured and the viewers eye separation is small compared to the distance to the display[119].

The Horopter and other curves of constant parallax value can be found by solving Equation 11 as $H_p = k$. k is the constant parallax value. For the Horopter $k = 0$, this yields the equation:

$$P_x^2 + (p_z - R_z + R_z \frac{\sin\frac{\phi}{2}}{\sin\phi})^2 = (\frac{R_z \sin\frac{\phi}{2}}{\sin\phi})^2$$

which describes a circle in the xz plane with centre $(0, 0, R_z(1 - \sin\frac{\phi}{2}/\sin\phi))$ and radius $R_z \sin\frac{\phi}{2}/\sin\phi$. This curve is undesirable for planar display as it conflicts with natural viewing which expects depth planes.

4.2.2 Stereopsis by horizontal translation

Stereopsis by horizontal translation produces "correct" stereograms for desktop displays. The two cameras have the same orientation, but are separated perpendicularly by a small distance, t, as shown in Figures 29a and 29b. For this reason, this model is also referred to as the "parallel axes projection method". Other common names for this technique are off-axis or on-axis projection, depending on the order of the translation and projection operations.

With the centre of projections of each camera positioned on either side of the z axis, separated by a distance t, a projected point, $r = (x, y)$, is calculated from the 3D point,

$P = (x, y, z)$, as:

$$r_x = \frac{P_x d + P_z \frac{\alpha t}{2}}{d + P_z}, \tag{13}$$

$$r_y = \frac{P_y d}{d + P_z} \tag{14}$$

where the left and right projection are specified by setting $\alpha = -1$ and $\alpha = 1$ respectively. It is clear that there is no vertical parallax, as the value for r_y does not change from the left to right projection. The horizontal parallax is given by the equation:

$$H_p = \frac{P_z t}{d + P_z} \tag{15}$$

$$= t(1 - \frac{d}{d + P_z}) \tag{16}$$

This equation produces parallax which is proportional to the eye separation and inversely proportional to the distance from the display. Additionally, as an object extends into the distance, $z \rightarrow \infty$, the parallax approaches the viewers eye separation, $H_p \rightarrow t$. The equation places bounds on the parallax in a natural manner.

If implemented directly, Equations 13 and 14 describe the off-axis system. Equation 13 can be manipulated to yield the following equivalent equation:

$$r_x = \frac{d(P_x - \alpha \frac{t}{2})}{d + P_z} + \alpha \frac{t}{2}. \tag{17}$$

This equation, combined with Equation 14, describes the on-axis projection. The standard perspective projection is all that is required and the implementation is described as:

1. A Translation of $\alpha \frac{t}{2}$ applied to the x component of all 3D points
2. A standard projection of the 3D points
3. A horizontal translation (pan) of $-\alpha \frac{t}{2}$, applied to the resulting image.

While Equations 13 and 17 are mathematically equivalent, the projection operation is not commutative in the discrete display space. As a result, the stereograms constructed for the same scene using on-axis and off-axis projection will be different. An effective means to understand the differences may found by examining the field of view (FOV) for each method — Figure 30a and 30b. In the on-axis method, the parts of the image lying on the edge in the direction of translation are discarded. Consequently, the final image

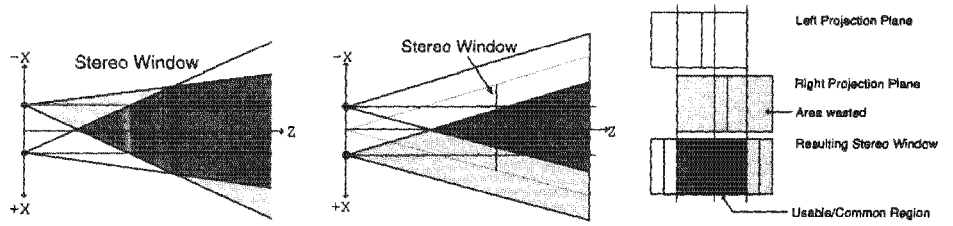


Figure 30: The field of view (FOV) for a) off-axis and b) on-axis projection. The darker region illustrates the effective FOV for the final stereogram. c) illustrates how much of the produced images are discarded in the on-axis method.

contains less information than either the left or right images — Figure 30c. The results are contradictory to natural viewing in which none of the additional information is discarded. When the reduced field of view is combined with the screen edges, a break down of the stereoscopic effect is likely. For this reason, the off-axis is considered more accurate and preferable.

It is customary in Computer Graphics to express projections as a matrix or a series of matrices. The following expresses the off-axis projection using a right hand system[2]:

$$S = T(-\vec{c}).T(\alpha(\frac{t}{2}), 0, 0).M_{per}.T(-\alpha(\frac{t}{2}), 0, 0).T(\vec{c}) \tag{18}$$

$\vec{c}$ specifies the centre of the object of interest relative to the origin. T and M_{per} are the standard 4×4 matrices for translation and perspective projection [33]. α parameterises the left and right projection matrices, -1 for left view projection and 1 for right view projection.

4.2.3 Stereogram parameters

The camera models include several parameters: eye separation (t), distance to the projection plane (d), FOV, and so on. Several of the system parameters are not captured by the camera models. These include ideal image contrast settings, position of the object in the FOV, viewers position and screen size. When values are chosen, the settings require several tradeoffs and continual tweaking for the best results. This process is considered a "black art" by some. To assist in choosing the values, it is worth considering the following [18]:

- The scene should have the deepest depth effect, while adhering to the parallax limits of the HVS. The eye separation and parallax values should be set based on the ranges of objects within the viewing frustum.

[2]In a right hand system, the matrix M would be applied to the vector $\vec{p}$ as $\vec{p}' = \vec{p} \cdot M$

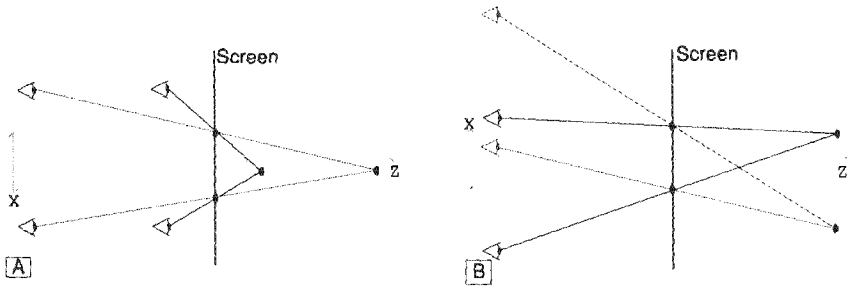


Figure 31: Depth perception and head position. a) Illustrates how, when the viewer moves closer to the display, the perceived depth range is squashed. b) Illustrates how movement parallel to the display shifts the perceived position

- Objects of interest should be centred on the horopter as this minimises the accommodation and convergence conflicts for these objects.

- The left and right image contents should match as closely as possible in brightness, focus and size in order to appear as natural as possible.

- Objects in the stereogram near the edges of the screen should not have negative parallax. Objects with negative parallax appear extruded from the display. If the object is cut by the screen edge (especially the horizontal edges), occlusion depth cues dominate and cause the stereoscopic effect to break down in a process known as *frame cancellation*[130]. It is preferable to reduce the depth effect and have only the positive parallax if frame cancelation is likely.

- Stereograms should be constructed for a particular viewer's position and display's properties. The display's properties, such as the resolution and screen size, are required to estimate the maximum parallax values correctly. The viewer's position effects the perceived depth — see Figure 31.

  Head tracking is used to ensure correct depth viewing in some systems [110]. However, head tracking introduces additional complexities and expenses. It can only be applied affectively in systems using synthetically generated stereograms or dynamically controllable stereocameras. Untracked head movement produces apparent movement or FOV shifts with parts of the image shifting artificially according to the parallax value. The HVS is robust enough to adjust, making much of the resulting distortion imperceptible after a few minutes [136].

Most of the parameters in stereoscopic systems are dependent on the viewing task and on user preferences. At first, one might consider using values which approximate real viewing for the default system values. Such *Orthostereoscopic* systems are often too restrictive, requiring the viewers eye separation and camera separation to match while the perspective constraint is maintained. The perspective constraint ensures that the normal image size, perspective and depth relationships are maintained. It is given by

$$V_d = mf,$$

where V_d is the viewing distance, f is the focal length[3] and m is the magnification factor. Southard [119] used this approach to define his stereoscopic system. An orthostereoscopic display is not useful for viewing scenes at either end of the scale spectrum — for example, visualising solar systems or atoms - as the depth effect is not noticeable at these scales.

A more flexible approach is to use experimentally determined values for some of the system parameters and from these calculate the unknowns. For example, consider, t, the eye separation parameter. If Equation 15 is rewritten to give t as a function of H_p, with H_p converted to the visual angle, the result is:

$$t = \frac{H_p}{1 - \frac{d}{d+z}} \tag{19}$$

$$= \frac{H_p(i+1)}{i}, \text{ where } i = \frac{z}{d} \tag{20}$$

$$= \frac{2d(i+1)\tan\frac{\beta}{2}}{i} \tag{21}$$

i may be interpreted as the depth of the scene in units of the viewing distance, d (An alternate derivation of t is given in [18]). If we substitute an acceptable value for the maximum parallax ($\beta = 1.5$) into Equation 21 along with the maximum and minimum z values for the scene, we obtain a suitable value for the eye separation. If the z values for the scene cannot be determined, an alternative is to use the approximation $t = 0.028d$, which is common in stereographics literature.

The equation for maximum parallax, P_m:

$$P_m = mft(\frac{1}{d_o} - \frac{1}{d}),$$

taken from [18], relates t to other system parameters. f is the focal length, related to the FOV. m is the magnification, related to the screen size. d_o is the distance from the

[3]Focal length and FOV are closely related, changes to the one, will force changes in the other.

object of interest to the Horopter and d is the viewing distance. The equation indicates that as t decreases, so the focal length should decrease, highlighting the link between the linear perspective PDCs and stereoscopic PDCs.

4.2.4 Other camera models

It is possible to use orthogonal projections for creating stereograms. However, the orthogonal projection is completely unnatural and should never be used when viewing "familiar" scenes. These stereograms cannot be created by horizontal image translation as the two images will never converge. Instead, the toe-in configuration must be used with a recommended maximum rotation of 4°. Alternatively, a shear operation along the x axis could be applied to the projected points with the magnitude of the shear dependent on the z value.

In general, orthogonal projections should be avoided as they are prone to reverse and unbounded parallax problems, failing Emmert's Law (refer to 3.2.1). In other words, the further away an object is from the viewer, the greater the disparity which makes the final stereogram difficult to fuse. The failure of Emmert's Law reflects heavily on the poor depth information provided by such systems.

4.3 Presentation techniques

It is possible to view stereograms with the naked eye, using either cross eyed or parallel eyed viewing (focusing on a point at infinity). It cannot be achieved easily by a layman as it requires the eye muscles to be relaxed in an unnatural fashion. The alternative is to use optical hardware which ensures only the correct image reaches the corresponding eye. The hardware can broadly be divided into two categories: time multiplexed and time parallel [43]. Time multiplexed displays provide images to left and right eyes alternately, whereas time parallel displays provide images to both eyes simultaneously.

4.3.1 Time multiplexed displays

Time multiplexed displays are usually based on polarisation schemes. These techniques allow a single display device to be used as a stereo display with obvious cost benefits. The low cost has lead to the popularity of these systems in consumer markets. The two most common schemes are:

- Shutter glasses

Current shutter glasses (Figure 32b) are made from liquid crystal filters. These may be polarised to be either opaque or transparent. The light to each eye is blocked alternately by polarising the appropriate filter. The polarisation changes are synchronised with the monitor's refresh period. The first major commercial shutter glasses were made by StereoGraphics and are called CrystalEyes. CrystalEyes use an infrared emitter and receiver to synchronise the glasses with the display and have been deployed successfully in several environments including CAVES [19]. Current consumer hardware (ATI and Nvidia video cards for example) include support for cheaper stereo glasses. These glasses plug directly into the display card and, as a result, limit the wearer's movement.

- Polarised screens

  Illustrated in Figure 32a — An LCD panel is placed in front of the display. The panel polarises the light passing through it using one of two orthogonal lenses. The viewer wears a pair of passive glasses with matching orthogonally polarised lenses. This approach is more suitable for a larger audience, as no synchronisation mechanism is required. Orthogonally circularly polarised glasses are used to reduce the effect of head movement on the polarising action of the glasses.

A projector can be used rather than a monitor to provide a larger viewing surface. In the polarising screen approach, the screen is replaced with a polarising lens placed in front of the projector. The projection screens must be made from materials that will not depolarise the light.

Time multiplexed displays suffer from *ghosting* (also known as *interocular cross talk*). Ghosting is undesirable as it causes viewer distraction and eye fatigue [52]. It occurs when the image destined for one eye can be seen by the other. Ghosting has several sources: imperfections in the polarisation/shuttering; slow screen phosphor decay, which is aggravated

Figure 32: Photos of typical time multiplexed presentation hardware. a) a screen filter placed in front of the display polarises the light, which is viewed through corresponding polarised glasses. b) Shutter glasses which alternatively block the displayed image from each eye.

by high refresh rates; monitor brightness (monitors should be as bright as possible, but the contrast setting of the image and the screen should be as low as possible); the resolution; the display techniques; the angle of the viewer to the display (the steeper the angle the worse the cross talk); and finally the parallax (the larger the parallax the greater the chance of ghosting [13]).

Image processing techniques have been used to reduce ghosting. One such technique approximates the cross talk and attempts to correct for it via image colour subtraction [84]. Otherwise, ghosting may be reduced by making various tradeoffs. For example: low persistence phosphor monitors may be used at a higher cost, or the image's contrast may be lowered. Lowering the contrast is not ideal as it requires a trade off with stereo acuity. Stereo acuity is related to visual acuity and both are proportional to the retinal illumination. Webber's Law, which deals with the detection of objects over a range of intensities, indicates that to improve acuity, the luminance should be as high as possible.

Time multiplexing implies that each eye receives half the normal light level. A perceived reduction in the image's brightness occurs as a consequence of the temporal integration done by the HVS [52].

The field switching does not have a negligible time and is perceived as a flicker. To prevent visible flickering, the polarising field and monitor must be synchronised and run at high refresh rates (twice that of normal monitor refresh rate). The minimum refresh rate for a monitor viewed under normal conditions is 60Hz, seeing that this is halved due to the multiplexing, the minimum refresh rate for the stereo setup is 120Hz. As higher refresh rates lead to less noticeable flicker, it is fortunate that today's monitors can be pushed to achieve frequencies of just above 150Hz, which results in a reasonable 75Hz refresh rate for each eye. Flickering is worsened when the viewing environment contains fluorescent light sources as aliasing occurs.

Over the past few years several manufactures have produced shutter glasses and, as a result, there exists a host of incompatible hardware/software solutions. Fortunately, de-facto standards are emerging.

4.3.2 Time parallel displays

Time parallel displays include a much larger category of devices and techniques, most of which use optical separation to achieve stereoscopic viewing. The earliest systems used mirrors and prisms to provide each eye with an image. Wheatstone's stereoscope, developed in 1838 [12], and the View Master stereoscope (1939) are examples of such systems[4]. Some

[4]The View Master was sold as child's toy and made stereo viewing popular amongst the general public.

of the more recent techniques are described.

Anaglyphs are a cheap stereoscopic solution, which has been available for several years. The left and right images are tinted with a colour and blended together. To view an anaglyph, a pair of cardboard glasses with the two coloured, plastic filters is worn. The colours used for the filters match the tint colours and are usually chosen to be reproducible using primary colours. For example, on monitors red/blue/green is used, while in printing cyan/magenta is more popular (primary printing pigments)[33]. Choosing a colour which can be reproduced without mixing reduces colour bleeding and ghosting. The defacto standard is to use red/blue filters with red for the left eye. Interestingly, placing the red filter in front of right eye tends to be more effective in general [72] and the human visual system has a low response to blue [33] making red/green seem the better option. Perhaps the justification for using red/blue over red/green are the issues with the longer persistence of green on the display phosphors. This justification only holds if the images are changing rapidly.

Traditionally, anaglyphs are constructed by converting the original images to grayscale. The grayscale images are then filtered with an appropriate colour mask before being combined into a single image. Newer techniques use "full colour" anaglyphs. However, the

Figure 33: Photos of time parallel stereoscopic presentation hardware. a) The Sythagram from StereoGraphics. b) A dual polarised projector setup. c) The ViewMaster. d) An anaglyph and 2 pairs of glasses. e) 2 Chromostereoscopic images and glasses

colour content of the stereogram has to be selected carefully to avoid ghosting and flattening of the image. Flattening occurs when an object, which should appear in both eyes, only appears in one due to its colouring. These problems may be reduced by lowering the saturation of the images using a gamma correction filter. It is common to find full colour anaglyphs made using red/cyan filters.

Chromostereopsis is a relatively new presentation technique. A single pair of glasses and one image is all that is needed. The glasses contains lenses with "ultra-precise micro optics" [122], which are essentially a set of diffraction lenses. The light striking the lens is diffracted by an amount proportional to the light's wavelength. So red, at the one extreme, is diffracted the most, whereas blue is diffracted the least. The lenses are arranged to diffract the light toward the viewer's nose. The left and right eyes receive light diffracted to different retinal positions, effectively creating a stereoscopic image. Figure 34 illustrates how the glasses create the depth effect. Images generated using chromostereopsis can be viewed sensibly without glasses as only a single colour image is used. For the same reason, chromostereopsis can be faithfully reproduced in printed media. Unfortunately, the technique cannot be applied to pre-existing stereograms as the image is pseudo-coloured according to depth. In addition, the pseudo-colouring eliminates the possibility of encoding additional information as colour. The glasses are relatively cheap, as the optics technology is well suited to mass production.

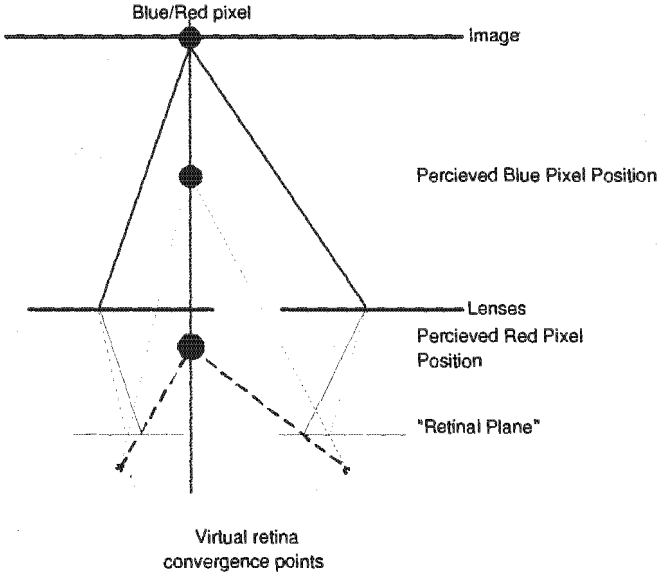


Figure 34: Chromostereopis creates a stereo image by diffracting parts of the image more or less depending on its colour. Red objects appear closest to the viewer and blue furthest

Projector-based displays are very popular, offering a large viewing area which may be watched comfortably by several people simultaneously. Two projectors are fitted with orthogonal polarising filters, while the viewers wear a pair of passively polarised glasses in a similar fashion to the time-multiplexed projector solution.

Head mounted displays (HMD) are a costly alternative with a wide FOV and a relatively low resolution. HMD's provide an immersive display which is worn. The weight of these displays is not negligible. They tend to restrict the users motions and require head tracking.

Several emerging technologies attempt to provide *autostereoscopy*. Autostereoscopic systems attempt to display stereograms without requiring the viewer to wear any glasses. These systems typically use lenticular screens or a tilton/barrier arrangement.

Lenticular screens have been used in the recently released SynthaGram^{TM}from Stereographics and the D4D technology from the University of Dresden [1]. The lenticular screen ensures each eye receives a unique view. The SynthaGram^{TM}displays several images simultaneously, with each pixel consisting of several subpixels - extracted from several images of the scene. This multiple image approach succeeds in allowing several head positions and viewers. However, it is processor intensive and not yet suitable for real time graphics[85]. The D4D system just displays 2 images, requiring the user's head to be in a fixed position. The viewer's eyes are tracked to ensure correct viewing. The display is suitable for a single user only.

Parallax barrier systems consist of a vertically slitted plate placed in front of a column interlaced image. The angle between the viewer's eyes and plate ensures the correct eye receives the correct image. The viewing position is limited to a narrow band in front of the slit and the final image resolution is effectively halved[56]. The Tilton [127], or moving slit, periodically moves the barrier left and right. The images are structured such that only the correct image can be seen[5][64].

4.4 Software frameworks

This section follows on from the theoretical and hardware aspects of stereoscopic systems by considering the software aspects such as libraries and interfaces.

4.4.1 Stereo software libraries

Stereoscopic applications can be described as belonging to one of four generations:

[5]The Tilton is effectively a time multiplexing approach.

1^{st} Generation

1^{st} generation stereoscopic applications detect and control the hardware directly (known as "raw control"). Applications written in this manner are bound to original stereoscopic configurations. The code is cluttered with hardware control details such as timing, page flipping, vertical retrace detection and synchronisation signal creation.

2^{nd} Generation

The lower level hardware control details are encapsulated by a set of drivers and libraries. The application has to identify the underlying hardware and notify the stereoscopic library through an API to load the correct drivers. The application then typically queries the drivers for hardware limitations. The individual left/right images are sent to the API for correct displaying. How the images are generated is left up to the application. A traditional graphics rendering engine may be used to render each image, using either the off-axis or on-axis method, depending on the available resources. Some custom graphics engines have been developed to take advantage of stereoscopic coherence and invariants.

The most common acceleration techniques avoid recalculating Equation 14 and parts of Equation 13 for both left and right images. Adelson and Hodges [2] extended this by using the invariants found in normal equations to reduce the number of backface culling and lighting calculations. An alternative acceleration technique uses the image generated for one eye and warps it to the view of the other, followed by patching to remove any artefacts.

An example of a popular 2^{nd} generation system is the WINx3D driver system[6] for the Windows^{TM}Operating Systems, which is used in combination with the DirectX, specifically the DirectDraw graphics APIs.

3^{rd} Generation

The application is built for systems with display drivers which support a stereo-capable graphics subsystem. The application typically queries the display driver to check for the existence of stereo capabilities. If available, the application renders the left-right images to pre-allocated video memory (buffers). The display driver handles the stereoscopic hardware control and produces the final stereoscopic output.

An example is an Open-GL application with stereo support. The application renders the left and right buffers using an appropriately specified projection matrix (an off-axis

[6] http://www.win3d.com

projection matrix is the most commonly used). Sample code for a complete stereo enabled application is available from [11].

4^{th} Generation

Newer stereoscopic display drivers support "OverDrive". In these systems, the driver detects when a 3D application is running in fullscreen and automatically generates a stereoscopic display from the 3D data the application passes to it. The application needs no stereoscopic specific code, however, the fullscreen requirement is occasionally inconvenient as standard interface components (for example menus and dialogs) cannot be used. In addition, debugging must be done remotely.

The stereoscopic mode may be toggled on or off at the user's behest and the parameters for the stereoscopic system are specified by setting the driver's runtime variables. This approach is of no use when it is necessary to display a pre-generated stereo image pair, as the actual 3D data is required. Many of the efficiency benefits found in 2^{nd} generation applications may be implemented within the graphics driver.

Nvidia recently released a stereo-enabled overdrive driver for Open GL and DirectX. For more information refer to [96].

4.4.2 $3D$-Cursors

$3D$-cursors are a natural extension of $2D$ cursors. When combined with a stereoscopic display, they allow direct $3D$ manipulation and measurement of $3D$ objects.

Cursor shapes

There are various common and uncommon shapes for $3D$-cursors, several of which are illustrated in Figure 35.

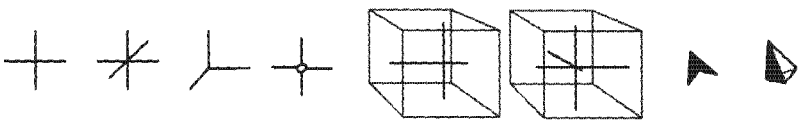


Figure 35: $3D$-cursor shapes. From left to right: crosshair, jack, tri-axis, crosshair with centre circle, full space crosshair, full space jack, $2D$ arrow, and tetrahedron.

$3D$-cursors need not have components in the z direction. Such cursors are still referred to as $2D$, even though they can be used to indicate $3D$ positions. Most of the cursor shapes are intuitive. A *jack* cursor is a cross hair with additional lines to represent the direction

of the z axis. The *tri-axis* cursor is half a jack with axes indicators on either positive or negative axis only. The *full space* variants of the cursors extend the axes lines to stretch across the entire viewing frustum.

The choice of cursor shape is not as simple as it may appear. Barham and McAllister [7] compared the use of several $2D$ and $3D$ cursors, in terms of accuracy, for use in selecting and moving control points of a spline. They recommend the use of a full space jack with a centre circle. The circle ensures that the horizontal and vertical lines are perceived as joined. Without it the lines appear to lie at different depths. They reported little performance differences between $2D$ and $3D$ cursors. The z lines of the $3D$ cursors tend to extend over a large range of depths making them difficult to fuse and consequently lower the expected performance. They also commented that the cursor and point marker lines should be thick enough to be clear, but not obscure the image and should have contrasting colours, making them easily distinguishable from the image.

Evidence suggests that the optimal cursor shape depends on the task and available PDCs. Goodsitt et al. [39] used a comb-shaped cursor with a horizontal slot, through which the small target objects could pass as the cursor's depth was varied. This cursor was tailored specifically for depth measurement of small targets in stereo-mammographs and proved more accurate than the cross shaped cursors. Additionally, the larger number of vertical lines assisted depth perception by emphasising the binocular disparity. For the tasks set by Barham and McAllister, stereopsis and linear perspective were used to rapidly position the cursor, while occlusion cues where used for finer adjustments.

Accuracy

Depth acuity is "exquisitely sensitive and a skilled operator can place a cursor to within 1 pixel"[18] of the desired target. Accuracies in the range of 1–2 mm have been recorded [39] using a crosshair $3D$-cursor. When considering accuracy, one needs to be aware of the inherent limitations of stereoscopic systems.

The first limitation is related to the resolution of the image and sampling errors. Figure 36a illustrates how the stereoscopic viewing volume sampling is affected by the resolution of the stereogram. It is clear from the figure that positive parallax is sampled more coarsely. The coarse sampling introduces errors in the disparity, projected location and size of the imaged objects. Objects positioned at the same depth, but different positions from the lines of sight, will appear at slightly different depths. The differences in position and size occasionally produce conflicting depth information. Consequently, the perceived depth in small regions of an image may shift randomly. Figure 36b illustrates how the intended

depth is affected by the coarse sampling.

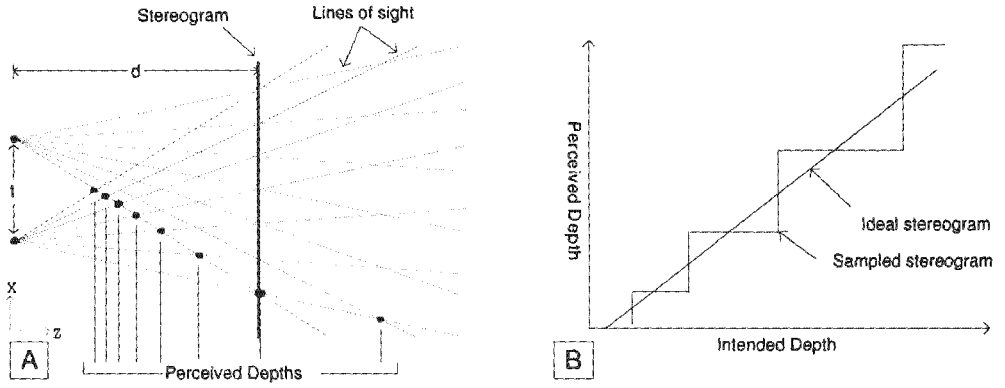


Figure 36: Depth accuracy in stereograms. a) The stereo viewing volume's sampling rate depends on the resolution of the stereogram. The lines of sight from each eye pass through the stereogram pixels. The intersection points of each line of sight are the sampling points of the stereo viewing volume. The sample points along one line of sight are shown. b) The sampling affects the accuracy of the perceived depth. A graph of perceived depth verses intended depth is given for the the ideally sampled and coarsely sampled stereogram. Images modified from [22]

Ledley and Frye [76] reported that the stereo acuity, with a maximum visual angle of $|\beta| = 1.5$, produces approximately 90 detectable steps in depth. When used with additional PDCs this value may increase. Stereo depth and perceived depth are not always equal, the perceived depth is dependent on other available PDCs.

The second limitation is related to the fusing of the stereo pair. Even in a perfectly sampled stereogram, there would be limited accuracy. Slightly different edges are fused from the left and right images — illustrated in Figure 37, with the gray region magnified on the right.

From this cross sectional view it is clear that the left eye sees the edge indicated by the point P_L, while the right eye sees P_R. L and R are the lines of sight from the left and right eye respectively. The perceived point is given by P_p. The difference between the actual object (the curve) and P_p is unavoidable and produces an error in depth, e_d, and boundary position, e_b. The equation characterising the boundary and depth errors are [83]:

$$e_d = r(1 - \frac{\sec\frac{\alpha}{2}}{d}r) \quad (22)$$

$$e_b = r(\sec\frac{\alpha}{2} - 1) \quad (23)$$

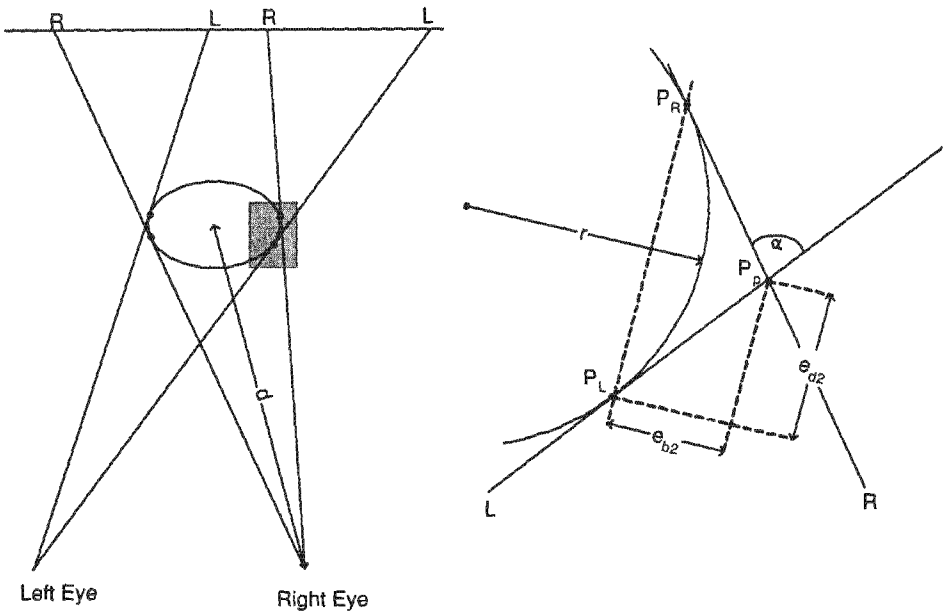


Figure 37: Boundary and depth errors inherent in a stereogram. If an object is represented locally by a sphere with radius r and distance d from an eye, then we can expect errors e_b and e_d for the boundary and depth perception of the perceived edge P_p. P_p is perceived as a result of converging edges P_R and P_L.

d is the distance from an eye view to the centre of the sphere approximating the objects local curvature, with r being the radius of the object. α is the angle between the lines of sight, L and R, passing tangentially to the object. These equations indicate that as the distance between the eyes and consequently, the angle α increase, so the expected error increases.

Beyond accuracy, there is a limit on how rapidly a 3D cursor can be positioned. It may take a viewer several milliseconds to fuse a stereograms, especially if the user is inexperienced at stereo viewing. In addition, if the scene contains rapidly moving objects, the amount of time needed to fuse the scene extends and fusion becomes more difficult [7]. Anu [4] reported that one second should be sufficient for the best accuracy, providing an upper limit.

4.5 Summary

This chapter presented the information required to implement and configure a stereoscopic system, including a detailed overview of several stereoscopic models, guidelines for composing stereograms and methods for calculating default values for system parameters. The available presentation systems were introduced. This was followed by a description of the available software frameworks, including a discussion on $3D$-cursors and the accuracy limitations of stereoscopic systems. In the next chapter the prototype systems developed are discussed. The information discussed in this chapter was used to build the stereoscopic displays for each of the visualisations. In addition, it is useful to consider the limitations of a stereoscopic system when examining the experiment results presented in Chapter 6.

Chapter 5

Prototype systems

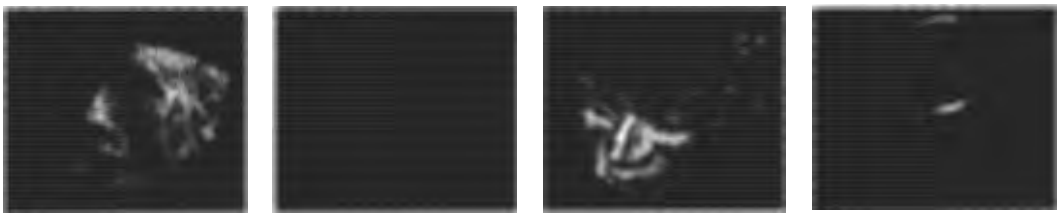
In the previous chapters both X-ray and volume visualisation systems have been examined with respect to perceptual depth cues. It was discovered that stereopsis and motion cues (KDE) were the only candidate PDCs which could be applied to *both* visualisation systems. The principles and practices of producing and presenting stereoscopic images were then described. In this chapter the initial prototypes and the paths taken to reach the final systems are described. The final prototypes are discussed in detail.

In Section 5.1 the preliminary developments of the volume visualisation system, the X-ray visualisation system and the stereoscopic display are described. In Section 5.2 the final volume rendering system is looked at, while, Sections 5.2.1–5.2.5 focus on the developed volume rendering library and the characteristics of the output. Section 5.2.5 describes a novel chromostereopsis extension to standard volume rendering. Section 5.2.6 discusses the volume rendering experimental application based on the presented library.

In Section 5.3 the final stereo X-ray system is described. In Section 5.3.1, how stereo X-rays can be captured using a fan beam X-ray scanner is investigated. Section 5.3.2 focuses the discussion on the LODOX machine and a model is developed to describe the machine's projection. A description of how the capture parameters can be chosen, based on the projection model, to capture an optimal stereo pair is included. Finally, in Section 5.3.3 the X-ray viewer software is presented and the various aspects which went into its design are described.

5.1 Initial prototypes

This chapter deals mostly with the development of final prototype systems. However, this section discusses the avenues explored and systems developed in reaching these final



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

Figure 38: A comparison of surface and voxel-based rendering for LACT and SACT data. a) Marching Cubes rendering, and b) volume rendering of LACT data. c) Marching Cubes rendering, and d) volume rendering of SACT data. Notice how the reconstruction artifacts appear as "noisy surface speckled with holes. In addition, the surface based rendering shows well defined surfaces where there are none. The LACT data was captured over 90 degrees using 31 projections. The SACT data uses 62 projections over 180 degrees. Refer to Figure 1 for a cross-section of each volume.

prototypes.

5.1.1 Volume visualisation

The first tool developed was an Open-GL texture-based slice viewer. It supported interactive selection of the slice axis and position for standard slice orientation and multi planer reprojection (MPR).

The Marching Cubes algorithm was implemented and used to explore the isosurface volume visualisation techniques. Open-GL was used for interactive isosurface rendering. The isosurface techniques are attractive in that a user only has one main parameter (the isovalue) to manipulate [33] and the results can be displayed using commodity video cards. From the initial developments it became evident that motions cues provide a strong sense of the depth. However, the extraction process is time consuming, and consequently motion cues could not be used during this phase. Several data sets were used to test the calibre of the output from the Marching Cubes implementation. The LACT and SACT data sets were handled poorly. The isosurfaces produced from these data sets were disjoint. When rendered, the smaller clusters of isosurfaces appeared as noise. While the large clusters appeared as well defined surfaces, which were pocked with with holes — See Figure 38a and 38c.

The possibility of using motions cues to allow the viewer to "see through" the clutter of reconstruction artifacts was explored. In order to do this a view dependent, interactive isosurface selection system was developed [87]. This system used a per view position preprocess to extract the visible isosurface information. A detailed description of the developed

system has been given in Appendix B. While the technique certainly does allow "isosurface browsing" and provides the viewer with the ability to "filter" out the reconstruction artifacts, it cannot convey the "uncertainty" in the reconstruction.

Voxel-based visualisation techniques do not suffer from the problems faced by isosurface techniques. In agreement with Kindlmann and Durkin [68], it was found that voxel-based rendering is particularly effective at conveying "surface quality or measurement of uncertainty". Amorphous and translucent features, which cannot be represented effectively by sharp surfaces, were handled well by the voxel-based rendering. The soft blending of regions in the volume rendering indicated a mixture or slow change in the underlying materials. For example, Figure 38 has a comparison of the voxel and surface-based rendering of a LACT and SACT data.

Finding a suitable transfer function for LACT/SACT data is difficult. The automatic and semi-automatic techniques, discussed in Chapter 2, are not well suited to handle the reconstruction artifacts. After testing and considering the options a traditional, manual specification was selected.

Having chosen voxel-based visualisation as the most suitable for the data, several renderings techniques were evaluated. Initially splatting, ray casting and shear warp implementations were tested in terms of their run time on the target platform. (Real time rates are essential for the support of motion cues.) The test machine was a 900 MHz AMD Duron, with 512MB of memory and an Nvidia graphics card. The VTK (Visualisation Toolkit[57]) implementations of splatting and ray casting were used, while the shear-warp implementation was taken from VolPack [74]. It was apparent from the evaluation that these software-based techniques were too slow for the required needs when viewing moderately sized volumes (256^3 and 512^3).

After examining the available hardware accelerated techniques, it was decided that general purpose graphics accelerators in combination with 2D texture based techniques would be used. At the time, 3D texturing was not supported by the graphics cards made available. The choice was additionally constrained by the high cost and unavailability of the dedicated hardware. Although, the availability of graphics cards with 3D texture mapping hardware has grown since then, the 2D texture mapping hardware is still prevalent.

5.1.2 X-ray visualisation

The initial X-ray visualisation prototype used a WAP volume rendering to generate model X-ray images. An animation was built from a sequence of these simulated X-rays. Although the animation sequence provided a strong sense of depth and the underlying structure, there

was a noticeable ambiguity. The ambiguity manifested itself in terms of the orientation of the object, which would toggle randomly between front facing and back facing. The ambiguity was removed by adding stereoscopic cues. OpenGL's stereo capabilities were used to extend the system (a 3^{rd} generation framework).

The next phase was the development of the final prototype system which dealt with real X-rays. The final prototype has been described in Section 5.3. Interestingly, the real X-rays did not exhibit the ambiguity to the same extent as the WAP images. The ambiguity was only noticeable when focusing on specific small regions. This indicated that an X-ray contains more depth information than the WAP images. This agrees with the analysis given in Chapter 3.

5.1.3 Stereoscopic tools

To investigate stereoscopic techniques a small 3^{rd} generation stereoscopic application was developed. It was written in Open-GL using the quad buffer stereo support. A pair of Crystal Eyes shutter glasses and a pair of anaglyph glasses were used. The anaglyphs offer a poor alternative to the shutter glasses. This test system supported the selection of several stereo camera models and $3D$ cursors. Many of the concepts presented in Chapter 4 were tested with this stereoscopic prototype. Several informal user tests were done to verify the system parameters and software choices.

Additionally, the use of chromostereopsis in volume rendering was explored. This application of chromostereopsis is relatively unique as it has only ever been applied to polygonal models in the past. Chromostereoscopic volume rendering is discussed in detail in Section 5.2.5. Unfortunately, chromostereopsis cannot be applied to X-ray visualisation as the depth of the image is unknown.

The final prototypes were written as a 2^{nd} and a 4^{th} generation stereoscopic applications. The alternate software solutions were adopted due to changes in the hardware available at the time (specifically the lack of native Open-GL stereo support in the replacement hardware). The X-ray visualisation system uses the 2^{nd} generation approach, which offers the flexibility needed for a completely custom stereoscopic application. The volume visualisation application uses the 4^{th} generation approach. This allowed the volume renderer to be developed quickly and relatively independently of stereoscopic system.

The display techniques which were used are all relatively intrusive. During the exploratory phase a HMD was tested, but it was found that it was ill suited due to its immersive and exocentric nature. The lowest cost, least intrusive approach would be to use projectors in combination with low profile, passively polarised glasses. This approach

would have the additional benefits of a larger display surface and support for multi-viewer collaboration.

5.2 Volume visualisation

The final volume visualisation system consists of two components. The first component is a library with the necessary application support structures. The second component is the actual experimental application.

The volume rendering library is based on the Open-GL API. While the core of Open-GL is platform independent, the extensions are vendor (manufacturer) specific. This poses hardware compatibility issues if the extensions are used. The volume renderer presented here works on any Open-GL implementation supporting version 1.3 and later.

5.2.1 Library structure

The library structure is given in Figure 39. The library can be divided logically into 3 main sections. The sections deal with the manipulation, storage and display of the volumetric data.

5.2.2 Volumetric data storage

The data storage classes take care of loading and saving the volumetric data stored in various formats. Access and control of the data is done through a common interface (*VolumeData*). In this way the format specific details are hidden. The *MemoryData* class was designed to hold run time generated data, and it serves to unify the memory allocation process with that of the data storage.

5.2.3 Volumetric data processing

Before volumetric data can be rendered it must be prepared. This need arises from the constraints Open-GL places on texture dimensions and formats. Open-GL restricts texture dimensions to powers of two. (This is due to efficiency concerns.) In other words, a 2D texture must have dimensions $2^n \times 2^m$ where $n, m \in \mathbb{N}$. There are several options for adapting non-compliant data: cropping, padding, resampling and bricking.

Cropping and resampling potentially result in data loss, while padding and bricking do not. However, padding the data results in larger memory requirements. Consider, for example, a volume with dimensions $100 \times 100 \times 130$. The padded dimensions would be $128 \times 128 \times 255$, which uses 3.2 times the memory. Bricking can be used to alleviates this

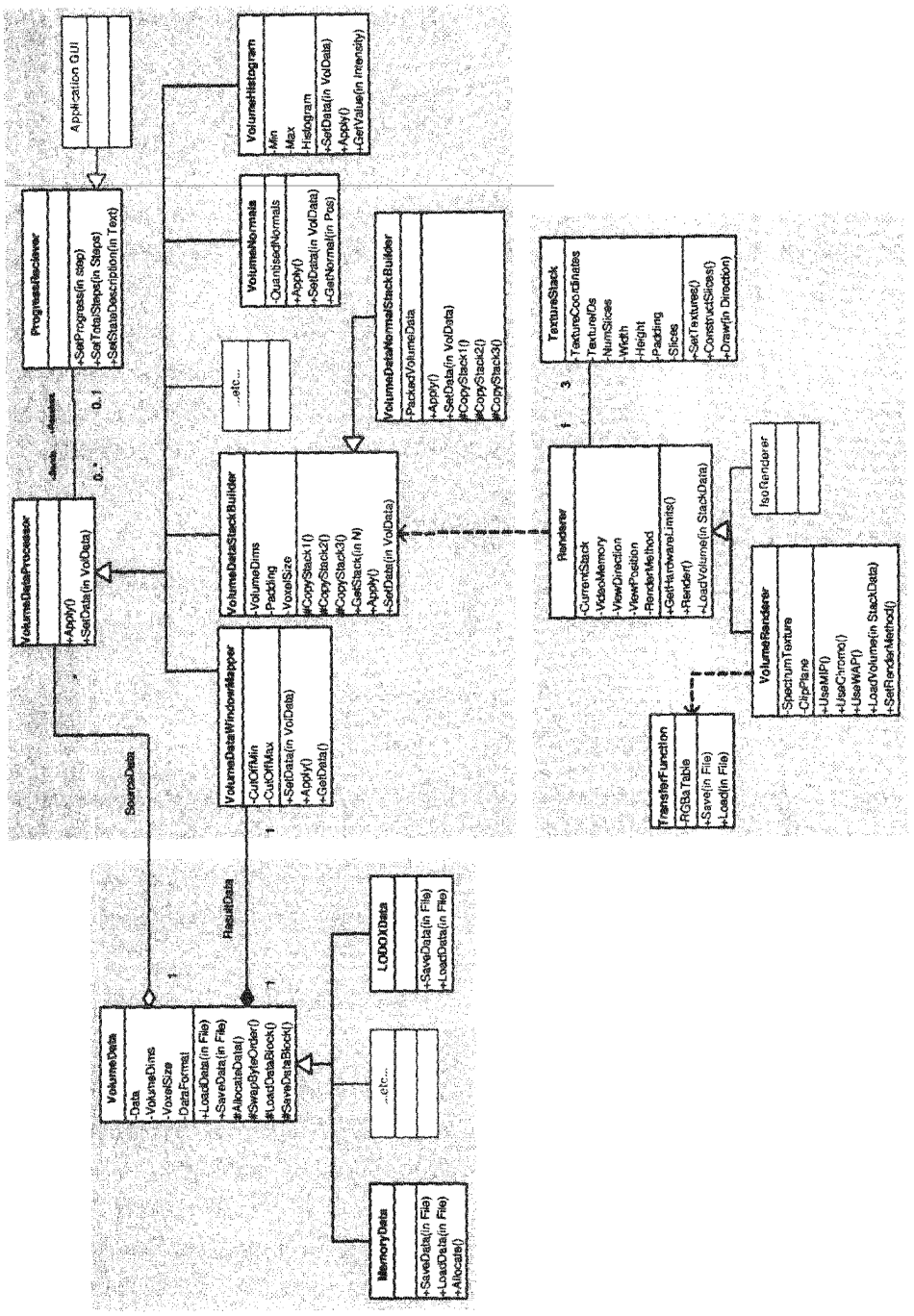


Figure 39: The volume rendering library structure illustrated abstractly using a UML class diagram.

problem by dividing the volume into smaller volumes (*bricks*). The dimensions of the bricks may be selected as a power of two. The "remainder" regions may be dealt with by further processing. Once the volume is divided, the bricks are rendered separately and the results composited in the final stage. See Figure 40.

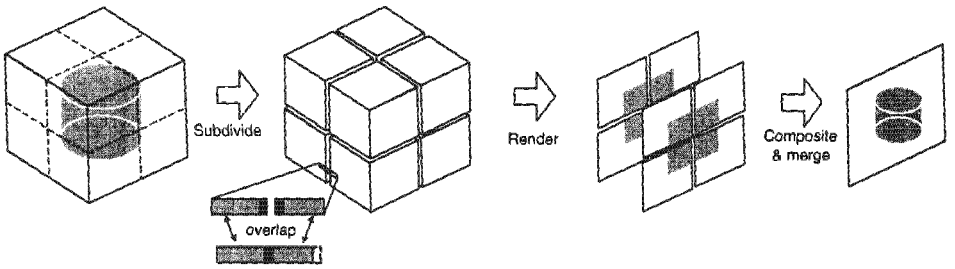
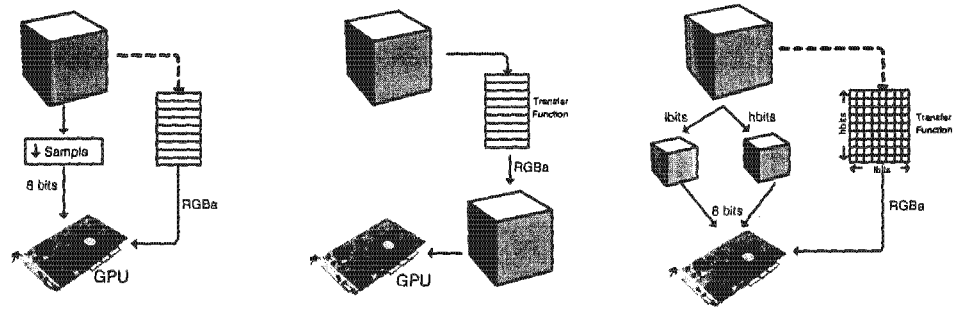


Figure 40: Bricking volumetric data. The volumetric data is divided into sub regions. The subregions are selected so that the bricks have power-of-two dimensions and are small enough to fit into the available texture memory. The bricks are overlapped to avoid artifacts appearing at the seams in the final image.

The bricks are overlapped to prevent seams appearing in the final image. The seams are the result of sampling artifacts occurring at the edges of the bricks during texture mapping. When the intermediate images are blended and joined, their opacities need to be correctly weighted and scaled according to the blending equations and the number of slices in the brick. The bricking approach may be used to render volumes larger than the available texture memory and offers an avenue for distributed rendering [70].

The cost of reading the intermediate images back from video memory, stitching and compositing them together is tangible. In distributed rendering the brick rendering is done remotely. Data transfer becomes the bottleneck in such systems. Fortunately, new technologies, such as AGP 3.0, which increase bus transfer rates and 1 Gbit network devices, are reducing the cost.

LODOX generated data is usually a power of two in the xy plane, while the z dimension is arbitrary. Padding in the z dimension is sufficient in most situations. The machine is capable of producing volumes with dimensions exceeding 512^3, which requires far more texture memory than is available on current video cards. Bricking resolves this issue. The LODOX reconstruction algorithms generate volumes of floating point values, which may be quantised to 16 bits without significant information loss. The 16-bit voxel representation has obvious memory saving results (two bytes per voxel as apposed to four). Neither of these two formats can be used directly to produce a classified renderings on current consumer graphics card's Open-GL implementations.



(a) (b) (c)

Figure 41: Classified volume rendering options. Open-GL only supports a limited set of texture formats. If the data is to be classified and rendered the data may be a) Quantised, b) pre-classified, or c) split into multiple textures and recombined on the video card.

There are three possibilities to work around these limitations and produce coloured (classified) renderings — see Figure 41. They are:

1. Quantise, or down sample, the representation to 8 bits per voxel and use the colour look up tables supported by the graphics hardware to perform real time classification.

2. Classify the data prior to loading into texture memory (*pre-classification*) and load the texture with the colour information (RGBα) directly.

3. Split the volume into two 8-bit texture volumes and use the features available on the latest hardware, which allows the two eight bit textures to be combined. The two values are then used to index a texture representing the classification table. This method is called *multi-dependent texturing*.

Down sampling to 8 bits results in a large loss of information and detail. It is difficult to select a down sampling scheme which still produces a reasonable visualisation. The pre-classification method uses 32 bits per texture element (8 bits for each channel - RGBa). The increase in flexibility comes at a heavy memory penalty (four versus one or two byte per voxel). This amount of data cannot be transferred between the CPU, where classification is done, and the video memory rapidly enough to support real time classification. The multi-dependent texturing method has the advantages of both techniques — the least possible memory use for a full representation with real time classification. However, it requires Open-GL extensions and may not be directly portable (competing vendors do offer similar extensions). Table 2 provides a brief comparison.

Method	Quantised	Pre-classified	Multi-dependent
Memory requirements	Least	Most	Exact
Classification	Post	Pre	Post
Data precision	8 bit	N bit	N bit
Open-GL extensions	No	No	Yes

Table 2: A comparison of three work arounds for Open-GL texture format limitations

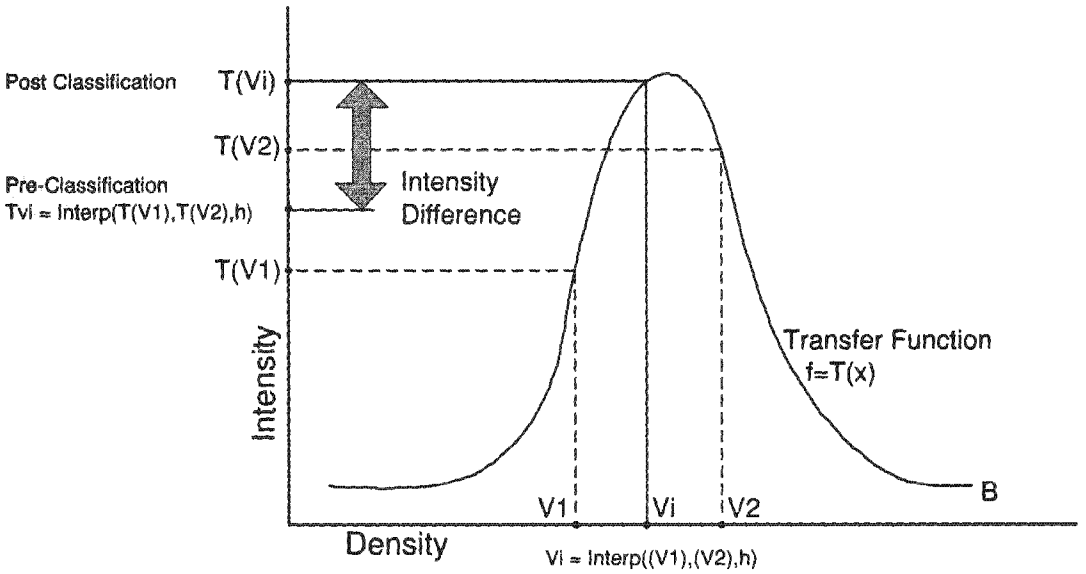


Figure 42: The differences between pre- and post- classification. During texture mapping the textures are interpolated. In pre-classification the texture contains the voxels' colour and opacity values, while in post-classification it contains the voxels' densities. The graph shows the blue channel of a transfer function ($f = T(x)$). Two voxels (v_1 and v_2) with different densities are shown along with their corresponding classified blue intensity values ($T(v_1)$ and $T(v_2)$). Post-classification uses the interpolated density value (v_i) to determine the final intensity ($T(v_i)$), where as pre-classification interpolates the classified intensities to get the intensity (T_{vi}).

The real-time classification techniques support what is known as *post-classification.* The difference is subtle, but can be seen when comparing the rendered images. When pre-classification is used, the classified voxel data (colour and opacity data) are interpolated during texture mapping. In post-classification, the voxel values (densities) are interpolated and the resulting value is classified. The difference is represented in Figure 42 and by the equations:

$$\text{Post classification:} \quad I = \text{classify}(\text{interp}(v_1, v_2, h))$$
$$\text{Pre classification:} \quad I = \text{interp}(\text{classify}(v_1), \text{classify}(v_2), h)$$

I is the resulting intensity, which is determined for each colour channel (RGBα). v_1 and v_2 are two voxel values (densities). h is the interpolation factor in the range $[0, 1] \in \mathbb{R}$.

Post-classification generally produces higher quality renderings and is the "correct" interpolation scheme if Equation 1 is considered. Additionally, the separate interpolations of the opacity and colour channels, in pre-classification, result in sampling artifacts[139]. Recently Engel, et al. [29] developed a technique to produce corrected, high quality renderings using pre-classification.

Variations of multi-dependent texturing are used to implement many of the PDC volume rendering extensions discussed in Chapter 3. For example, the volume gradients used in the shading calculations may be stored as a separate texture or part of a texture. A recent Open-GL extension includes quantised normal representation into the list of available textures formats. This extension simplifies the implementation and reduces the memory requirements for shading enhancements. Many new volume rendering techniques have resulted from the increased programmability of the combination stage. Draft versions of the new Open-GL specification include this functionality via *fragment programs*. Fragment programs are small segments of code which tell the graphics card how to combine texture and other environment variables. At the time that this system was developed, these techniques were only available through Open-GL extensions. For more reading on this topic refer to [107]

The data processing classes in the library reflect the needs of the renderer. *VolumeDataProcessor* is the base class for the other processing classes. It provides a common interface and reporting mechanism through which sub-classes may report the progress of their processing. The *VolumeDataWindowMapper* class supports quantisation of the volume data down to 8 bits. The conversion interpolates the data between the specified maximum and minimum cut off values. The *VolumeDataHistogram* class calculates the histogram for a given volume. The histogram can provide some insight for the selection of the cut-offs for the *VolumeDataWindowMapper* and identification of regions of interest. The *VolumeNormals* class approximates volume normals using central difference gradient calculations and quantises the results accurately using 16 bits. The *VolumeDataStackBuilder* and *VolumeDataNormalStackBuilder* classes convert the volume and gradient data into formats suitable for use as textures, including reordering the data according to each of the axes and adding padding.

```
1       /* Select the primary axis (largest value) */
2       Vector3 v = abs(viewVec);
3       i = (v[0]>v[1] && v[0]>v[2])? 2:((v[1]>v[0] && v[1]>v[2])? 1: 0);
4       curTexStack = &textureStack[i]; // select the texture stack
5
6       // Determine the stack traversal direction (back to front rendering)
7       direction = (viewVec[curTexStack->getPrimaryAxis()] < 0)? 1: -1;
```

Listing 1: Code for choosing which stack should render

5.2.4 Volume renderer

The volume rendering classes are constructed from a base class (*Renderer*). This class establishes and maintains the Open-GL state (settings) required for rendering. It includes three texture stacks, one for each volume axis. For a given view position and direction, a single stack is used. The stack used for rendering is chosen according to which of the three contains slices that are at the greatest angle relative to view direction. Listing 1 gives the code to do this.

The *TextureStack* class contains the details necessary to build the slices, upload the textures to the graphics card and render the textured slices. Open-GL uses an integer to identify a texture which has been uploaded to the video memory. The command *glGenTextures* fills in a list of integer names which may be used as identifiers. Named textures may then be loaded and their environments set by binding their name to the current texture environment. The data transfer is done through a call to *glTexImage2D*. The *format* and *internal* parameters are used to specify whether the data is 8 bit indexed (look up table based) or 32 bit RGBa. Other than this, both the pre- and post-classified data is handled the same way — see Listing 2. If post-classification is used, the colour look up table is loaded using the *glColourTable* command.

The *TextureStack* uses an array to represent the actual textured slices (quadrilaterals). The array consist of interleaved vertex and texture coordinates, in other words:

$$[V_{0,x}\ V_{0,y}\ V_{0,z}\ T_{0,s}\ T_{0,t}\ \ldots\ V_{i,x}\ V_{i,y}\ V_{i,z}\ T_{i,s}\ T_{i,t}\ \ldots]$$

where V_i is the i^{th} vertex and T_i is the associated texture coordinate. The array is filled by constructing a template slice and copying the result after shifting its position for each slice. This is possible, as only one of the components of each vertex changes between slices. The texture coordinates and the vertex positions are specified to skip any padded data.

As a result of this construct, drawing the textured quadrilaterals is relatively simple — see Listing 3. The array traversal direction is chosen to ensure the textures are drawn back-to-front.

```
1   glGenTextures(nSlices,texNames);  // Create a list of texture names
2
3   for (i=0; i<nSlices; i++) {
4       // Upload the texture and associating it with the specified name
5       glBindTexture(GL_TEXTURE_2D,texNames[i]);
6       glTexImage2D(GL_TEXTURE_2D,0,internal,
7           width,height,                 // texture dimensions
8           0,format,                     // data description
9           GL_UNSIGNED_BYTE,             // input data type
10          (void*)volume);               // pointer to the start of the slice
11
12      // Apply to each named texture environment
13      glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
14      glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
15      glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, FILTER_TYPE);
16      glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, FILTER_TYPE);
17      volume += step;       // Next slice
18  }
```

Listing 2: Open-GL for binding, loading and setting the environment for a texture.

```
1   /* point to the array of slices */
2   glVertexPointer(3,GL_FLOAT,stepSize,sra);
3   glTexCoordPointer(2,GL_FLOAT,stepSize,sra+3);
4
5   int slice      = (direction>0)? 0: nSlices-1;
6   int inc4       = direction*4;
7   int arrayStart = slice*4;
8
9   for (i=0; i<nSlices; i++, arrayStart+=inc4, slice+=direction) {
10      /* Bind the texture environment for the slice */
11      glBindTexture(GL_TEXTURE_2D,texNames[slice]);
12      glDrawArrays(GL_QUADS,arrayStart,4);
13  }
```

Listing 3: Open-GL code to render the textured slices in a stack

During rendering, the actual composite operation depends on the GL rendering engine's state. MIP, WAP and the standard volume composit are implemented simply by changing the blending settings, as shown in Listing 4.

```
1   glEnable(GL_BLEND);
2   switch(method) {
3     cmWAP: glBlendFunc(GL_CONSTANT_ALPHA,GL_ONE);
4            glBlendColor(1,1,1,1.0f/nSlices);
5            break;
6     cmMIP: glBlendingEquation(GL_MAX);  // alternatively GL_MIN
7            break;
8    default: glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)
9   }
```

Listing 4: Open-GL code for setting the compositing method

Texture-based volume rendering is *fill limited*. This means that the speed of a rendering is limited by the dimensions of the final image. Ideally, the volume should be rendered to the display with the pixels roughly the same size as the projected voxels. If the volume is to far away (the pixels are much larger than the projected voxels), undersampling is likely to occur. If the volume is too close, the slices become apparent and aggravate the aliasing artifacts associated with the use of bilinear interpolation.

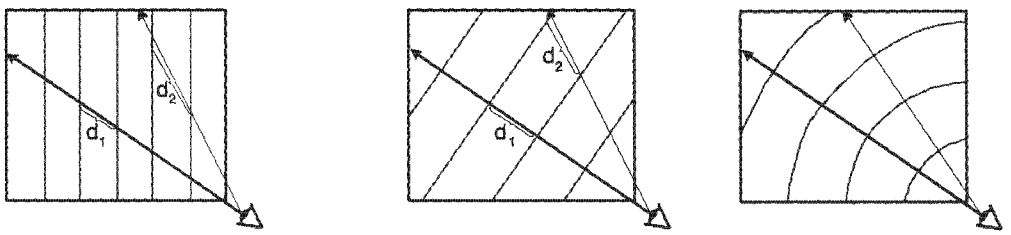
The geometry of close viewing emphasises the difference between the sampling rates along the rays — see Figure 43a. Three dimensional texture methods suffer from an equivalent problem (Figure 43b). However, an efficient solution exists which is known as spherical shells [131]. Rather than representing the data using slices, the spherical shells method uses a set of tesselated concentric spherical shells — Figure 43c.

During interactive rendering the texture stacks are switched depending on the view direction. This action is noticeable and is a result of the differences in sampling of the two texture stacks. Figure 44 illustrates the differences.

This switching artifact is considerably worsened if the number of slices (sampling rate) between the stacks differ. This is attributed to the non-linear nature of the blending operations. Either the volume should be resampled, to have the same sampling rate along each axis, or the opacity values used in the classification should be adjusted. The adjustment required for the opacity is approximated by [70]:

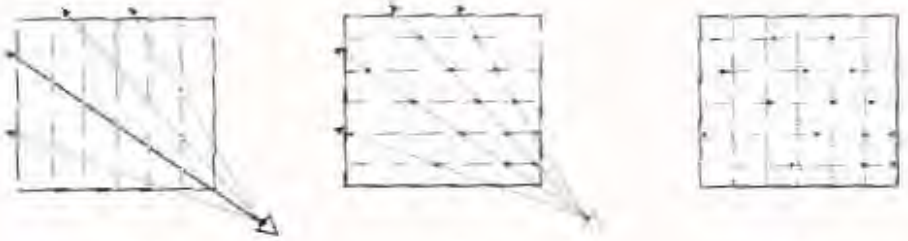
$$\alpha_n = 1 - (1 - \alpha_o)^{\frac{r_o}{r_n}}$$

where α_n and α_o are the new and old alpha value respectively. r_n and r_o are the old and new sampling rates respectively.



(a) (b) (c)

Figure 43: Sampling artifacts visible in close viewing. a) The close viewing geometry emphasises the different sampling rates along the rays, indicated by distances d_1 and d_2. b) This problem also occurs in 3D texture methods. c) A solution is to use a 3D texture method with concentric spherical shells rather than slices.



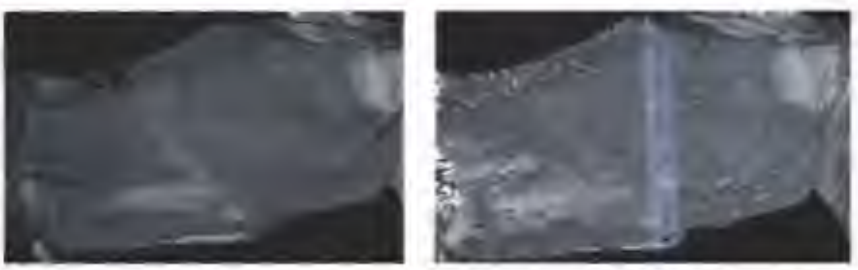
(a) (b) (c)

Figure 44: Sampling differences between texture stacks. a) and b) show sampling for two texture stacks. c) Shows the sample points overlayed. The sampling differences result in a noticeable switching artifact during interactive rendering.

Even in a perfectly sampled volume, aliasing artifacts may still occur as a result of the transfer function. All volume rendering techniques suffer from this problem. The final image is the result of both the classification and the volumetric data. Consequently, the highest spatial frequency is a function of both the transfer function and the data. Using the notation introduced in Chapter 2, we have

$$F_{max} = \max(\text{freq}(s(\rho)\ \ c(\rho))), \quad \forall \rho \text{ in the volume}$$

where $\text{freq}(x)$ is a function returning the frequency of signal at x. $\text{freq}(s(\rho) \cdot c(\rho))$ is returning the frequency of a classified volume for the given point ρ. c and s are the colour and extinction coefficients. Figure 45 illustrates this problem.



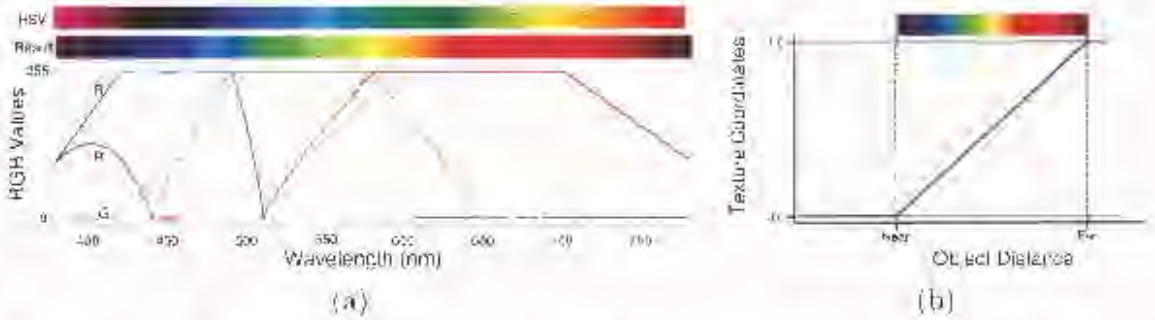
(a) (b)

Figure 45: Sampling artifacts in a well sampled volume. The sampling rate depends on both the data and transfer function. If the transfer function contains high frequencies then blocky sampling artifacts will appear. The foot data set is shown with a) low frequency transfer function and b) the same transfer function containing a localised sharp change, which results in high frequencies.

The 2D texture-based, trilinear interpolation and acceleration techniques based on the work reported in [28] were implemented. However, it was not necessary to use these extensions in the experimental application. Thus, they are not covered here. The interested reader should refer to the original work [28, 29].

5.2.5 Chromostereoscopic rendering

Here a volume rendering extension is presented, which implements chromostereopsis in volume rendering. This technique is novel in that chromostereopsis has never been applied to volume rendering. Chromostereopsis relies on the pseudo colouring of an image to generate the stereoscopic effect. The perceived depth is dependant on the wavelength of the colour. The image must be coloured according to the visible spectrum, ranging from "red", the closest, to "violet", the furthest. A one-dimensional texture map representing the spectrum is generated using the wavelength to RGB model presented in [37, 111]. The model approximates the spectrum more effectively than using the HSV colour model, as was previously used in polygonal mesh chromostereoscopic systems[5]. There is no unique mapping between wavelength and RGB, as it varies according to viewer perception and display characteristics. Figure 46a illustrates how the RGB values vary with respect to the wavelength for the chosen colour model.



(a) (b)

Figure 46: a) The spectrum texture needed for chromostereopsis is generated using a spectral model. The model maps wavelength (nm) of visible light to RGB values. The resulting colour spectrum is shown above. Compare this to HSV based spectrum shown. b) The texture is mapped into the scene according to the viewing distance. The parameters, near and far, are used to calculate the texture coordinates.

Once the spectrum texture is generated, multi-texturing is used to combine the volume textures with the spectrum texture. The 2D texture volume rendering algorithm remains unchanged. The volume must, of course, be classified using a gray-scale transfer function. Using colour destroys the depth effect completely. The texture coordinates for the spectrum

```
1   static float white[4] = {1.0f,1.0f,1.0f,1.0f};
2   float texGenParams[3];
3
4   // Setup the second texture environment
5   glActiveTextureARB(GL_TEXTURE1_ARB);
6   glDisable(GL_TEXTURE_2D);
7   glTexParameteri(GL_TEXTURE_1D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
8   glTexParameteri(GL_TEXTURE_1D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
9   glTexParameteri(GL_TEXTURE_1D, GL_TEXTURE_WRAP_S, GL_CLAMP);
10  glTexEnvfv(GL_TEXTURE_ENV, GL_TEXTURE_ENV_COLOR, white);
11  glTexEnvi(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
12
13  // load the texture
14  glTexImage1D(GL_TEXTURE_1D,0,4,spectTexWidth,0,
15          GL_RGBA,GL_UNSIGNED_BYTE,spectTex);
16
17  glEnable(GL_TEXTURE_1D);
18
19  // setup texgen params
20  texGenParams[0] = texGenParams[1] = 0.0f;
21  texGenParams[2] = -1./(far-near);
22  texGenParams[3] = -near/(far-near);
23  glPushMatrix();
24  glLoadIdentity();
25  glTexGeni(GL_S,GL_TEXTURE_GEN_MODE, GL_EYE_LINEAR);
26  glTexGenfv(GL_S, GL_EYE_PLANE, texGenParams);
27  glEnable(GL_TEXTURE_GEN_S);
28  glPopMatrix();
29
30  // switch to first texture environment (volume rendering as normal)
31  glActiveTextureARB(GL_TEXTURE0_ARB);
32  glTexEnvfv(GL_TEXTURE_ENV, GL_TEXTURE_ENV_COLOR, white);
33  glTexEnvi(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);
```

Listing 5: Open-GL setup for chromostereoscopic volume rendering

texture are generated automatically using the *glTexGen* command.

glTexGen is setup so that objects between the *near* and *far* parameters are textured with the appropriate colour, while outside of this range the colours are clamped to "red" and "violet" respectively — refer to Figure 46b. In an ideal situation, *near* and *far* bound the visible objects tightly to achieve the best depth effect. The bounding box is used as an indication of the range. Depending on the transfer function, this may not be very effective, in which case the user must adjust the settings.

Listing 5 gives the Open-GL commands needed to setup the volume renderer for chromostereopsis. The results are shown in Figure 47. It is interesting to note that this technique can provide additional depth information for MIP and WAP composites. These composites are usually considered "flat"[44]. Additionally, this technique can be used to create the atmospheric depth effect by using a texture which darkens (lowers the attenuation) regions of the volume further from the viewer. This cue has not been applied using this approach

in prior texture-based volume rendering systems.

(a) (b) (c)

Figure 47: Chromostereopsis results shown for the foot data set using different volume rendering compositing strategies. a) Volume composite, b) MIP, c) WAP.

5.2.6 Volume rendering application

The experiment application provides a user interface to the volume rendering library, along with some additional functionality, for example a 3D cursor.

Stereo support

The application supports stereoscopic volume rendering. Stereoscopic rendering is achieved via the Nvidia stereo driver, using "OverDrive". The user controlled stereo parameters (stereo enable, eye separation, depth extent, et cetera) are specified via the display properties and are stored in the Windows' registry. Although Nvidia claim to have an API available to manipulate these parameters, it has not been released. The currently available driver[1], which provides stereoscopic support, is somewhat dated and is only available for Windows. The drivers are not stable, occasionally failing completely or producing incorrect stereo images.

The application enables and disables stereo viewing by switching between full screen and windowed modes. In the windowed mode, the display manager decorations (title bar, task bar, etc) are removed or minimised to reduce the difference in appearance between the two states.

User interface

The user interface is as simple as possible. There is no facility to modify the transfer function as this process is too time consuming and not relevant to the experiments. Another tool was used to prepare and classify the data for the experiments. The interface is modal. The

[1]Stereo driver version 30.87

mouse is used to change states and interact with the application. The state is indicated by icons and the cursor shape. The modes are "view point selection" and "3D cursor" – see Figure 48.

In the "viewpoint point selection" mode, a trackball manipulation widget[33] is used to change the view position and direction. In "3D cursor" mode a 3D cursor appears. Its position on the screen is specified by the mouse position. The depth in the 3D scene is specified using the mouse scroll wheel. The 3D cursor is used to measure and mark the position of features. When the trackball is combined with the 3D cursor it is easy to loose track of the cursor position. A function for resetting the cursor position is provided. Refer to Appendix D for a detailed description of the interface.

3D-cursor and point markers

The flat (2D) 3D-cursor and markers are shown in Figure 49. They are always oriented towards the viewer. The flat shapes were selected to ensure that the volume and X-ray feature selection mechanism would appear similar. Mouse movements change the cursor position with the motion relative to the view direction, providing natural direct manipulation.

The cursor and markers are opaque. They are incorporated into the volume rendering by enabling the depth buffer and depth testing in Open-GL. The objects are rendered before the volume. When the volume is rendered, the parts of the slice falling behind the object are not blended, while those that are in front are blended. This effectively embeds the objects

Figure 48: A screen shot of the volume rendering application. Several feature points have been marked

Figure 49: The 3D cursor and point markers used in the test applications. The left image illustrates how these objects look when they are embedded in the volume.

in the volume. Partially transparent objects require special attention. These objects have to be split between the slices and then rendered interleaved with the volume — see [73] for more details.

5.3 X-ray visualisation

The stereo enabled digital X-ray viewer software relies on the capture of valid stereo pairs for its effectiveness. The visualisation algorithms are simpler than those used in volume rendering. However, the presentation system is complicated by several issues. For example, X-rays generally have 2 bytes per pixel (12 – 14 bits) and have high resolutions.

Several CT/X-ray scanner geometries exist, for example spiral, fan beam and parallel beam. The LODOX machine has a *fan beam* geometry, and the discussion is limited to this form.

5.3.1 Fan beam stereo pairs

Fan beam scanners have a triangular cross-section and sweep out a prism volume as illustrated in Figure 50. The scanning volume effects the resulting images such that they represent an orthogonal projection in one dimension, while perspective in the other.

The perspective projection produces images which are consistent with natural viewing, containing size and linear perspective PDCs, which are reliable sources of depth information. The perspective depth cue and stereopsis have a strong relationship, in that the stereoscopic effect is scaled by the perspective PDC (refer to Sections 3.2 and 4.2.3). The same stereoscopic effect can be achieved with a lower parallax if a wider field of view/fan beam is used. For the same detector width, a wider field of view can be achieved by moving the source and detector closer together.

The orthogonal projection is unnatural to the HVS. Fortunately, X-rays are not "familiar scenes", so a fair amount of leniency exists. Mixing perspective and orthogonal projections

results in an image with mixed properties. The effect of the perspective FOC is considerably reduced. In addition, this atypical projection distorts the image of scanned objects. For example a sphere is projected as an elongated ellipse.

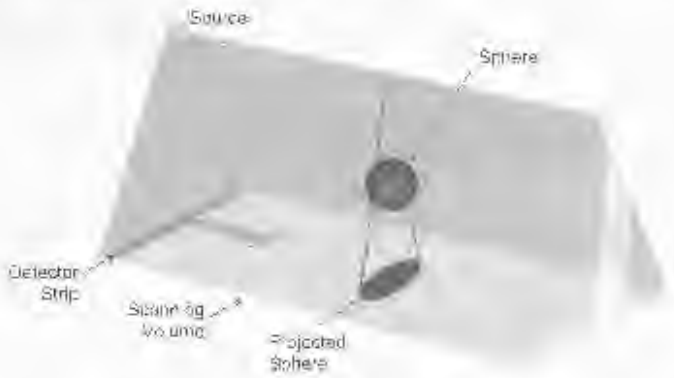


Figure 50: The fan beam X ray scanning volume. Fan beam scanners have a triangular cross section which sweeps out a prism. This geometry produces a projection operation which distorts the shape of imaged objects.

Capturing a stereo pair involves scanning the subjects from two different viewpoints. Several options exist for capturing stereo pairs, depending on how the detector-source gantry is able to move. Fan beam scanners have either a moving gantry or moving table.

If a small horizontal motion is built into either the gantry or table's movement mechanism, an off-axis like geometry can be used to capture the stereo pair. A similar motion can be achieved by shifting the subject. It is difficult to achieve a uniform movement along the length of the subject. Beyond this, in a medical setting, patient movement is frequently undesirable. Another capture approach involves rotating the gantry around the subject. Such an arrangement produces a toe-in like stereo geometry with one important difference. There is *no* vertical disparity, due to the semi-orthogonal projection. For the same reason, the distortion introduced by the rotation is less noticeable than that found in standard toe-in geometry. The distortion remains in the horizontal parallax and the shape of planes of equal parallax. In other words, Equation 11 still holds true.

The separation between the viewpoints should be chosen according to the recommendations discussed in Section 4.2.3. The largest possible separation constrained by the maximum acceptable parallax should be selected to maximise the depth effect and minimise the error in depth discrimination [16]. The maximum acceptable parallax between two images is a function of the size of the scanned object and the physical geometry of the machine. For off axis like configurations, the desirable translation distances can be determined

from Equation 21 in Section 4.2.3 with relevant variable substitution. For the toe-in like configuration, Equation 11 can be used to determine the separation angle, ϕ, numerically.

Construction of the stereogram from the captured stereo pair includes similar calculations to ensure optimal viewing for the display properties and for the viewer to display distance. Allowing the viewer to control the horizontal separation between the images in the stereo pair caters for the individuals preference and has the added advantage of allowing horopter movement. As an object moves further away from the horopter the accuracy of the depth perception decreases (see Section 4.4.2). Allowing the viewer to adjust the horopter position localises the depth perception to the region of interest. The horizontal separation must not be excessive, otherwise diplopia occurs.

For the prototype system only phantom subjects where used. Live subjects will introduce additional complexities, for example, subject movement. Subject movement is undesirable as it introduces false disparities which may destroy or warp the depth information. Faster time multiplex scanning (left and right scan lines alternately) using a stereo gantry would be a possible solution. Design techniques and technologies suitable for building low cost stereo gantries are covered in [126, 131, 143].

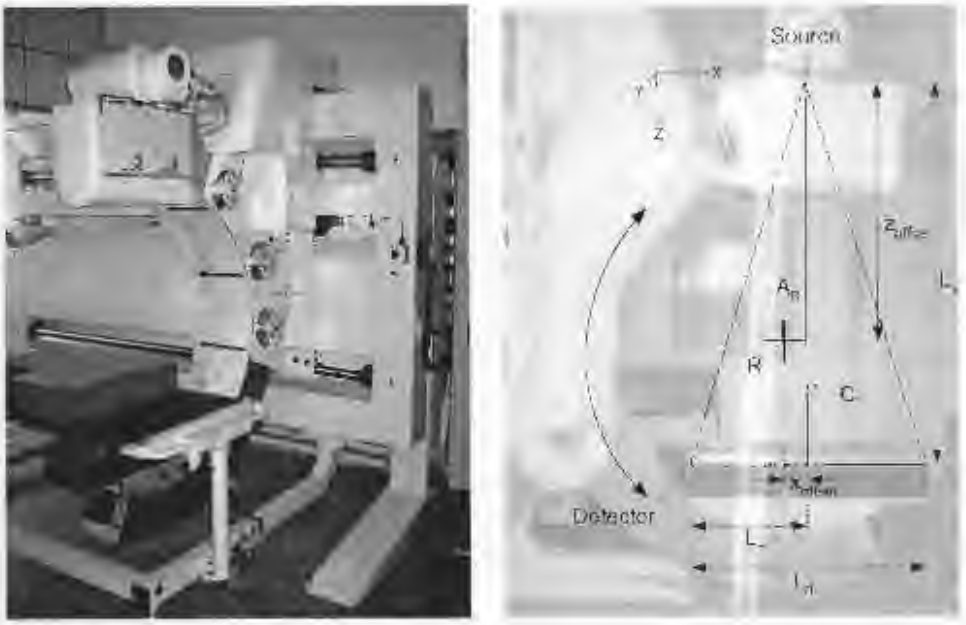
5.3.2 LODOX machine stereo pairs

In order to construct optimal stereograms and to allow the X-rays to be augmented with computer graphics, the visualisation system must be calibrated to the X-ray machine's geometry. Computer graphics is used for the 3D-cursor and markers. It is relevant to examine the specifics of the LODOX machine for this purpose. Fortunately, the measurements required for the viewing system are also required for CT reconstruction. The measurements shown here were taken from earlier LODOX-CT work [23].

LODOX geometry

The LODOX machine's overall configuration is illustrated in Figure 51a. The machine consists of a C-arm and wall mounted support structure. The X-ray source and detector are attached to the ends of the C-arm. The C-arm moves laterally along the scanning bed, via stepper motors. It may be rotated through an angular range of 90°. The centre of rotation is off centred from the central X-ray beam. A close up of the C-arm configuration is illustrated in Figure 51b with an overlayed schematic representation.

The schematic diagram illustrates some of the basic measurements required to model the scanners projection. The axes were chosen to ensure that x and y correspond to the horizontal and vertical components of the generated scan image. The y axis protrudes out of



(a) (b)

Figure 51: The LODOX geometry a) The LODOX machine consist of a C-arm gantry and wall mounted support structure. b) Schematic representation overlaying an image of the C-arm

the page. C represents the central ray, which is perpendicular to the detector. R represents the centre of rotation of the gantry. The parameters are described in Table 3.

Parameter	Description	Length
L_s	The distance between the X-ray source and detector.	$1304.5mm$
L_{st}	The width of the active detector strip.	$665.3mm$
L_c	The distance along the detector from the first pixel to the central ray, C.	$333.1mm$
x_{offset}	The perpendicular distance between central ray, C, and the centre of rotation, R.	$60.0mm$
z_{offset}	The distance, along the central ray, C, between the source and the centre of rotation	$956.5mm$

Table 3: LODOX parameters. A description of the parameters and their measured values used in the schematic diagrams of the LODOX machine.

In addition to the machine parameters, parameters describing how the X rays images were captured are needed. The image specific parameters are the angle at which the image

was captured, θ, and the image resolution. θ is measured around the centre of rotation, R, relative to the axis A_R. A_R is the vector representing the zero angle — in Figure 51b $\theta = 0$. The resolution is stored along with the image data. These parameters may vary for each scan. The detector width, L_d, and scan length, in combination with the image resolution, can be used to determine the pixel size. The X-ray image's pixels are generally square.

The projection model

A projection model is required to support the use of $3D$ computer graphics with the X-ray images. The adopted model assumes the projection to be ideal, ignoring errors which result from imperfections in the detector and source. The model assumes that the source can be represented as point source and the detector is perfectly linear. For a detailed discussion and modeling of detector and source errors see [24]. The projection is modeled using a combination of the traditional perspective pin hole and parallel camera models [33]. The model places the origin at the point representing the X-ray source, positioned at the centre of the scan. If the gantry is rotated by the angle θ, the 3D coordinates are rotated by $-\theta$ with respect to the centre of rotation, before projecting the points. The relative motion is the same and keeps the projection model simple. Thus, for a $3D$ point, $P = (x, y, z)$, the projected point, $r = (x, y)$, is given by:

$$
\begin{aligned}
P' &= T(x_{\text{offset}}, 0, z_{\text{offset}}) R_y(-\theta) T(-x_{\text{offset}}, 0, -z_{\text{offset}}) P, \\
r_x &= \frac{L_s P'_x}{P'_z}, && (24) \\
r_y &= P'_y && (25)
\end{aligned}
$$

$T(x, y, z)$ represent the standard translation matrix and $R_y(\theta)$ represents the rotation matrix, rotating by θ around the y axis. Unfortunately, the projection equations cannot be combined into a single 4×4 matrix operation due to the perspective division occurring in only one component. Consequently, standard $3D$ graphics libraries cannot be used for rendering the graphics. It is convenient to represent the measurements of the LODOX parameters in terms of the image pixels. Doing so implies that Equations 24 and 25 produce the projected points in the image coordinate space, requiring no additional transformations.

Capturing stereo pairs

Capturing a stereo pair is a fairly straight forward procedure. The subject is place on the scanning table. The table is lifted to ensure the C-arm will not collide with the table when it

is rotated. The first scan is made. The gantry is rotated to the second position and another scan is made. The resulting images must be vertically aligned and cropped. The vertical alignment is necessary as the scanner is not guaranteed to return to the original position. (Reliable gantry positioning is required for LODOX-CT and will be incorporated in future hardware revisions.) Cropping removes noise, which occurs along the edges of the scan. The edge noise can vary significantly and may cause problems with depth perception. The noise contains high intensity values which invariably over-saturate the image, appearing as bright white regions. This reduces the viewers visual acuity when viewing the surrounding regions (refer to Webber's Law, Section 4.3.1).

The most difficult step in the procedure is deciding what angular separation to use between the scans. As stated earlier, the equation for horizontal parallax (Equation 10) in combination with the maximum acceptable parallax ($|\beta| = 1.5°$) should be used as a guideline. Equation 8, and consequently Equation 11, were derived for a geometry with the centre of rotation lying on the z axis (equivalent to the central ray of the fan beam). However, the LODOX geometry has an off-centred rotation. Consequently, the equation for horizontal parallax needs to be recalculated using the LODOX geometry and projection model.

Given a $3D$ point, P, the two projected points r_L and r_R corresponding to the left and right images are determined using Equation 24. r_L and r_R can then be used to determine the horizontal parallax, as given by Equation 10. Notice that each projection calculation requires an angle, say θ_1 and θ_2. Due to the off-centred rotation, the angular separation cannot be used to represent both angles from some arbitrary reference position.

Remember that the results of the equations are in terms of the *detector's pixels*. If the X-rays are displayed without any scaling in the horizontal dimension, then the visual angle can be determined using Equation 5 with d equal to the approximate viewer to screen distance. d is measured in terms of the *display's* pixels. If the image is scaled, then the horizontal parallax must be scaled equally. The differences in the pixel aspect ratios between the X-ray and the display may require vertical scaling to prevent image distortion, but this does not affect the horizontal parallax. The resulting visual angle is a function of several variables. Many of the variables are constrained by the system setup and the viewer.

For example, the display resolution and monitor size are generally fixed, fixing the display's pixel size. The machine parameters are fixed and the initial scanning angle would be selected by the machine operator to provide the best view of the region of interest. The viewers distance from the screen may be approximated using ergonomically recommended viewing distances, or it may be measured. Several points should be considered when determining the maximum parallax for the subject. Good candidate points are the vertices of

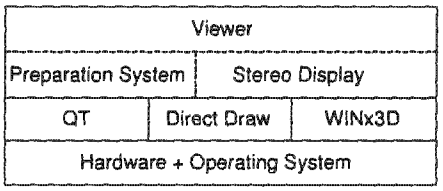


Figure 52: A structural overview of the X-ray viewer.

the bounding region. The only remaining parameter is the second scanning angle, which can be calculated numerically using the information which has been gathered.

5.3.3 The X-ray viewer

The X-ray viewer software is divided into two sub-systems. The first part deals with the data preparation and storage. The other is the actual stereoscopic viewing subsystem. Figure 52 illustrates a structural overview of the system.

The user interface was written in QT [125], a platform independent widget library, while the stereo viewer uses WINx3D, a second generation stereo system library.

Data preparation

The preparation sub-system deals with the parameter specification, file management (loading and storing) and data preparation. In terms of data preparation, the X-ray images may be loaded as a sequence to build an animation or as a pair to build a stereo pair. A series of stereopairs may also be built from animation sequences. The number of frames between each of the images in the stereopair may be specified and in this way the angle between the images controlled.

Cropping, alignment and a simple noise reduction operation have been implemented. The cropping and alignment markers are specified by the user interactively. In an effort to reduce the work required to process long image sequences, the marker position updates are synchronised across all the images.

Stereoscopic display

The stereoscopic display is based on WINx3D, which provides a uniform hardware access layer to the underlying stereo device. WINx3D interacts with DirectDraw. DirectDraw is the 2D graphics subset of Microsoft's DirectX API. The API exposes the graphics cards display buffers to the application. The application can then transfer image data directly to

the video memory through a *blitting* operation.

Depending on the underlying stereo hardware, the stereo driver acts differently. Nvidia graphics cards support page flipped stereo. In this method the stereo pair is loaded into the video memory and the hardware alternates between the left and right images (by changing the starting memory address of the displayed buffer). The stereo pair is built by placing the images side by side and overwriting the first few pixels of the resulting image with a signature byte sequence. When the image is blitted to the video memory, the driver detects the signature and takes the appropriate action. WINx3D combines the individual images, writes the signature and times the update of the stereo image (to avoid any noticeable flicker).

This approach has implications in terms of the maximum rendering speed attainable. It requires at least three screen resolution size blit operations per image update. The first two blits are required to transfer the image data to the video memory. The images are then augmented with additional graphics (for example, the 3D cursor and icons). The final blit operation combines the two intermediate images into a stereopair and prepares it for final display. Despite the additional blitting requirements, it was found that the images could be updated smoothly in real time. The size of the X-rays do not pose much of a problem provided there is sufficient main memory. The screen resolution determines the amount of data manipulated by the system and not the actual X-ray image sizes.

User interface

The stereoscopic display subsystem has a user interface similar to that of the volume rendering application — see Figure 53. The interface is also modal in nature with the mouse used to control most of the features. The modes are "zoom", "contrast/brightness", "3D cursor" and "Image separation". For a detailed description of the interface refer to Appendix D.

There is no viewpoint selection mode. The user may toggle the X-ray animation on or off. The animation effectively changes the viewpoint and presents the X-ray images as an oscillating sequence.

Mouse movement pans the image across the display, except when the 3D cursor is being used. This is done so that the viewer can place the region of interest at the centre of the screen easily. This strategy was encouraged as an attempt to draw the viewer's focus away from the edges of the screen where frame cancellation may occur.

Zoom

The zoom function allows the image to be zoomed in or out. The calculations used to determine the angular separation during scanning assume that no scaling has occurred. Provided the viewer only zooms out, the maximum parallax in the image will not be exceeded. When zooming out, the parallax in the image will decrease, as will the total depth effect. If the viewer zooms in, diplopia is likely to occur. From a practical perspective, the zoom is useful to adjust the maximum parallax to the individual viewers preference. Additionally, it allows the images to be optimally viewed on a configuration different than was originally planned. For example, using a 15″ monitor at a resolution of 800×600 instead of a 17″ monitor at 1024×768.

The zooming can either be implemented by modifying the blitting parameters (size of the source and target regions) or by preprocessing the images. Preprocessing imposes a heavy delay and the zoomed images require more memory. The blitting approach relies on hardware support for linear or higher order interpolation. If the hardware does not support this functionality the image quality is degraded when zooming. The zoom preprocess method was used in the prototype.

Figure 53: A screen shot of the X-ray viewer application. The viewer is running in column interlaced stereo mode. This is an alternative to the page flipped stereo format supported by WINx3D. This stereo format is not supported by the Nvidia based graphics cards.

Contrast and Brightness

The X-ray data is 14 bits per pixel, but only 8 bit grayscale images can be displayed. The 14 bit data is mapped to 8 bits by grouping the values using a linear function. This is similar to the technique used by the current LODOX X-ray viewer system. The "contrast/brightness" mode allows the viewer to adjust the mapping function by moving the end points of the linear function. Moving the end points up and down changes the brightness of the image. While changing the distance between the end points results in contrast adjustments.

Contrast and brightness were made user adjustable for two reasons. Firstly, the X-ray data uses the full 14 bits. Creating a static map results in large detail loss and reduces the digital X-rays to the equivalent of a standard film X-ray. Secondly, it is a relatively easy task that can be performed quickly by the user. The contrast and brightness settings do affect the stereoscopic display in terms of ghosting and visual acuity (refer to Section 4.3.1). The user should be made aware of these facts, so that the image can be adjusted for the best results.

The X-ray images need reprocessing whenever the mapping function changes. This process does take a fair amount time. The exact time depends on the amount of data (which is related to the size of the X-ray images and the number of images). Ideally, this process should be threaded, allowing the viewer to continue examining the current image with as little delay as possible. The remaining image would then be processed in the background. If further adjustments are made the background processing can be restarted. The animation must be stopped for the duration of the updates. In the current implementation, the viewer must wait for the processing to complete.

3D-cursor and markers

The same cursor and markers are used as in the volume rendering application (refer to Figure 49). The partial parallel projection limits the usefulness of 3D-cursors and markers. For example, with a 3D cross hair, only the x components of the lines extending along the z dimension would change. Lines with depth appear as horizontal lines. It was found that such lines have a larger negative impact than the effects of removing them.

An attempt was made to create the impression that the cursor and markers were embedded in the x-rayed subject by using semi-transparent lines. However, this approach failed to create the embedded look and made the 3D graphics difficult to see against the X-ray images. As a result, opaque graphics elements were used.

Image separation

As suggested earlier offsetting the left and right images horizontally from each other, allows minor adjustments to be made to the image parallax. This caters for viewer preference and allow the horopter position to be changed according to region being examined. The viewer must be warned about the occurrence of diplopia which can result from excessive adjustment.

5.4 Summary

In this chapter, the prototypes and techniques developed to build the visualisation systems were examined. In the preliminary work, surface-based volume visualisation was tested and found less suitable for LACT and SACT data than voxel-based rendering. The failure lay in the representation of "uncertain" regions as crisp surfaces and the failure to handle "noise" in the data. From the available voxel-based rendering techniques it was found that a 2D texture mapping approach was the most suitable. The choice was made based on interactivity requirements and the cost and availability of the hardware acceleration solutions.

During the stereoscopic display development phase, the use of chromostereopsis, anaglyphs, shutter glasses (both the crystal eyes and clones), and a HMD were explored. The final systems used shutter glasses. A 3^{rd} generation stereoscopic application was developed to explore and test the stereoscopic system parameters and configuration. The final prototype systems used, a 4^{th} generation framework for the volume visualisation system and 2^{nd} generation framework for the X-ray system.

The final volume visualisation system was then described. The volume rendering application was built using the presented volume rendering library. The library was discussed, covering the implementation details, issues which were faced and the volume rendering implementation limitations. As part of the discussion, a novel volume rendering extension which produces chromostereoscopic images was presented. The final section on volume rendering covered the application developed specifically for the experiments described in the next chapter. The functionality of this application includes stereopsis, 3D cursor positioning and point marking for feature point selection and view point selection.

In the next section, the final X-ray visualisation system was discussed. It begun by discussing how a stereo pair can be captured using a fan-beam X-ray scanner. The LODOX machine was then used as a specific example, describing a geometric model for the hardware and a projection model for the imaging process. The described projection model was used to

augment the X-ray images with 3D graphics. How the projection model was combined with theory presented in Chapter 4 to determine the optimal capture parameters for a known workstation configuration was discussed. Finally, the X-ray viewing software which is used to prepare the X-rays and display them was presented. The stereoscopic display subsystem, including a discussion on the user interface and functionality made available through this interface was described. The rationale behind the decision made for each feature was given. The display subsystem was used in the experiments described in the following chapter.

Chapter 6

Evaluation

In the previous chapters the two PDC enhanced prototype systems and the theory which lead to their development were described. In this chapter, the effectiveness of the PDC enhancements for a given task is investigated. Experiments are conducted using both systems.

This chapter is begun with the aims of the experiments in Section 6.1. The evaluation method and the reason for its selection is discussed in Section 6.2. The participants' information is described in Section 6.3. This is followed by a discussion of the experimental task (Section 6.4), the experimental procedure (Section 6.5), and the equipment used in the experiments and its configuration (Section 6.6). The results are presented in Section 6.7 and Section 6.8 and discussed in Sections 6.9.

6.1 Aims

Two visualisation prototypes have been implemented. Both systems have been enhanced using PDCs, specifically stereopsis and KDE. Reproducing the PDCs incurs additional costs in terms of computational requirements and increases patient radiation in the case of the X-ray system. Therefore, it was decided that the PDCs, stereopsis, KDE and their combination should be investigated, to determine which is the most effective in each visualisation for a given task. The task selected was that of locating and marking feature points. The rationale behind this selection is that this task is frequently performed in a medical environment, especially in a trauma setting where it is a common task to locate and estimate the position of objects [35, 89, 132].

The experimental data will be used to rank the relative effectiveness of the PDCs. In addition, by completing the experiments it will be shown that the X-ray visualisation can

be used to perform a diagnostic task which is three dimensional in nature. Further, it will be proven that the X-ray system can be used to perform tasks as quickly and as accurately as performed in the volume rendering system.

6.2 Evaluation method

There is no universally accepted approach to quantitatively evaluate the effectiveness of a visualisation. Two common approaches are the use of questionnaires and task performance evaluations [60, 99].

Questionnaires, by their very nature, are subjective requiring that the subject give their opinions and judgement. If not designed correctly, a questionnaire may lead a subject towards an answer or alternatively produce random, unrepeatable results. Several guidelines and statistical tests exist and can be applied to ensure correctness. The data obtained from questionnaires can easily be skewed by the social and educational status of the subjects. A wide sample of subjects is required to avoid such problems. Fortunately, questionnaires are administered relatively easily.

Task-based evaluation provides a group of subjects with a set of tasks to be performed using the system [60]. The performance of the subjects during the task is used as an indication of the effectiveness of the system. Task-based evaluation is not always suitable as questions related to the effectiveness of a visualisation may not be easily represented by a set of tasks. In addition, the results may be skewed by several factors including the user interface, personal skills and the nature of the task[113].

Task-based evaluation has been selected for this experiment. It is a convenient approach which reflects the combinational models presented in the PDC combination theory (Section 3.3.2). The adaptive combinational model describes the construction of a perception based on a specific task. By using task based evaluation the cognitive model built for the task and the perception is effectively measured.

Two approaches have been used to measure the effectiveness of the PDCs in each systems. One is metric based, measuring accuracy and time to perform the task[20], while the other is an observation technique. While the metric-based measures are simple to analyse, they do not capture other aspects of the task such as the strategy developed to complete the task. Both prototype systems include the functionality needed to record and playback user interaction events.

The user actions were analysed during the experiments and again after, using the recorded events. The subjects were monitored during the experiments and notes were made on their performance and on any questions and comments they had. A discussion

with the subject at the end of the experiment was used to confirm the notes and provided additional detail.

6.3 Participants

Twenty four participants were recruited for the experiments. The participants were unfamiliar with the prototype systems, none had practise with stereoscopic viewing and none had any formal medical training. Lay-participants were chosen based on the desire to have a common level of expertise. In addition, this decision was supported by the fact that it is "commonly accepted in the field of psychology that visual perception, including stereo vision, is an innate ability"[134].

All the participants were computer literate and accustomed to using the standard input devices (keyboard and mouse). To avoid usability and user interface factors influencing the task performance, the subjects were given training for approximately ten minutes a day over a period of a week. They were trained in using both systems (see Appendix D for the training document) and in stereo viewing. The stereo viewing involved looking at a set of stereoscopic images for several minutes and was done to allow the subjects to become accustomed to the differences between the stereoscopic system and the natural viewing (see Section 4.1, with respect to the convergence-accommodation conflict).

6.4 Experimental tasks

The participants were asked to perform similar tasks for the experiments using the X-ray and volume visualisation prototypes. Given a set of feature points (targets), they were expected to mark the position of each target in the data as accurately and quickly as possible. The participants used a $3D$-cursor to specify the position of the targets in three dimensions. Once they were happy with the accuracy of the position, they placed a numbered marker. Positioning a marker required the participant to use the available depth information to make relative depth judgements between the target and the cursor or marker.

A model of a human head ("Elliot") was chosen as the subject for the experimental data. A human head is an everyday familiarity to which the participants can relate their knowledge of size and shape to assist them in their task. In addition, the basic structures of the head are part of common vocabulary which makes describing targets easier. The targets were described in a table and indicated on a figure by circling the regions. See Appendix C for the cover sheet and instructions sheets given out at the beginning of each of the experiments.

Simple targets, appropriate for the non medically specialised participants, were selected for the experiments. The targets were chosen carefully. For example, targets close together were used to test depth perception acuity, while objects at the depth extreme were selected to test general accuracy. The data used in both the visualisation systems was captured from the same set of scans. Unfortunately, the data are not entirely the same as the reconstructed (volumetric) data has been cropped, excluding the upper and lower regions. Consequently, some of the targets differ between the X-ray and volume visualisation experiments.

6.5 Experimental procedure

For the experiments the participants were divided randomly and evenly into three categories. The category they fell into decided their test case and which augmented PDCs were available in the visualisation systems. The test cases were: stereopsis only, motion only and both motion and stereopsis.

Participants were tested in both the X-ray and volume visualisation systems. They had no prior knowledge of the displayed data. The order in which experiments were performed was randomised to negate the effects of data set learning. At the beginning of each experiment the participants were given the opportunity to refresh their memory of the user interface. They were then given the instruction sheet describing the task. Before beginning the task, each participant was prompted for any problems and reminded to perform the task as quickly and as accurately as possible. During the experiment, the next instruction the participant was expected to follow was read out loud to save the viewer from looking down and breaking away from the perceptual task. Each experiment ran for approximately thirty minutes. At the end of each experiment, the participant was invited to make comments. Between the first and second experiment, the participants were allowed to rest, in order to prevent fatigue.

Each participant was instructed to wear the shutter glasses, regardless of the allocated test case. The shutter glasses reduce the contrast and brightness due to the partial filtering of light going to each eye. The difference in luminance has been proven to give different response times and accuracy [52]. In addition, wearing the glasses at all times imposes the same field of view restrictions and the same level of discomfort for all the participants.

6.6 Equipment and configuration

All the experiments were run on the same machine. The machine was setup in a small, quiet room with soft white non-fluorescent lighting. The chair, desk and monitor were positioned

at approximately the same place for each subject. The accuracy of the positioning was changed for the subjects' comfort. These environment settings were kept as constant as possible, as per the recommendations in [52].

The machine used had the following specifications:

- Athlon 500MHz Processor
- 384 MBytes of DIMM memory (100MHz)
- An Asus GeForce 3, with 64 MBytes video memory and stereo shutter glasses support
- An Epox motherboard, with support for AGP 2X
- A standard 15 inch monitor.

The equipment was set up to minimise viewer fatigue when using the stereo glasses. The display was set to a resolution of 800x600, with a refresh rate of 108 Hz. This is the highest combined resolution and refresh rate possible on the available monitor. The refresh rate is the limiting factor, as the display begins to flicker noticeably as the value drops. The monitor was set as bright as possible, while minimising the ghosting artifacts for the data. These settings were retained for all the experiments.

6.7 Observation results

A summary of the notes made during the experiments is given below.

6.7.1 General comments

Subjects preferred or were more confident about their results when using the volume visualisation system. Only seven out of twenty-four subjects preferred the X-ray visualisation system (none of whom had the stereo only PDC case). Subjects gave various reasons for their personal preference. The two underlying themes were the freedom of control (reflecting the finding of earlier studies such as [20]) and "easiness" of the task. Only two subjects indicated that they were equally confident about their results in both system.

Even though subjects went through training, it was found that they worked more slowly and less accurately initially and improved during the course of the experiment.

Although, size and perspective cues were not being studied, they were naturally present in both systems. Nine subjects explicitly used this information to assist their depth judgements. The remaining subjects seemed to ignore or were unaware of these cues. Those

that did use these cues made it clear that they were not useful until several markers had been correctly positioned. This reflected the fact that these cues are used for relative depth judgements and require experience.

As far as work styles are concerned, subjects would either place the $3D$-cursor alongside the target for comparison or place it directly over the target. Once the initial position was selected, either the position would be marked and the marker's position adjusted, or the cursor would be moved and only once the final position was selected, would the point be marked. The later approach proved to be faster, as it avoided the marker selection step.

Subjects tested with motion cues tended to use the motion for checking marked positions. Seven out of sixteen subjects with motion cues attempted to estimate depth during motion. When checking and adjusting the marked position, subjects would either use a wide (*semi-orthogonal*) or narrow angular position to change between the view points. After estimating the required change in depth, subjects using the narrow angle view changes would return to the original viewpoint to make the adjustments. All of the subjects that preferred the X-ray system used the narrow angular separation between the view points. This approach proved to be the faster and more accurate, as the semi-orthogonal adjustments lead to unexpected changes in position. In other words, bad depth estimation.

Some subjects preferred either only positive or only negative parallax in the stereo images. Four subjects found the ASUS stereo glasses too big and had to hold them in place which affected their time results.

6.7.2 Volume visualisation system

Seven subjects used the semi-orthogonal viewpoint change only, while fourteen subjects used small viewpoint changes, four of which occasionally used semi-orthogonal viewpoint changes. Three subjects could not be classified as using either method particularly. Five subjects tried to ensure that they always worked in the same depth plane. However, this approach was not successful and was discarded after several bad depth judgements.

Three subjects out of eight found motion & stereo the most effective. One subject found motion only more effective than stereo only. Three out of the sixteen subjects with the stereo PDC brought the target regions to the foreground for marking, which is an indication of their preference for negative parallax. One subject had difficulty using the stereo cue and stated that it did not help at all. One subject reported a conflict between the depth information presented by stereo and motion, and was unsure which was more believable.

Two subjects said they used stereo to position the cursor quickly and used the other

available cues for fine adjustments. Nine subjects said they used occlusion to assist their depth judgments and seven of them said that they occasionally found occlusion a hinderance when the markers and cursor became obscured. While subjects used their familiarity with the human head to identify feature points, only two subjects explicitly used their knowledge of the structures to position the markers.

6.7.3 X-ray visualisation system

None of the subjects seemed to struggle with the lack of occlusion information. Six subjects did find judging depth difficult when multiple markers and the cursor were close to each other. Targets in complex regions, such as areas with large amounts of X-ray occlusion, proved problematic for all users, especially subjects with the stereo only case. In these situations the less complex surrounding regions were used along with the knowledge of structures of the head to estimate the depth.

Motion was generally used to verify depth judgments. The relative direction of motion between the cursor or marker and the target was used to indicate the differences in depth. Three out of the eight subjects, with motion only, commented on the ambiguity which occured. The ambiguity arose as a switching between the perception of which regions were in the foreground or background.

Subjects with motion & stereo said they found the depth provided by motion to be much stronger than stereo. Four out the sixteen subjects with stereo cues found the stereo depth effect broke down occasionally when examining complex regions. Surprisingly, eight subjects could completely ignore regions with large parallax (where diplopia would occur) provided that the target was not near to these regions and that the horopter was near to the target. The larger parallax values were selected by the subjects via the zoom and horizontal image separation settings.

Only one subject noticed the accommodation-convergence conflict in the stereo display. The same subject and one other said that motion helped to "focus" the data. This is a reflection of the HVS higher tolerance to parallax when the objects are under motion.

All the subjects, except one, adjusted the contrast and brightness of the images for optimal viewing of the current target.

6.8 Metric results

Statistical analysis [50] was used to compare the results of each system separately and together using the two metrics — time and accuracy. The time is measured in seconds.

System	PDC cases	Accuracy (distance)		Time (seconds)	
		Mean	Std dev	Mean	Std dev
X-ray	Stereo & Motion	0.07593	0.08821	43.63	14.99
	Motion Only	0.04193	0.02244	70.66	29.17
	Stereo Only	0.09961	0.04017	55.28	49.82
Volume	Stereo & Motion	0.03332	0.01034	68.43	19.13
	Motion Only	0.02598	0.00837	79.31	44.48
	Stereo Only	0.03640	0.02621	86.59	31.67

Table 4: The means and standard deviations of for the X-ray system and volume rendering system experiments. The statistics for each PDC case is shown for each of the metrics.

System	PDC cases	Pearson r Value	Significant
X-ray	Stereo & Motion	-0.37	n
	Motion Only	-0.71	y
	Stereo Only	-0.67	n
Volume	Stereo & Motion	-0.270	n
	Motion Only	-0.32	n
	Stereo Only	0.45	n

Table 5: Product moment correlation between time and accuracy for each of the experiment cases. The statistically significant results (at $p < .05000$) are marked.

The accuracy is measured using the distance between the reference markers and the user positioned markers. The value is calculated using the Euclidian distance. Both the data set coordinate systems are scaled to the the unit cube as to allow the measured accuracy to be compared between the systems. The data sets are give in Appendix E.

Table 4 compares the average distance for each marker and average time it took for the subjects to mark each point. The three PDC cases for each system are shown. Box plots for each case and each system is shown in Figure 54 for accuracy and in Figure 55 for time.

Table 5 gives the correlations between the time and accuracy for each PDC category in each system. Using the mean to rank the PDC yields the following results: For the X-rays system based on time (fastest to slowest) we have stereo, motion & stereo, and motion, while based on accuracy (most to least) we have motion, motion & stereo, and stereo. For the volume visualisation system based on time we have stereo & motion, stereo, and motion, while for accuracy we have motion, stereo and motion combined and stereo. ANOVA tests were used to check for significance in the ranking of the PDC cases for each of the systems. For X-rays system we have $F(2, 21) = .69049, p = .51236$ for distance and $F(2, 21) = 1.2399, p = .30976$ for time, while for the volume system we have $F(2, 21) = .79602, p = .46427$ for distance and $F(2, 21) = .59837, p = .55881$ for time. The results

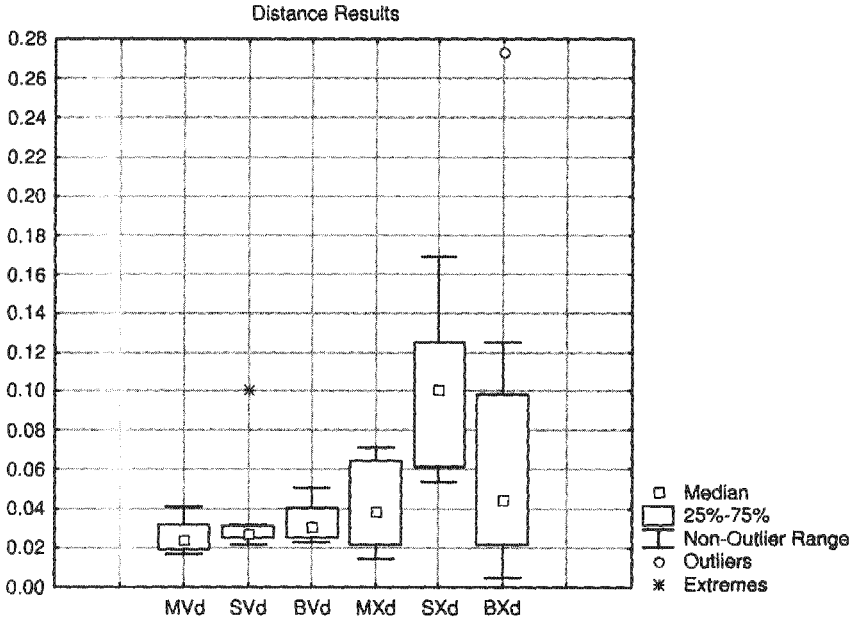


Figure 54: A box plot showing the median and quartile average distance results for each of cases in each of the systems. The cases are represented by the symbols. M = Motion, S = Stereo, B = Both (Stereo & Motion), V = Volume visualisation system, X = X-ray visualisation system and d is for distance.

indicate that the rankings are not statistically significant.

We compared each PDC case across the systems using a t-test for independent samples. The results are shown in Table 6. These results indicate that that there is a statistically significant difference between the systems for the Distance-Stereo and Time-Stereo and Motion combined cases. The remaining cases are statistically similar. Figure 56 summarises these results in a box plot.

Metric	PDC cases	t-value	p	F-ratio variance	p Variance	Significant
Distance	Stereo & Motion	-1.356	0.196	72.731	0.00001	n
	Motion Only	-1.883	0.080	7.1891	0.01848	n
	Stereo Only	-3.726	0.002	2.3500	0.28217	y
Time	Stereo & Motion	2.886	0.011	1.6292	0.53513	y
	Motion Only	0.460	0.652	2.3256	0.28793	n
	Stereo Only	1.499	0.155	2.4754	0.25473	n

Table 6: T-tests comparing the results between the systems for each PDC case using the different metrics.

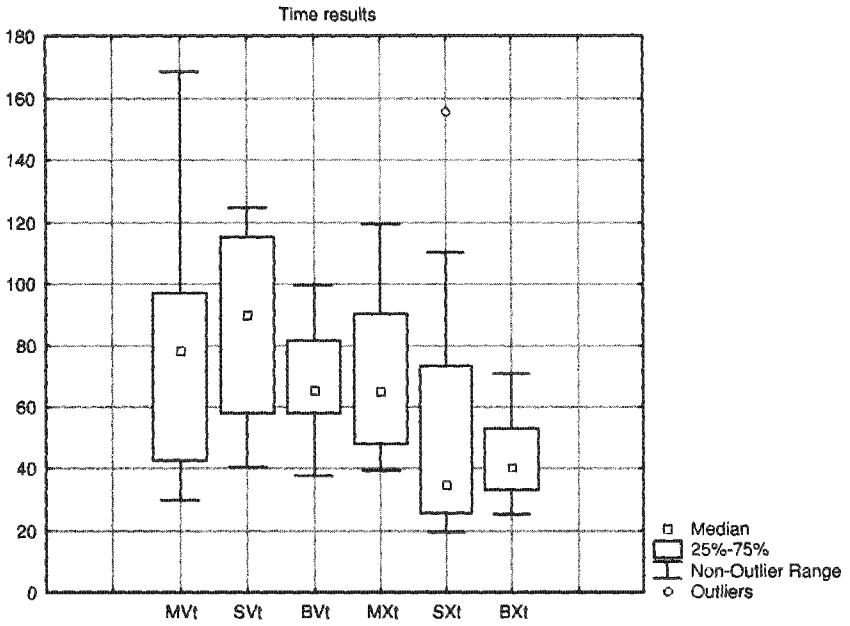


Figure 55: A box plot showing the median and quartile average time results for each of cases in each of the systems. The cases are represented by the symbols. M = Motion, S = Stereo, B = Both (Stereo & Motion), V = Volume visualisation system, X = X-ray visualisation system and t is for time.

6.9 Discussion

As a consequence of the lack of statistical significance between the ranked PDCs, other methods have to be used to evaluate the cues for the current experiment. The cues are ranked according to the least variance and best accuracy.

6.9.1 Accuracy results

It was expected that the X-ray system would be be consistently better than the volume visualisation system due to the greater resolution of the X-ray images and similar depth cues. However, this was not the case. The accuracy results are relatively poor across the board. For example, consider the best (unit normalised) average distance of 0.026, which is an error of 2.6%.

The highest accuracy was obtained using the X-ray visualisation system using stereo and motion combined (an error of 1.68%). However, this test case has the greatest variance. The large variance found in the X-rays system results compared to the volume system results

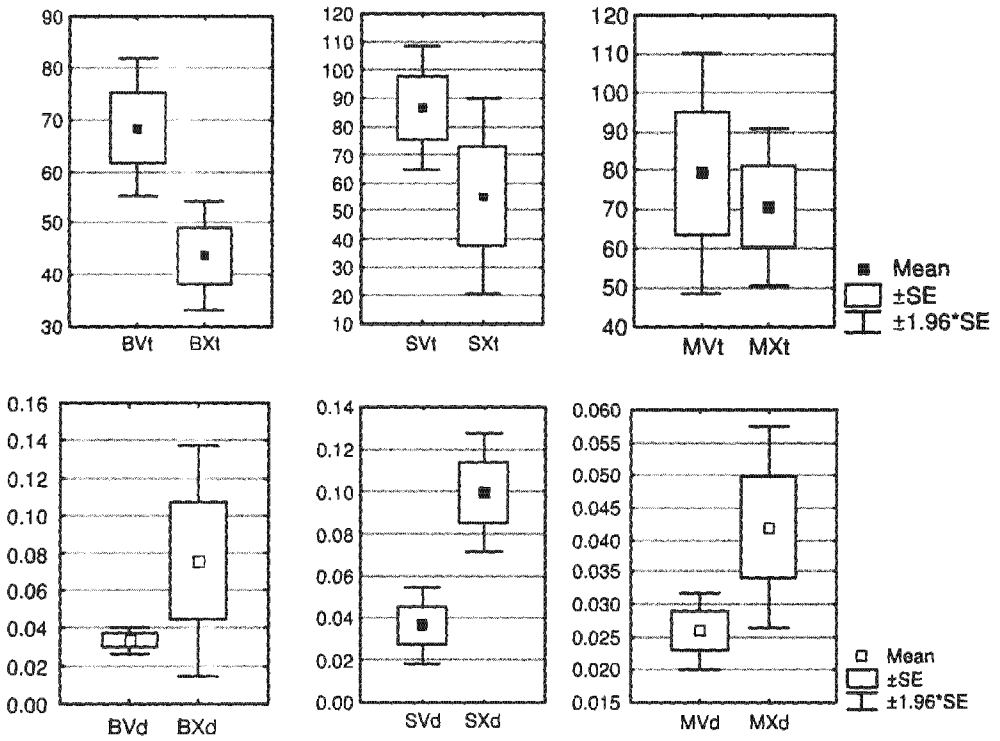


Figure 56: Box plots of the T-test results for cross system PDC case comparisons. On the top row we are comparing the time results and the on bottom the distance results. From left to right we are comparing Stereo & Motion, Stereo, and Motion. The symbols are the same as used in Figure 55

is a reflection of the preference of the subjects. It also indicates that the X-ray system is prone to coarse depth estimation.

The worst results are from the subjects using the X-ray system with stereo only. Stereo only depth judgements are notoriously difficult [99]. The problem with using stereo alone is that the depth information is relative and requires familiarity with the objects and their perceived sizes. In addition, stereo is used for close viewing. If the subject *perceives* the visualisation as being distant, the depth effect will be lessened. In the X-ray system the size consistency test partially fails due to the unorthodox projection. The difference in variance between the two systems is explained as a product of this failure, combined with the difference in occlusion information. The evidence would suggest that these cues reinforce the depth information, such that the subjects are more confident of their results.

6.9.2 Time results

The time results lie between 20 and 100 seconds for each point on average. One would expect the time and accuracy to be related. However, the correlations between distances and times are weak or statistically insignificant.

A possible explanation for the large variance in the experimental results are the inconsistencies between the subject's perception of distance (accuracy) versus speed. In other words, subjects have different perceptions of how accurate "accurate" is and how quickly they should be working. The lack of a strong correlation can be explained by the fact that corrections do not always reflect improved results due to over correction. Subjects decided at their own discretion whether to spend more time adjusting marked points.

Future experiments should rather use fixed time slots for marking each point. The best time slot length will have to be determined experimentally. From the results it can be suggested that 60, 80 and 100 seconds time slots should be tested.

The stereo X-ray system case has the fastest times. This is understandable as motion cues are derived from temporal integration. From the results, it would appear that the combined overlapping time taken for the individual cue test cases is related to the time taken for the combined cues test case.

6.9.3 General

From the discussions with the subjects and observations, it was found that subjects generally are unaware of PDCs and how they perceive depth. They can detect an increasing level of difficulty when performing tasks which require PDCs which are not present. The experimental task was chosen to require the estimation of relative positions. However, depending on what the target was, the task may have shifted to include judging surface shape and target surface detection. Referring back to Section 3.3.2 you will see these tasks have different dominant cues.

When the results are compared across the systems they were found to be statistically equivalent, except under the time metric using stereo and motion combined and under accuracy using stereo only. With respect to these two exceptions, the time performance for the X-ray system was higher, while the accuracy was higher in the volume system. The fact that these two systems are comparable in most circumstances is a positive result, when one considers that far fewer scans (10 X-ray images, at 3°) were used in the X-ray system, compared to the 60 images used to reconstruct the volume data. This indicates that it is possible to use the X-ray system for some 3D diagnostic tasks. In this way a CT scan and associated reconstruction could be avoided in some situations.

6.10 Summary

This chapter described the experiment design and evaluation of the PDCs enhanced visualisation systems. The aims of the experiments presented were to investigate which cues are the most effective, to test whether the X-ray and volume systems could be used to perform a 3D diagnostic task equally well and finally to rank the PDCs.

Two experiments were done, one for the volume rendering system, the other with the X-ray system. The participants were divided equally between the PDC test cases (stereo, motion, and stereo & motion). The participants were given the task of marking feature points in the data sets. The time taken and the accuracy of the marked points were used as metrics. The experiments were performed in a controlled environment and the subjects' activities monitored.

It was found that the variance in the results of the two metrics made it impossible to conclusively rank the PDC test cases based on the averages. It has been suggested that the best possible accuracy combined with the least variance should be used as an indication of how the PDCs cases should be ranked. In this case, for the X-ray system the PDCs are ranked as stereo and motion combined , motion and finally stereo. For the volume system, they would be ranked as motion, stereo, and stereo and motion combined. The time results were not considered, as the time and accuracy results are not correlated. When the systems were cross-compared, it was found that they were statistically equivalent in all but 2 cases. This result is positive when one considers that far fewer X-ray images are needed in the X-ray system compared to the number of images needed for a volume reconstruction. This implies that it may be possible, in some situations, to use the X-ray system for diagnostic purposes, rather than performing a CT reconstruction.

Chapter 7

Conclusion

X-rays are generally perceived as flat 2D images. Consequently, CT data is commonly used when 3D information is needed for a diagnosis. LACT and SACT reconstructions contain artifacts which limit their usefulness for such diagnoses.

The aim of this dissertation was to explore and enhance the visualisation of the data available from a LACT and SACT X-ray/CT system. In particular, the use of perceptual depth cues as an enhancement strategy was investigated. PDC's were used to create "3D X-rays". Also, they were used to allow the viewer to see beyond the artifacts in the LACT and SACT volume data and to extract useful information. A task based user test was developed and implemented to evaluate the enhancements. The test results were used to determine the effectiveness of selected individual depth cues and their combination, and cross compare the visualisation systems.

7.1 Results obtained

In this section the work done and the main results obtained through the course of this dissertation are summarised.

7.1.1 Literature review and PDC analysis

In search of an appropriate method to display LACT and SACT data, the volume visualisation literature was reviewed. An overview of PDC theory was provided and the implementation of PDCs in both volume rendering and X-ray systems surveyed. Both volume renderings and X-rays were analysed based on PDC theory. The analyses revealed which PDCs are "naturally" present and allowed it to be determined which PDC could be used to augment these cues.

It was found that between various past volume visualisation systems, most PDCs had been implemented. However, the X-ray PDC enhancements were far more limited due to the 2D nature of X-rays. The implementation of many of the PDC enhancements required 3D information. This implied the need for a reconstruction (in other words, volumetric data). Motion cues and binocular disparity were the two PDCs which could be implemented in both systems. A stereoscopic display and real time rendering (animation) were used to reproduce these cues. The implementation of motion cues was restricted to KDE.

7.1.2 System implementation

Several prototypes were implemented to test the various concepts developed throughout this dissertation. Surface and voxel-based volume visualisation were both evaluated in the context of LACT and SACT data. As part of the initial study, a technique to support real time isosurface extraction was developed. This interactive system resulted in motion cues which allowed the viewer to look beyond the reconstruction artifacts. However, as is typical with isosurface techniques, the "uncertainty of the surface" (a consequence of the reconstruction) could not be conveyed. Voxel-based visualisation techniques were found to provide the necessary flexibility to handle LACT and SACT data gracefully.

Several software and hardware volume rendering techniques were tested. A commodity hardware approach was adopted. Commodity 3D graphics accelerators are prevalent and low cost, while still supporting the real time rendering rates needed for motion cues in volume rendering.

A stereoscopic test system was built. This allowed for experimentation with the toe-in and off-axis stereo camera models, several 3D cursors and a variety of other stereoscopic system parameters. The established system parameters and software strategies were used in the final prototypes. Several stereoscopic presentation methods were evaluated, specifically anaglyph, chromostereopsis, shutter glasses and a HMD. Chromostereopsis was applied to volume rendering. This use of chromostereopsis is novel as it has only been used on polygonal models in the past.

An Open-GL based volume rendering library was implemented. The platform independent features of Open-GL were used. The available 2D texturing hardware was used extensively. The problems faced in implementing volume rendering in this manner were presented and the output characterised. The volume rendering library was coupled with an override stereoscopic video driver and used to create an application suitable for evaluating the PDC enhancements.

An X-ray visualisation system was developed and presented. The system constructed

stereoscopic animation sequences from multiple digital X-rays acquired on a fan beam scanner. The LODOX scanner was used as an example and a simple calibrated imaging model described. This model was used to augment the X-rays with computer graphic elements. The final prototype included a subsystem which pooled all the developed techniques to provide a PDC evaluation system similar to the volume visualisation prototype.

7.1.3 PDC enhancement evaluations

A set of experiments was designed and implemented to evaluate the effectiveness of the PDCs and their combination. The tests used the developed prototypes and required the completion of a $3D$ diagnostic task. The task involved the marking of feature points. A head data set with simple targets was used. The targets were appropriate for the participants and familiar to them. Refer to Appendix C for the experiment instruction sheets. The speed and accuracy at which the points were marked were used as metrics. In addition, the participants were observed during the experiments.

Lay-participants were trained to use the systems. This was used to eliminate the effects of previous experience, usability and user interface factors. The experiments were designed to be statistically balanced. The subjects were randomly and equally assigned to a test case in each system. The order of the tests were randomised to prevent data learning trends. The equipment and environment configurations were kept as constant as possible.

The results from the experiments were used to rank the effectiveness of the PDC enhancements and compare the two visualisation systems. While ranking based on the mean was not statistically significant, due to the large variance in the results, the results for the two system did prove to be statistically similar for most cases. These results imply that the X-ray system, which requires far fewer scans than needed for volume reconstruction, can be used to perform 3D marking equally as well. However, it should be noted the viewers were generally less enthusiastic about using the X-ray system. Having said that, it has been shown that it is possible to avoid a CT scan and associated reconstruction for some 3D diagnostic tasks.

7.2 Future work

There are a number of areas that could be researched further to extend the goals and themes explored in this dissertation.

7.2.1 The X-ray system

The X-ray system needs to be implemented in a live system and tested clinically. A concurrent sub-project of the LODOX system research group is examining the use of the system for mass TB (tuberculosis) screening. It would be worth investigating whether the stereo X-ray system could improve TB detection rates.

7.2.2 Volume rendering

As new hardware features and performance improvements become available, new techniques exploiting these feature will be needed. Even as this dissertation was being completed, new feature were being released. New features will soon make it possible to combine most of the PDC enhancements discussed in Chapter 3, while still achieving real time rendering.

Volume classification is a current research area. While several promising techniques are being developed, they all assume well sampled volumetric data. Consequently, research is required to find robust automatic or semi-automatic transfer function specification techniques which can handle LACT and SACT data sets. In addition, there are a host of application specific problems. For example, system integration, interfacing and repeatability of classification functions. Repeatability in this case refers to the ability of classifying two different volume data sets using a single transfer function and producing comparable results.

7.2.3 PDC studies

There is a need for more experimental data to evaluate PDCs to ensure replicability of the results. In the case of volume visualisation, additional PDCs could be implemented and tested. Shading and shadows may be the dominant cues if surface detection is the underlying perceptual task. Additional experiments are required to either refute or validate this further. Future experiments based on the experiment design presented in this dissertation should use fixed time slots for point marking. The optimal length for the time slots will have to be determined experimentally and may vary according to the task.

Appendix A

Acronyms and definitions

A.1 Acronyms

AGP Advanced Graphics Port

ANOVA Analysis of variance

API Application programmers interface

CT Computed Tomography

FOV Field of View

HSV Hue, Saturation and Value. An alternative colour model to RGB scheme

HVS Human Visual System

KDE Kinetic depth effect

LACT Limited Angle Computer Tomography

MDTF Multi-dimensional transfer function.

MIP Maximum or minimum intensity projection. One of the common volume compositing strategy

Open-GL A standard Graphics libraries

PDC Perceptual Depth Cues

RGB Colour specified as three channels. The channels represent 3 primaries: Red, Green and Blue

RGBa RGB with an additional channel a, or alpha, to indicate opacity level

SACT Sparse Angle Computed Tomography

SE Standard Errors of the mean (in the context of statistical analysis)

VRE Volume rendering equation

WAP Weighted additive projection. Another common volume compositing strategy

A.2 Definitions

Coherence - A similarity between items or entities. Coherence is frequently used in computer graphics to accelerate rendering algorithms. There are various levels at which coherence may be exploited — in pixel space, in object space, between rays and between frames (known as temporal coherence). Exactly how the coherence is exploited depends on the algorithms and their implementation.

Appendix B

Real time isosurface browsing

Real Time Isosurface Browsing

Caleb Lyness Edwin Blake

Collaborative Visual Computing Laboratory
University of Cape Town

Abstract

As volumetric datasets get larger, exploring the data sets becomes more difficult and tedious. Two approaches have previously been used to solve this problem: presentation of an abstraction of the data and acceleration of extraction and rendering of the data. We present a user centered approach which decouples the volume visualisation into two modes. Selection of the mode is done based on the user's actions. The first mode uses traditional isosurface rendering and extraction techniques and is applied when the user knows the isovalue of interest. The second mode uses a novel view dependent, sampling based isosurface rendering and extraction technique, which allows interactive browsing of the isosurfaces.

CR Categories: I.3.3 [Computer Graphics]: Pictures/Image Generation; I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism

Keywords: User centered approach, decoupling, isosurface browsing, volumetric data exploration

1 Introduction

Volumetric data sets, which are common in many fields, are often visualised using isosurfaces. Isosurfaces are typically generated using a technique such as Marching Cubes [11]. The triangle mesh output from these techniques can be displayed using standard polygonal rendering. The mesh is an intermediate format, which can be manipulated and re-rendered from a different view with little cost.

While Marching Cubes is very effective, it does, have a few problems. Some of the problems have been resolved, for example the ambiguities in choosing triangles to represent voxel intersections [12, 18]. Other problems, such as the selection of isosurfaces, have yet to be solved satisfactorily. If the desired isosurface's isovalue is known in advance, the algorithm can be run once with an acceptable delay between value selection and visible output. However, when the isovalue is unknown and the viewer wishes to explore the data, this delay makes interaction difficult, especially as the volume grows in size. Several techniques have been proposed and reported in the literature to speed up isosurface extraction with varying degrees of success.

This sketch presents a user centered approach to isosurface viewing. The isosurface visualisation is separated into two stages based on the viewer's actions. One stage deals with the viewing of a selected isosurface, while the other the exploration of all possible isosurfaces from the current viewpoint. A novel view dependant, sampling based preprocess and isosurface extraction technique is presented, which allows interactive volume exploration when used in this setting.

Section 2 gives a review of the relevant literature, Section 3 discusses the decoupling scheme, Section 4 covers the real time isosurface implementation. Section 5 gives some timings and results. And Section 6 ties up with a conclusion and possible future work.

2 Background

The need to support the user in exploring volumetric data has been recognised by [4]. They proposed an abstraction based on Hyper Reeb graphs. Their system uses the topological changes of the isosurface within the volume to select isovalues of interest. The abstraction seems very effective. However, the authors do not report results for volumes consisting of complex topologies, which may cause the abstractions usefulness to break down.

Prior to the above abstraction, a more direct approach of accelerating the extraction and rendering of isosurfaces was attempted. Rendering acceleration was achieved by reorganising the triangle mesh to form strips [7]. Further, decimation techniques where used to reduce the number of rendered triangles, while remaining within an acceptable error bound [15, 14].

The extraction process is accelerated by reducing visits to voxels not containing the rendered isosurface. As there are usually far more empty voxels than voxels which contribute to the isosurface, the speed increase is often significant. The accelerated extraction algorithms may be catagorised as belonging to one of two groups: seed based and range based.

Seed based algorithms make use of a list of isovalues and a set of starting voxels. From these starting points the isosurface is constructed via propagation to adjacent cells. Seed based techniques typically have a preprocess of $O(n)$, where n is the number of voxels [16, 1, 8].

Range based methods can be considered part of the space partition group of algorithms. The first reported technique was based on octrees [19]. Later techniques more suited to unstructured grids where developed [5]. These techniques group the voxel data into buckets. A similar approach [6] sorted the buckets on the maximum and minimum isovalues and took advantage of isosurface coherence to accelerate the rendering of isosurfaces whose isovalues where close. These algorithms where reported to have a run time of $O(n)$ and require a preprocess of similar order. More recently [10, 2] kd-trees were used to reduce the complexity to $O(\sqrt{n} + k)$. k is the number of contributing voxels.

Adaptive reconstruction of isosurfaces using a modified octree structure (average pyramids) was used in [17] to provide real time exploration. The typical cracking and sampling problems associated with adaptive reconstruction were resolved, providing a view independent solution. The algorithm described provides real time exploration by trading off quality for speed and requires the user to

specify a point of interest, around which higher detail rendering is done.

Recent developments in high end consumer hardware, specifically Nvidia's GeForce family of graphics cards, has renewed interest in accelerating isosurface rendering. Using clever techniques and the new features on these graphics cards has allowed interactive rendering of isosurface and direct volume rendering [13, 3]. These techniques are firmly tied to the underlying hardware and architecture.

Rather than relying heavily on hardware, we have adopted a decoupling approach. The idea of decoupling a problem can be found in volume rendering literature [9]. In his thesis work Lacroute suggested the use of two different data structures for volume rendering. Each data structure being optimal for the different task — classification and rendering of the voxels.

3 The decoupling scheme

The proposed decoupling follows a natural separation of user actions when exploring volumetric data. Two distinct modes of operation become clear when observing a subject using standard isosurface rendering techniques. Generally a subject will start with an arbitrary isovalue. If the generated isosurface is of interest they will examine it from several different views, otherwise another isovalue is selected based on the observed isosurface. This process continues until an interesting isosurface is found.

The decoupling allows two separate algorithms to be used. Figure 1 illustrates the decoupling and how user actions change the modes and associated algorithms. The mode switching occurs as transparently as possible.The switching occurs based on what actions the user applies to the interface — dragging the trackbar versus trying to rotate the object.

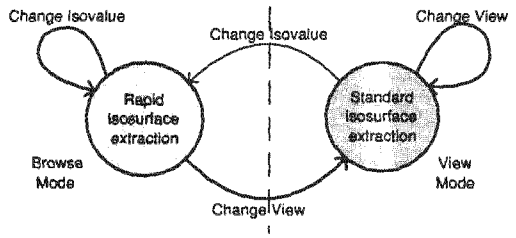


Figure 1: Isosurface rendering is decoupled into two states. User actions control which state system is in and thus which algorithm is run.

The current system uses an implementation of the marching cubes algorithm for displaying a selected isosurface from an arbitrary view point (View Mode in Figure 1). Note that any of the acceleration techniques mentioned in the Background section can be applied. For isosurface browsing (Browse Mode in Figure 1), a modified ray casting algorithm is used. This technique is presented in the next section.

4 Isosurface browsing

Rapid isosurface extraction allows the viewer to find isosurfaces of interest interactively. The rapid extraction is achieved by reducing the amount of data and reordering it with respect to the isovalues.

The algorithm is based on ray casting and runs in two stages. The first stage runs as a preprocess for a selected view. It involves calculating and storing the first occurrence of an isovalue and its distance along a ray from the viewpoint. The second stage is run every time the user selects a new isovalue and is responsible for generating the mesh which approximates the isosurface. A more detailed description of the stages follows.

4.1 Stage 1: Preprocessing

Isosurface construction techniques assume volumetric data is a sampled representation of a continuous field. Calculating the first occurrence of an isovalue along a ray can be described more accurately as finding the first interval in which the isovalue lies. Each interval is described as a pair of samples in the data.

The preprocess samples the volume by firing a set of rays from the viewpoint. As a ray is fired into the volume it is intersected with the volume slices in the x,y and z planes. A sample is extracted for each intersection. Every pair of samples in the volume form an interval. As the ray moves away from the eye, there is greater and greater chance that a new interval contains no new iosvalues and can thus be discarded. The discarding of the redundant information reduces the amount of data considerably.

As new intervals are constructed they are added into a list by insert sorting, based on the first unique isovalue they contain — see Figure 2. As the list reorders the samples, the distance from the viewpoint is calculated and stored. However to avoid the expense of calculating the full distance, only the squared distance is calculated. Once the final, reduced set of intervals has been found, the true distance is calculated along with an approximation of the normal at the sample points.

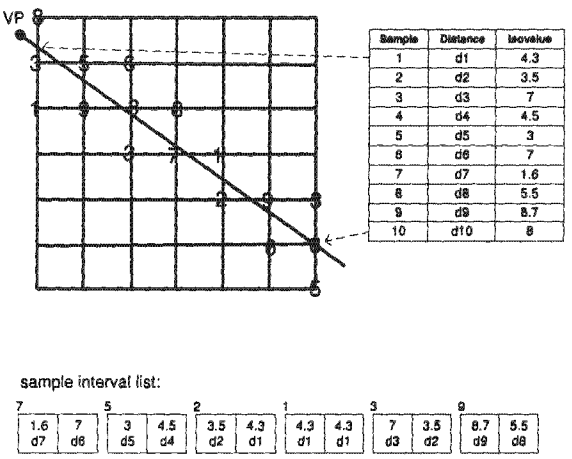


Figure 2: During the preprocess stage, rays are fired into the volume. The first occurrence of the isovalues and their distances are calculated. The table to the right indicates the sample at each grid-ray intersection, along with the associated distance (d_i) and isovalues, the final sample interval set is shown beneath. Each block represents the sample interval, with the top, left value representing the unique, sorted isovalue and the the right value the sample partner.

4.2 Stage 2: Extracting and rendering Isosurfaces

With the set of sorted sample intervals associated with each ray fired into the volume, extracting an isosurface becomes a problem of searching the set of intervals. Once the interval which contains the user specified isovalue is found, the distance and normal are calculated using linear interpolation.

Triangle meshes are constructed from the information extracted from each ray. The approximated, triangle mesh isosurface is then

rendered using OpenGL. The left image in Figure 3 show the extracted points used to construct a triangle mesh.

5 Results

Some initial timings for the algorithms running on a Pentium II, 400Mhz with 256Mb of memory and a Riva TNT2 video card are given in Figure 4.

Volume size	128x128x93
Marching Cubes	3,515 ms
Mesh Rendering	4,907 ms
Preprocess	22,111 ms
Extraction	290 ms
Mesh Rendering	350 ms

Figure 4: The average number of clock ticks spent running marching cubes, the preprocess and the extraction stages and the time to render the constructed mesh.

The short extraction time indicates that real time browsing of the volume is possible. For the 128x128x93 volume 3.4 extractions per second are possible and we can construct the mesh at rate of 2.8 per second. This means that on average 1.5 new isosurface can be displayed per seconds. If one tried to use marching cubes for real time isosurface browsing, you would be able to display 0.11 isosurfaces per second, in other words it takes 9 seconds to generate 1 isosurface.

The reduced rendering times associated with the isosurface browsing technique can be attributed to the reduced number of triangles and vertices rendered, a side effect of the hidden face removal.

6 Conclusions and possible future work

We have presented an alternative technique to support a user exploring volumetric data. The visualisation process is split into two modes, based on observed user actions — a browsing mode and a viewing mode. In the viewing mode traditional isosurface extraction and rendering techniques are used, while in browsing mode a new techniques, which was presented, offers interactive isosurface extraction and display rates of 1.5 isosurfaces per second. These rates make it possible for the user to browse the data interactively.

While the initial results indicate that rapid browsing is possible, the time spent in the current preprocess is significant. Future work will be to reduce the run time of the preprocess. Two avenues of work exist:

1. Implementation of an alternate ray casting engine, which makes greater use of occlusion information to prevent expensive calculations, and
2. A reduction in the number of rays fired. Currently the system samples the volume with rays at half the size of the projected voxel size. However in large volumes this is often unnecessary as the volume is either too large to be viewed entirely or is too distant to discern the samples. In such situations screen space sampling would be very effective.

Further, to improve user interaction, modifications to support a progressive ray casting algorithm and extraction step would allow the user to interact with coarsely sampled data, while the in between samples are being completed.

The isosurface browsing algorithm is limited to the current viewpoint, and the preprocess needs to be rerun when a new viewpoint is selected. When used in the decoupled scheme presented, the effects of this limitation are reduced.

7 Acknowledgments

We would like to thank African Medical Imaging (AMI) Inc. for their generous support, as well as the MRC/UCT Medical Imaging Research Unit under which this project falls.. Further we would like to thank Shaun Nirenstein for providing much food for thought.

References

[1] C. L. Bajaj, V. Pascucci, and D. Schikore. Fast Isocontouring for Improved Interactivity. In *ACM Symp. Volume Visualization '96*, 1996.

[2] P. Cignoni, P. Marino, C. Montani, E. Puppo, and R. Scopigno. Speeding Up Isosurface Extraction Using Interval Trees. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):158–170, 1997.

[3] K. Engel, M. Kraus, and T. Ertl. High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading. Accepted for Siggraph/Eurographics Workshop on Graphics Hardware 2001, 2001.

[4] I. Fujishiro, Y. Takeshima, and et al. Volumetric Data Mining Using 3D Field Topology Analysis. *IEEE Computer Graphics and Applications: Visualization*, 20(5):46–51, September/October 2000.

[5] R. Gallagher. Span filter: an optimization scheme for volume visualization of large finite element models. In *Proceedings of Visualization 1991*, pages 68–75, San Diego, Los Alamitos, CA, October 22-25 1991. IEEE Computer Society Press.

[6] M. Giles and R. Haimes. Advanced interactive visualization for CFD. *Computing Systems in Engineering*, 1(10):51–62, 1990.

[7] C. T. Howie and E. H. Blake. The Mesh Propagation Algorithm for Isosurface Construction. *Computer Graphics Forum*, 13(3):65–74, 1994.

[8] T. Itoh and K. Koyamada. Automatic Isosurface Propagation Using an Extrema Graph and Sorted Boundary Cell Lists. *IEEE Transactions on Visualization and Computer Graphics*, 1(4):319–327, 1995.

[9] P. Lacroute and M. Levoy. Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation. In *Proceedings of SIGGRAPH '94*, pages 451–458, July 1994.

[10] Y. Livnat, H. Shen, and C. Johnson. A near optimal isosurface extraction algorithm for structure and unstructured grids. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):73–84.

[11] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In M. C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 163–169, 1987.

[12] P. Ning and J. Bloomenthal. An evaluation of implicit surface tilers. *IEEE Computer Graphics and Applications*, 13(6):33–41, 1993.

[13] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl. Interactive Volume Rendering on Standard PC Graphics Hardware Using Multi-Textures and Multi-Stage Rasterization. In *SIGGRAPH/EUROGRAPHICS Workshop On Graphics Hardware*, pages 109–118, August 21-22 2000.

Figure 3: Left: The output from marching cubes for a given isovalue. Middle: Another view of the same isosurface, in which a large portion of the mesh has been hidden by the front part. Right: The sampled point cloud extracted for construction of the triangle mesh during Stage 2 using the same isovalue. The image has been rotated to illustrate how the hidden surfaces have been excluded.

[14] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit, An Object-Oriented Approach To 3D Graphics*. Prentice Hall, 1996.

[15] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. *Computer Graphics*, 26(2):65–70, 1992.

[16] M. van Kreveld, R. van Oostrum, and C. Bajaj. Contour trees and small seed sets for isosurface traversal. In *SIGGRAPH'85*, 13th Annual ACM Symposium on Computational Geometry, pages 212–219, 1997.

[17] R. Westermann, L. Kobbelt, and T. Ertl. Real-time exploration of regular volume data by adaptive reconstruction of isosurfaces. *The Visual Computer*, 15(2):100–111, 1999. ISSN 0178-2789.

[18] J. Wilhelms and A. V. Gelder. Topological considerations in isosurface generation, extended abstract. *Computer Graphics, Special Issue on San Diego Workshop on Volume Visualization*, 24(5):79–86, 1990.

[19] J. Wilhelms and A. V. Gelder. Octrees for Faster Isosurface Generation. *ACM Transactions on Graphics*, 11(3):201–227, 1992.

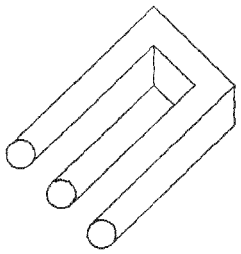
Appendix C

The experiment task

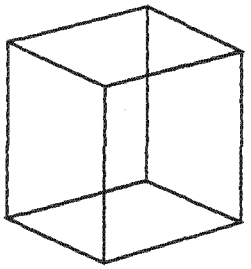
The experiment

Puzzles for the brain

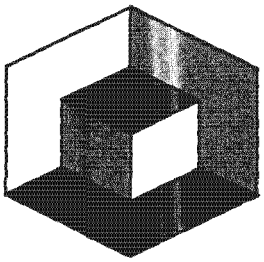
While you wait for your experiment session to take place, have a look at the following "optical illusions" – they are all the result of too little depth information being presented pictorially.



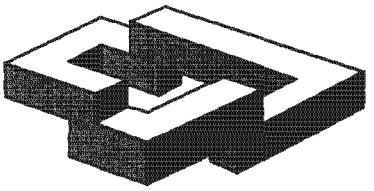
Can you make sense of this?



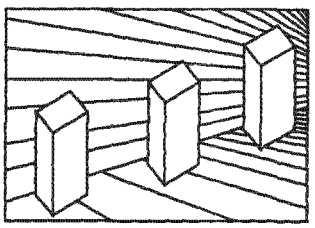
Which side is the front?



3 walls and a cube? Or a cube with a corner missing?



Where is the top of the stairs?

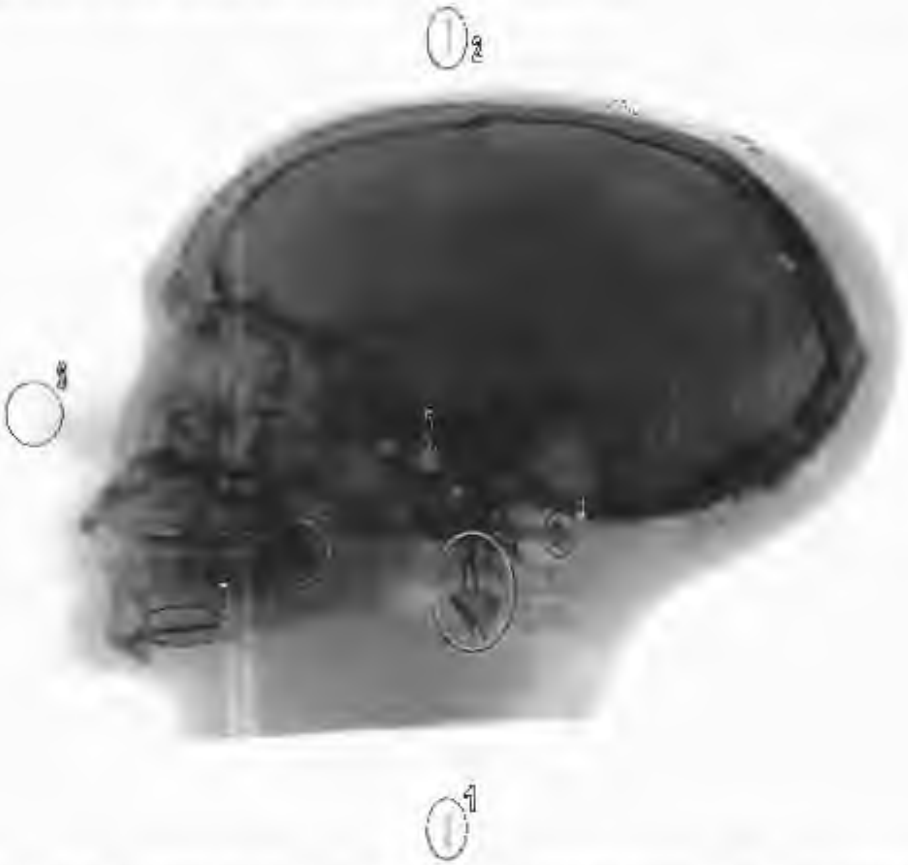


Smallest to biggest? Or the same size?

X-ray system experiment

Instructions:

At the start of the experiment the computer will generate a user ID for you. Please ensure that this allocated ID is filled in above. The test case you should use in this experiment has been marked above, please check that you are using the correct case.



The regions of interest are circled and labelled 1-8 in the diagram above. Using this diagram mark the objects of interest in the indicated regions according to the following:

Regions	Marking targets	Point #
1	The highest point of the nail	1
2	The lowest point of the nail	2
3	The point of each nail	3-6
4	The lowest and highest point of the section of the vertebrae indicated	7,8
5	The front upper corner of the jawbone (left and right)	9,10
6	Tip of the nose	11
7	Both ends of the pin	12,13
8	The roots of the last, bottom molars (left and right)	14,15

X-ray system experiment

Experiment details:

User ID: _____

Test case ☐Stereo ☐Motion ☐Stereo+Motion

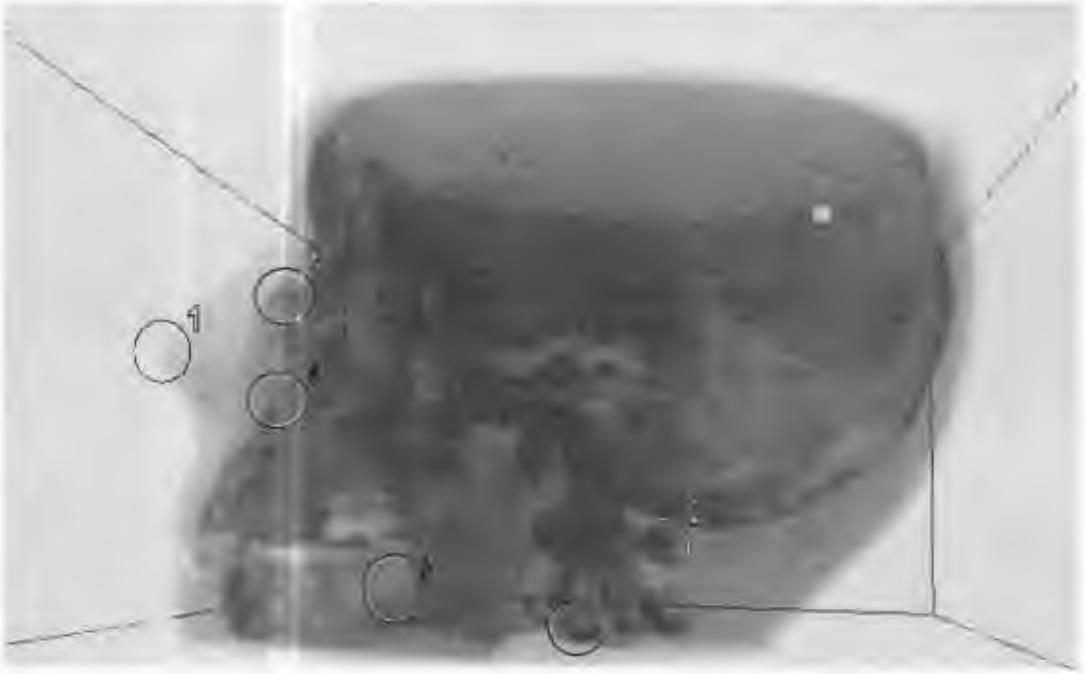
Feedback:

Please use this space to make any general comments:

Volume rendering system experiment

Instructions:

At the start of the experiment the computer will generate a user ID for you. Please ensure that this allocated ID is filled in above. The test case you should use in this experiment has been marked above, please check that you are using the correct case.



The regions of interest are circled and labelled 1-7 in the diagram above. Using this diagram mark the objects of interest in the indicated regions according to the following:

Regions	Marking targets	Point #
?	The 8 corners of the bounding box	1-8
1	The tip of the nose	9
2	The lowest and highest point of the vertebrae indicated	10,11
3	The roots of the last bottom molars (left and right)	12,13
4	Bump at the bottom of the nose cavity	14
5	Highest point of the nose cavity	15
6	The lowest point of the vertebrae	16

Volume rendering system experiment

Experiment details:

User ID: _____

Test case ☐Stereo ☐Motion ☐Stereo+Motion

Feedback:

Please use this space to make any general comments:

Appendix D

Experiments training document

Perceptual Depth Cues Experiments

Background

The experiments, in which you will be participating, have a perceptual and medical theme. Depth perception is our ability to judge the depth of objects in a scene. Perceptual depth cue theory examines the different elements of what we see and how we see in order to explain how we can accurately say that object X appears in front of or behind object Y.

These experiments study the use of two elements of perceptual depth cue theory: motion cues (movement of objects in the images) and stereopsis (stereo viewing). Stereo viewing provides separate images to the left and right eyes. Your brain merges the two slightly different images and extracts depth information.

The images, which will be presented to you during the experiments, are from two different modalities – x-rays and CT (or CAT). The tools used were developed for these experiments and to suite the type of data being viewed.

What is expected

The experiments and the tasks you are to perform are relatively easy and will involve marking points on objects in the displayed data. You will be told which objects and specifically which portions of the objects need to be marked. This process should be done as *accurately* and as *quickly* as possible. As an attempt to eliminate the effects of the user interface and stereo related problems from the experimental results, you are asked to come daily to the experiment room for a short (around 10 minutes) training session. During this time you will view stereoscopic images, which will train the eyes and brain for viewing computer generated stereo and give you the opportunity to familiarise yourself with the tool's interfaces.

Learning to use the systems

In this section I briefly describe each system and the interfaces.

The volume rendering system

The volume rendering test tool is used to display the CT/CAT data. The first interface you will see (right image) is used set up the parameter used by the volume renderer. During the final experiment the options will be limited, and only the "*Text Case*" options will need changing. The value you are to use will be allocated to you just before the experiment. During the training sessions you can choose which transfer function you want to use and the volume data by selecting one from the appropriate drop down list. The transfer function specifies how colours are assigned to the volume data. The volume data is the actual CT data. The names of the entries will give an indication of what it contains. (A little experimentation will help your understanding.) To begin the experiment you need only click on the *Ok* button. The volume rendering application will start

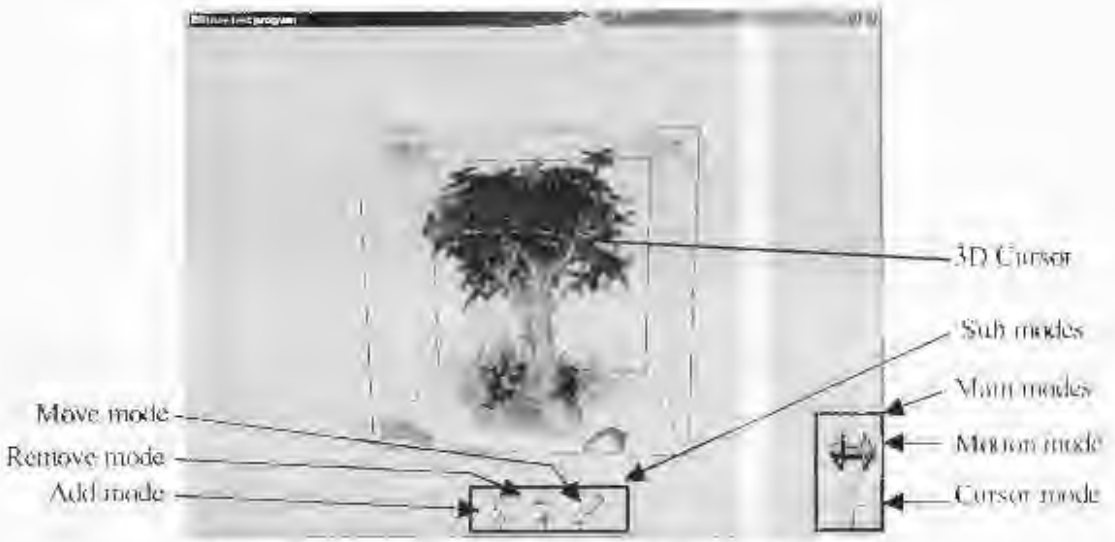


Figure 1 An overview of the volume rendering application

The volume rendering application has 2 main modes (indicated by the icons on the left). The modes are motion mode and cursor mode. When a mode is selected the associated icon becomes reddish. Clicking the right mouse button once will change between the modes. The initial mode is motion.

Motion mode allows you to adjust the orientation of the object. In this mode a small yellow square indicates the hand position. Holding the left button down and moving the mouse will rotate the object. Release the mouse button when you are done rotating. The amount of rotation depends on how far you drag the mouse, while the rotation direction depends on the starting and ending position of the motion. Once again a little experimentation will provide the most understanding.

When the cursor mode is active the cross hair will become white and the sub mode icons will appear along the bottom of the screen. From left to right the icons represent: Add a point marker, remove a point marker and move a point marker. The default mode is "add". Clicking the middle mouse button (the mouse wheel) changes the sub-modes. Moving the mouse will move the cursor across the screen. Turn the mouse wheel to move the cursor towards or away from you. (Moving your finger towards you will bring the cursor towards you and visa versa.) Clicking the left mouse button applies an action depending on the sub-mode (adding, removing, moving). When adding, a point marker will appear inside the cursor circle. When in removing or moving sub-mode, the point marker, which is closest to the cursor, will be highlighted yellow (left image). Left clicking the highlighted cursor will then remove or grab the point. If the point is grabbed it will follow the cursor, until you left click to release the point again.

Once you have completed the experiment press the escape key to exit the viewer.

The x-ray system

The x-ray system is similar to the volume rendering system in many aspects. The viewer has more main modes and makes more use of the keyboard. The viewing system is built into another system as one of the components, but you need not worry about the other parts of that system. Left is the

experiment dialog, which you can reach from the menu commands: *Test | User Test...* Unfortunately only a limited number of interesting data sets are available, but you can take your pick from the image dataset drop down list. The test case field has the same purpose as before. Click on the *Ok* button to start the experiment.

Figure 2: An overview of the x-ray viewer

The x-ray viewer has several main modes. The main mode icons are shown on the left. From top to bottom the modes are: zooming, contrast adjustment, 3D cursor and image movement. A differently coloured icon indicates the current mode (usually reddish). As before right clicking the mouse changes the mode. The default mode is the zoom mode. Sub-modes are available for Zooming and 3D cursor. The sub-modes are selected by clicking the middle mouse button (mouse wheel).

In zoom mode the sub-modes are, from left to right, zoom in, zoom out and actual size. Clicking the left mouse button will perform the appropriate zoom depending on the state. The current zoom factor is shown at the bottom left corner of screen.
The contrast mode, allows adjustment of the range of visible x-ray data. Holding the left or middle mouse button down will popup a range adjustment box.

Figure 3: Contrast adjustment popup

Dragging the mouse left and right while the left button is depressed will slide the range of values up and down. While depressing the middle button will and moving the mouse left and right will scale the range.

The cursor mode is the same as for the volume rendering tool. The same set of sub-modes will appear. There is one additional function: if the shift key is pressed during changes to the depth of the cursor (rolling the mouse wheel), a finer grain adjustment occurs.

The image movement mode allows adjustment of the distance between the left and right images, which affects the perceived depth of the images. This value should be adjusted to suit

your preference. Adjust the value by scrolling the mouse wheel. Scrolling the mouse wheel with the shift button down provides a finer grain adjustment.

In all the modes, with the exception of the cursor mode, moving the mouse will shift the image. In cursor mode, moving the mouse moves the cursor. The arrow keys can be used to move the image in all modes. If a test case with motion is selected, and the dataset allows it, pressing the space bar will switch the motion on or off.

When the experiment is done press the escape button to exit.

About the test cases

Both systems have a test case parameter. This test case selects which of: motion, stereo and motion+stereo are available during the experiment. Depending on which of the cases you select the display and functionality will differ slightly.

Volume rendering:

Motion:	No stereo will be available, but you can see the objects as they are moving.
Stereo:	Only stereo. You will not see the objects when they are moving.
Stereo+Motion:	Stereo is available and you will be able to see the objects as they move.

X-ray system:

Motion:	The x-ray is animated.
Stereo:	The x-ray is visible as a stereo image only.
Stereo+Motion:	The x-ray is visible as a stereo image and as an animation sequence.

Regardless of the case you are allocated you will have to wear the stereo glasses during the experiment.

Appendix E

Experiment data sets

Volume Rendering Results M=Motion Only S=Stereo Only B=Motion+Stereo x=xray v=volume

			Distance between marked points and reference markers (based on a unit cube)																
User Number	Test Case	xravol	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	sum	avg
1	M	v	0.011348	0.01518	0.022807	0.024722	0.022933	0.014279	0.009508	0.008949	0.012552	0.019677	0.030852	0.029848	0.022079	0.017572	0.021948	0.284255	0.01895
2	M	v	0.038294	0.033142	0.033219	0.010097	0.029812	0.040385	0.006587	0.024606	0.106078	0.086875	0.05146	0.071277	0.016207	0.037373	0.032599	0.618012	0.041201
3	S	v	0.00862	0.621774	0.016392	0.009551	0.00391	0.00887	0.034303	0.556911	0.040399	0.047661	0.053259	0.038692	0.029083	0.030995	0.010792	1.511212	0.100747
4	B	v	0.014547	0.009968	0.008014	0.006236	0.004033	0.005291	0.011236	0.012107	0.043187	0.045562	0.083952	0.059411	0.033893	0.025325	0.013493	0.376255	0.025084
5	M	v	0.006821	0.003892	0.018275	0.007346	0.007221	0.009057	0.014799	0.016384	0.06362	0.064164	0.017583	0.036291	0.043712	0.04374	0.029169	0.382074	0.025472
6	S	v	0.012588	0.014809	0.016612	0.016489	0.006906	0.009474	0.012498	0.027128	0.073478	0.073053	0.024688	0.059059	0.042123	0.028475	0.009705	0.427084	0.028472
7	B	v	0.031941	0.023487	0.036704	0.017683	0.049533	0.006556	0.047932	0.011071	0.107354	0.083208	0.085118	0.110207	0.093594	0.025508	0.027779	0.759675	0.050645
8	S	v	0.014166	0.035675	0.015116	0.004802	0.014953	0.009021	0.01122	0.029594	0.056059	0.057327	0.02932	0.05089	0.086058	0.029282	0.022988	0.466469	0.031098
9	M	v	0.014635	0.005805	0.012812	0.006827	0.006823	0.006229	0.008156	0.008347	0.04142	0.041372	0.014885	0.016651	0.012497	0.024938	0.029984	0.251381	0.016759
10	B	v	0.020411	0.004028	0.016028	0.050165	0.012898	0.013574	0.01484	0.063456	0.01431	0.018474	0.041025	0.109058	0.022775	0.036249	0.03145	0.468743	0.03125
11	M	v	0.02251	0.030624	0.01708	0.003485	0.035676	0.004023	0.014996	0.040842	0.098671	0.082057	0.019646	0.038441	0.019201	0.034516	0.032834	0.494601	0.032973
12	S	v	0.013166	0.013955	0.002406	0.004841	0.007589	0.004462	0.007831	0.015743	0.120355	0.084318	0.013321	0.012559	0.024249	0.024096	0.038868	0.387758	0.025851
13	M	v	0.014414	0.023972	0.016223	0.007254	0.003884	0.009247	0.012591	0.025432	0.033555	0.037956	0.014862	0.028102	0.022712	0.006709	0.035527	0.292441	0.019496
14	S	v	0.00955	0.01712	0.002945	0.016155	0.004709	0.01275	0.010112	0.007891	0.062458	0.06512	0.029737	0.012381	0.019144	0.03171	0.02731	0.329091	0.021939
15	S	v	0.020318	0.009005	0.023252	0.012165	0.007824	0.022677	0.009308	0.01924	0.023189	0.024908	0.095088	0.134815	0.020852	0.029048	0.030013	0.481704	0.032114
16	S	v	0.018641	0.005902	0.015788	0.00352	0.005735	0.003597	0.009685	0.021199	0.056674	0.056159	0.031677	0.072802	0.021473	0.029052	0.041691	0.393597	0.02624
17	B	v	0.026616	0.032682	0.026866	0.020676	0.01297	0.012821	0.010269	0.048332	0.031872	0.035576	0.055149	0.064462	0.015103	0.022731	0.0403	0.456425	0.030428
18	S	v	0.018409	0.015934	0.003502	0.011583	0.010635	0.00193	0.006301	0.01204	0.058653	0.052187	0.02324	0.050507	0.040525	0.028215	0.038006	0.371666	0.024778
19	B	v	0.015706	0.033184	0.02025	0.013414	0.018234	0.014319	0.015336	0.022372	0.038726	0.033122	0.015592	0.026865	0.004563	0.050812	0.054284	0.376781	0.025119
20	B	v	0.018927	0.007037	0.042167	0.01007	0.022749	0.00901	0.012695	0.017323	0.090477	0.090242	0.046027	0.039806	0.041142	0.045583	0.022206	0.51546	0.034364
21	M	v	0.010329	0.003135	0.006002	0.006672	0.00786	0.002199	0.011937	0.007467	0.00906	0.010841	0.049803	0.153857	0.016468	0.0089	0.032427	0.336955	0.022464
22	B	v	0.014885	0.008887	0.034509	0.006934	0.026137	0.00442	0.018807	0.041603	0.090276	0.112923	0.088888	0.080441	0.061755	0.080085	0.034166	0.704717	0.046981
23	B	v	0.014198	0.001427	0.011644	0.005734	0.003944	0.009007	0.01107	0.012521	0.046193	0.043042	0.057086	0.06292	0.007668	0.019909	0.033764	0.340127	0.022675
24	M	v	0.014924	0.025947	0.014179	0.016404	0.027678	0.037033	0.019779	0.044843	0.071857	0.059678	0.016877	0.013178	0.037823	0.037267	0.020784	0.458249	0.03055

Volume Rendering Results M=Motion Only S=Stereo Only B=Motion+Stereo x=xray v=volume

User Number	Test Case	xravol																
1	M	v	74	9	95	5	42	78	50	76	30	5	43	58	36	15	32	648	43.2
2	M	v	16.7	14.5	26.4	45.7	34	30.9	71.2	33.1	32.3	3.6	24.8	22.5	26.5	36.7	24.7	443.6	29.57333
3	S	v	231	175.1	308	128.3	169.1	112.1	93.2	157.7	153.2	7.4	135.1	45.9	76.2	40.2	23.7	1856.2	123.7467
4	B	v	154.8	73.6	131.8	83.8	61.8	67.1	57.1	65.7	164.6	19.4	74.5	64.6	190.9	176.8	107.7	1494.2	99.61333
5	M	v	92.4	83.1	42.7	21.7	36.2	33.9	16.7	6.4	79	15.7	14	13.1	7.7	81.3	89.3	633.2	42.21333
6	S	v	128.8	80.3	62.1	80	44.4	22.6	24	49.8	112.4	96.6	17.7	47	41.2	8.3	68.4	883.6	58.90667
7	B	v	36.2	188.5	41.5	37.4	47.2	54.9	57.7	98.1	32.9	21.7	68.2	46.2	57	36.8	48	872.3	58.15333
8	S	v	74.6	65.6	49.8	36.4	21.2	24.4	84.5	58.1	27.2	17.3	33.8	41.5	14.7	24.4	35.8	609.3	40.62
9	M	v	42.9	62.2	53.5	51.1	48.1	104.6	91.4	81.9	94.4	39.6	98	101.7	92.4	83.8	88.9	1134.5	75.63333
10	B	v	61	21.8	20.3	44.8	48.6	8.6	4.3	172.8	95.2	97.9	65.1	25.4	40.1	22.6	140.3	868.8	57.92
11	M	v	48.8	109.3	80.7	100.9	62.1	83.6	21.1	164.9	124.8	106.2	104.9	81.2	30.3	147.6	90.2	1356.6	90.44
12	S	v	40.8	47.5	72.3	89.8	31.1	38.3	33.7	60.2	91.5	12.3	85.2	49.6	75.9	70.4	61.2	859.8	57.32
13	M	v	139.5	87	166.5	107.9	69.4	126.6	215.2	86.6	73.4	76.2	48.4	32.5	222.7	57.7	55.2	1564.8	104.32
14	S	v	160.1	81.1	89	106.4	24.1	84.7	46.1	249.9	163.3	58.6	135.8	95.6	113.9	102.7	95.5	1606.8	107.12
15	S	v	14.8	329.4	74.4	110.2	89	112.1	51.7	98.5	177.9	78.4	111.4	366.9	133.6	51.4	70.9	1870.6	124.7067
16	S	v	78.7	88.3	60.6	50.8	54.7	37.6	79.8	134.4	154.5	23.3	91.3	74	275	72.5	54.8	1330.3	88.68667
17	B	v	97.9	47.6	24.6	36.6	23.3	19.8	39.7	49.7	54	11.9	43.5	29	39.6	12.7	34.7	564.6	37.64
18	S	v	64.2	241.5	91.7	71.2	86.1	62.1	57.2	98.4	217.4	30.5	118	38.5	84.5	60.5	52.6	1374.4	91.62667
19	B	v	146.9	41.9	249.7	121.4	54.4	88.4	97.3	64.7	134.3	24.9	80.1	72	54.9	11.6	49.2	1291.7	86.11333
20	B	v	114	51.4	30.6	96.5	36.7	36.8	40.3	96.9	32.8	8	96.9	48	109.5	78.6	49.8	926.8	61.78667
21	M	v	306.9	134.2	532.2	104	73.4	76.8	313.5	201.8	144.5	87.3	205	124.7	95.3	72.3	53.7	2525.6	168.3733
22	B	v	98.3	63	86.9	35.4	70.1	39.9	37.7	149.3	84.7	96.2	115.8	57.5	64.2	26.2	125.6	1150.8	76.72
23	B	v	98.7	14.5	55	67.6	74.9	77	90.7	123.1	83.9	36.5	63.2	56.3	90.1	46	65.9	1043.4	69.56
24	M	v	176.4	110.1	106.9	44.1	40	108.3	32	62.2	109	45.5	51.8	23.7	97.2	93.3	111.5	1212	80.8

Xray Results M=Motion Only S=Stereo Only B=Motion+Stereo x=xray v=volume

User Number	Test Case	xravol	Distance between marked points and reference markers (based on a unit cube)															
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	sum	avg
1	S	x	0.003596	0.060273	0.060324	0.014582	0.014232	0.036697	0.037177	0.052063	0.18561	0.010492	0.013999	0.050931	0.01598	0.36323	0.07043	0.989615	0.065974
2	S	x	0.018577	0.005849	0.005804	0.001414	0.004622	0.040291	0.014277	0.288188	0.060481	0.149534	0.010406	0.041632	0.019435	0.125574	0.059039	0.845124	0.056342
3	M	x	0.005105	0.009408	0.010046	0.005619	0.004074	0.072943	0.00217	0.202566	0.098134	0.019683	0.009392	0.01541	0.037155	0.065664	0.042482	0.599852	0.03999
4	B	x	0.001633	0.009396	0.009683	0.004905	0.001386	0.054381	0.003025	0.005466	0.215821	0.037682	0.016244	0.004835	0.033341	0.031302	0.032882	0.461982	0.030799
5	M	x	0.027895	0.023368	0.023368	0.013142	0.014963	0.049612	0.007947	0.043621	0.00802	0.067205	0.125427	0.064043	0.014639	0.027869	0.037171	0.54829	0.036553
6	B	x	0.000248	0.024857	0.025299	0.020133	0.016031	0.096917	0.043007	0.04968	0.023044	0.05122	0.0595	0.074973	0.020363	0.160137	0.051166	0.716576	0.047772
7	S	x	0.018531	0.050901	0.051108	0.074392	0.078986	0.077874	0.023473	0.41723	0.109245	0.219751	0.190137	0.203756	0.124947	0.116246	0.133696	1.890273	0.126018
8	B	x	0.037109	0.265302	0.265431	0.259768	0.254907	0.219463	0.273628	0.446201	0.212118	0.348973	3.022846	3.049339	0.273152	0.058112	0.688403	9.674752	0.644983
9	M	x	0.01888	0.005014	0.004708	0.002029	0.004556	0.058891	0.006177	0.009307	0.008235	0.010046	0.009651	0.003915	0.042166	0.022512	0.0144	0.220487	0.014699
10	M	x	0.018783	0.013847	0.005684	0.019148	0.014294	0.077291	0.023531	0.47397	0.024981	0.033215	0.023759	0.055483	0.006466	0.204694	0.075105	1.070249	0.07135
11	S	x	0.018446	0.005795	0.005895	0.001809	0.005851	0.058606	0.00459	0.306889	0.209523	0.295263	0.042013	0.074077	0.005356	0.373056	0.106825	1.513993	0.100933
12	M	x	0.018845	0.009857	0.009462	0.010536	0.006378	0.058892	0.009685	0.034411	0.028914	0.023416	0.006033	0.009239	0.032507	0.029977	0.020716	0.308869	0.020591
13	S	x	0.01867	0.013935	0.014145	0.002043	0.005079	0.077192	0.023264	0.102461	0.014709	0.106745	0.04185	0.055683	0.050887	0.222197	0.056168	0.805029	0.053669
14	M	x	0.01862	0.005427	0.001212	0.005789	0.004858	0.063659	0.009988	0.033228	0.024005	0.020312	0.007746	0.004721	0.051139	0.070651	0.023287	0.344643	0.022976
15	M	x	0.018685	0.005002	0.005002	0.00659	0.007105	0.058784	0.023388	0.312925	0.208655	0.005032	0.085489	0.017085	0.074797	0.011385	0.063172	0.903097	0.060206
16	B	x	0.001565	0.032493	0.032352	0.037092	0.041597	0.077207	0.023474	0.01102	0.02425	0.009495	0.004801	0.018707	0.023397	0.039956	0.026911	0.406315	0.027088
17	S	x	0.018446	0.037281	0.148333	0.03297	0.21291	0.122386	0.176405	0.104215	0.05142	0.151045	0.21469	0.004748	0.111251	0.348174	0.131987	1.866261	0.124417
18	B	x	0.018726	0.014234	0.014162	0.00632	0.011042	0.058751	0.006111	0.074933	0.075757	0.013897	0.004814	0.004813	0.070432	0.010016	0.028099	0.412106	0.027474
19	B	x	0.014066	0.002029	0.018706	0.002464	0.005107	0.0589	0.005107	0.092638	0.029347	0.024076	0.005655	0.009194	0.12546	0.046424	0.0327	0.471873	0.031458
20	B	x	0.001322	0.001119	0.019597	0.132489	0.028301	0.008865	0.047013	0.036675	0.074985	0.035822	0.04854	0.004953	0.056394	0.035082	0.040757	0.571915	0.038128
21	S	x	0.03713	0.014939	0.04196	0.03711	0.041717	0.169951	0.060235	0.788155	0.042291	0.051604	0.154342	0.148147	0.199013	0.575652	0.178855	2.541101	0.169407
22	B	x	0.003581	0.004738	0.013868	0.019753	0.015583	0.021982	0.032544	0.106414	0.227873	0.063941	0.033652	0.002506	0.004878	0.118288	0.051232	0.720831	0.048055
23	S	x	0.000686	0.050887	0.032515	0.037053	0.041693	0.077383	0.02347	0.362249	0.153063	0.294876	0.023327	0.055516	0.035163	0.207118	0.107255	1.502254	0.10015
24	M	x	0.004963	0.01857	0.042076	0.00581	0.004817	0.04483	0.052903	0.073276	0.462302	0.018978	0.009584	0.005842	0.111372	0.107553	0.073686	1.036582	0.069105

Xray Results M=Motion Only S=Stereo Only B=Motion+Stereo x=xray v=volume

User Number	Test Case	xravol																
1	S	x	143	62	52	352	73	20	64	50	202	202	109	157	24	85	62	1657	110.4667
2	S	x	17.7	27	37.6	6.5	3.1	2.5	17	3	39.6	53.6	126.1	50.8	7.8	58.3	62.5	513.1	34.20667
3	M	x	70.2	144.7	70.5	32.9	63.5	5.7	31.3	12.2	233.4	89.1	74.3	28.3	45.7	32	48.8	982.6	65.50667
4	B	x	20.9	5.6	58.9	2.5	14.6	2.2	14.6	6.9	99.9	34.5	60.5	66.8	36.4	15.7	70.9	510.9	34.06
5	M	x	56.2	29.3	50.8	5.1	83.5	19.1	12.9	8	45.4	24.6	26.9	72.2	59.7	81.4	11.8	586.9	39.12667
6	B	x	22.1	20.7	51.6	0.8	1.9	0.2	7.8	4.1	134.4	55.9	37.2	41.2	41.4	101.8	46.5	567.6	37.84
7	S	x	14.8	33.7	29	13.2	27.1	4.9	11.9	9.3	72.6	18.5	20.7	24.1	11.5	26.1	37.4	354.8	23.65333
8	B	x	57.7	24.1	3.3	2	6.2	1.3	14.8	12.7	15.8	39.4	45	119.2	4.3	22.8	7.4	376	25.06667
9	M	x	95.6	70.9	148	60.1	98.6	73.3	74.9	25.4	101.3	114.2	303.3	108.7	171.5	93	79.2	1618	107.8667
10	M	x	63.4	18	36.3	22.9	20.7	40.1	25.4	13.3	58.4	38.7	65.9	31.7	8.3	89.7	68.2	601	40.06667
11	S	x	26.2	13.3	9.6	1.9	5.1	3.7	20.1	3.5	88.4	34.7	59.8	42.1	5.2	46.9	47.3	407.8	27.18667
12	M	x	377.1	278.8	179.8	4.6	86.9	10.6	27.3	43	91.5	86.6	136.2	222.9	108.7	84.4	53	1791.4	119.4267
13	S	x	111.2	30.4	139.2	41.5	101.8	30.5	83.8	32.1	76.8	396.5	445.7	288.2	15.5	27.6	513	2333.8	155.5867
14	M	x	55.8	68.1	110.9	34.4	87.7	16.6	50.8	6.8	134.1	130.6	134.2	18	85.8	58.2	91.5	1083.5	72.23333
15	M	x	322.3	25.7	56.3	53.7	17.6	9.7	21.1	21.2	142.3	24.8	65.5	58.6	25.8	64.6	56.5	965.7	64.38
16	B	x	46.3	61.5	78.2	15.9	11.4	11.6	32.3	10.2	129	176.2	119.2	68.6	32	18.4	252.7	1063.5	70.9
17	S	x	19.7	12.6	43.2	62.7	9.5	15.8	22.1	12.4	66.1	45.1	112.6	39.6	14.4	28.8	40.7	545.3	36.35333
18	B	x	72.6	73	131.8	65.1	38.7	11.9	17.7	8.6	98	54.1	17.9	21	21.9	67.3	40.7	740.3	49.35333
19	B	x	198.1	52.4	31.5	55.4	96.8	28.8	34.6	13.2	125.4	35.5	44.2	34.4	18.9	45.7	41.3	856.2	57.08
20	B	x	15.6	18.8	67.7	44.8	16.7	80.9	15.2	12.3	50.8	12.8	46.3	13.4	13.9	26.2	37.7	473.1	31.54
21	S	x	21.3	13	7.9	10.3	14.7	10.7	10	6.8	36.7	14.1	14.4	26.6	22.5	22.9	59	290.9	19.39333
22	B	x	59.7	22.5	42	18.8	98.4	7.6	41.1	14.1	89.9	82.5	50.8	23.3	10.5	28.1	59.1	648.4	43.22667
23	S	x	40.1	16.8	43.4	35.3	7.6	6.9	20.5	14.9	60.9	5	36.3	32.9	9.7	32.7	168.6	531.6	35.44
24	M	x	72.7	44.3	83.5	102.7	79	4.6	91.8	20.2	74.7	5.3	82.8	5.6	24.8	152.8	5.9	850.7	56.71333

Bibliography

[1] Dresden 3D. Dresden3d: Welcome to dresden3d gmbh. World Wide Web, 2002. `http://www.dresden3d.com/` — Still available 27/05/2002.

[2] Stephen Adelson and Larry Hodges. Stereoscopic ray-tracing. *SIGGRAPH93 Course Notes*, pages 4–25–4–53, 1993.

[3] Taosong He adn Arie Kaufman. Fast stereo volume rendering. In *Proceedings Visualization '96*, pages 49 – 56, San Francisco, California, October - November 1996. ACM/SIGGRAPH, IEEE Computer Society Press.

[4] Anu. Chapter 6: Design Considerations for Augmented Telerobotic Control. World wide web, September 2001. `http://vered.rose.utoronto.ca/people/anu_dir/thesis/fchp6.fm.html`.

[5] Michael Baily and Dru Clark. Using ChromaDepth to Obtain Inexpensive Single-image Steroevision for Scientific Visualization. *Journal of graphics tools*, 3(3):1–9, 1998.

[6] C.G Barbor. Visual cues and pictorial limitation in photorealistic images. *ACM SIGGRAPH 1991 Course Notes*, pages 4–1–4–36, 1991.

[7] Paul Barham and David McAllister. A comparison of stereoscopic cursor for the interactive manipulation of b-splines. *SPIE Stereoscopic Displays and Applications II*, 1457:18–26, 1991.

[8] Uwe Behrens and Ralf Ratering. Adding shaddows to a Texture-Based Volume Renderer. In *IEEE Symposium on Volume Visualization*, pages 39–46. ACM/SIGGRAPH, IEEE Computer Society Press, October 1998.

[9] Wes Bethel. Visualisation viewpoints: Visualization dot com. *IEEE Computer Graphics And Applications*, pages 17–20, May/June 2000.

[10] M. Bolin and G. Meyer. A Perceptually Based Adaptive Sampling Algorithm. *Computer Graphics, Siggraph 99 proceedings*, 32:299–309, August 1999.

[11] Paul Bourke. 3d stereo rendering using opengl (and glut). World wide web, November 1999. `http://astronomy.swin.edu.au/~pbourke/opengl/stereogl/` — Updated May 2002.

[12] Martin Brady, Kenneth Jung, HT Nguyem, and Thinh Nguyen. Two phase perspective ray casting for interactive volume navigation. In Hans Hagen Roni Yagel, editor, *IEEE Visualisation '97*, pages 183–190. IEEE, IEEE Computer Society Press, November 1997.

[13] N. Bruno and J.E. Cutting. Minimodality and perception of layout. *Journal of Experimental Phsycology*, 117:161–170, 1988.

[14] Christoph Bungert. Crosstalk (ghost images). World wide web. `http://www.stereo3d.com/crosstlk.htm` — Still available 22/05/2002.

[15] B. Cabaral, N. Cam, and J. Foron. Accellerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware. In *ACM Symposium on Volume Visualization*, pages 310–316, May 1994.

[16] H.P. Chan, M.M. Goodsitt, K.L. Darner, J.M. Sullivan, L.M. Hadjiiski, N. Petrick, and B. Sahiner. Effects of stereoscopic imaging technique on depth discrimination. World wide web, 2000. `http://www.sunnybrook.on.ca/~iwdm2000/abstracts_html/Chapter1.htm`.

[17] D. Chelberg, J. Hsu, C. Babbs, Z. Pizlo, and E. Delp. Digital stereomammography. In *Excepta Medica International Congress Series 1069, York, England*, pages 181–190, July 1994.

[18] StereoGraphics Corporation. StereoGraphics Developers' Handbook: Background on Creating Images for CrystalEyes and SimulEyes. World wide web, 1997. `http://www.stereographics.com` — Still available 8/7/2002.

[19] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In *In ACM SIGGRAPH '93 Proceedings*, pages 135–142. ACM SIGGRAPH, IEEE Computer Society Press, 1993.

[20] Gina Crvarich. An exploration of techniques to improve relative distance judgments within an exocentric display. World wide web, 1995. `http://www.hitl.washington.edu/publications/crvarich/home.html` — Still Available 20/09/2002.

[21] Jas Mackenzie Davidson. Remarks on the value of stereoscopic photography and pathalogical appearances. *British Medical Journal*, pages 1669–1671, Decemeber 1898.

[22] E. Davis and L. Hodges. Human Stereopsis, Fusion and Stereoscopic Virtual Environments. In W. Barfield and T. Furness, editors, *Virtual Environments and Advanced Interface Design*, pages 145–174. Oxford: Oxford University Press, 1995.

[23] M. de Villiers. Limited Angle Tomography. Technical report: Digital image processing, University of Cape Town, DIP Lab, Department of Electrical Engineering, University of Cape Town, Cape Town, Rondebosch, 1999.

[24] M. de Villiers. Limited angle computed tomography using the lodox machine. Phd technical report: Digital image processing, "University of Cape Town", DIP Lab, Department of Electrical Engineering, University of Cape Town, Cape Town, Rondebosch, 2002.

[25] B. Dosher, G. Sperling, and S. Wurst. Trade-offs between stereopsis and proximity luminance covariance as determinants of percieved 3D structure. *Vision Research*, 26(6):973–990, 1986.

[26] R.A. Drebin and P. Hanrahan L. Carpenter. Volume Rendering. *Computer Graphics, Proc SIGGRAPH '88*, 22(4):67–74, 1988.

[27] S.R. Ellis and B.D. Adelstein. Visual performance and fatigue in see-through head mounted displays. World wide web. `http://duchamp.arc.nasa.gov/research/seethru_summary.html`.

[28] K. Engel, C. Rezk-Salama, M. Bauer, G. Greiner, and T. Ertl. Interactive Volume Rendering on Standard PC Graphics Hardware using Multi-Textures and Multi-Stage Rasterization. In *SIGGRAPH/Eurographics Workshop on Graphics Hardware 2000*, 2000.

[29] Klaus Engel, Martin Kraus, and Thomas Ertl. High-Qualitz Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading. In *SIGGRAPH/Eurographics Workshop on Graphics Hardware 2001*, 2001.

[30] J.P.O. Evans, M. Robinson, and S.X. Godber. 3-D X-ray image modelling-latest developments. In *European Conference on Security and Detection - ECOS97*, number 437, pages 28–30. IEEE, Apr 1997.

[31] J.P.O. Evans, M. Robinson, and D. Lacey. Testing and evaluation of an advanced (3-D) X-ray screening system. In *Proceedings IEEE 32nd Annual 1998 International Carnahan Conference on Security Technology*, number 98CH36209, pages 50–54. IEEE, October 1998.

[32] Shiofen Fang, Tom Biddlecome, and Mihran Tuceryan. Image-Based Tranfer Function Design for Data Exploration. In *Visualisation '98*, pages 319–326, October 1998.

[33] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practices.* Addison Wesley, Reading, Massachusetss, second edition in c edition, 1990.

[34] Issei Fujishiro, Taeko Azuma, and Yuriko Takeshima. Automatic Transfer Function Design for Comprehensible Volume Rendering Based on 3D Field Topological Analysis. In *Visualization '99*, pages 467–470. ACM/SIGGRAPH, IEEE Computer Society Press, October 1999.

[35] Youmei Ge. Noncontact 3d biological shape measurement from multiple views. Thesis, University of Western Australia, Computer Science Department, University of Western Australia, May 1994. `citeseer.ist.psu.edu/ge94noncontact.html`.

[36] B. Gilliam. The Perception of Spatial Layout from Static Optical Information. *Perception of Space and Motion*, pages 23–67, 1995.

[37] Earl Glynn. efg's Computer Lab: Spectra Lab Report. World wide web, November 1998. `http://www.efg2.com/Lab/ScienceAndEngineering/Spectra.htm`.

[38] E. Goldstein. *Sensation and Perception.* Wadsworth Publishing, Belmont, California, 3^{rd} edition edition, 1989.

[39] M.M. Goodsitt, H.P. Chan, J.M. Sullivan, K.L. Darner, and L.M. Hadjiiski. Evaluation of the effect of virtual cursor shape on depth measurements in digital stereomammograms. World wide web, 2000.

[40] C.W. Grant. *Visibility Algorithms in Image Synthesis.* Ph.d. dissertation, Davis, University of California, 1992.

[41] T. Günter, C. Poliwoda, C. Reinhart, J. Hesser, R. Männer, H. Meinzer, and H. Baur. VIRIM: A Massively Parallel Processor For Real-Time Volume Visualization in Medicine. In *9th Eurographics Workshop on Graphics Hardware*, pages 103–108, Oslo, Norway, 1994.

[42] Adel Hanna. Stereo viewing for Scientific Visualization. World wide web, Fall 1998. `http://www.nyu.edu/acf/pubs/connect/fall98/SciVisStereoFall98.html` — Available 06/09/2000.

[43] Harrison. Introduction to Depth Perception & Stereo Computer Graphics Concepts. In *SIGGRAPH 93 Course Notes: Stereo Computer Graphics with Applications to Virtual Reality*, pages 1–1–1–27, 1993.

[44] Helwig Hauser, Lukas Mroz, Gian Italo Bischi, and M. Eduard Gröller. Two-Level Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):242–252, July/September 2001.

[45] Taosong He, Lichan Hong, Arie Kaufman, and Hanspeter Pfister. Generation of Transfer Functions with Stochastic Search Techniques. In *IEEE Visualization '96*, pages 227–234, October 1996.

[46] Stephen Pizer Henry Fuchs, Marc Levoy. Interactive visualisation of 3d medical data. In *IEEE Symposium on Volume Visualization*, pages 140–145. IEEE Computer Society Press, August 1989.

[47] Hiroko. Computed tomography. World wide web, 7 1999. `http://epi.bk.tsukuba.ac.jp/~hiroki/html/ct.html` — Still Available 25/03/2003.

[48] L.F Hodges. Tutorial: Time-Multiplexed Stereoscopic Computer Graphics. *IEEE Computer Graphics and Applications*, 12(2):20–30, March 1992.

[49] L.F Hodges and D.F. McAllister. Computing stereographic views. In *ACM SIGGRAPH '89*, page Course Notes No 24, Stereographics, Boston, MA, July/August 1989. ACM.

[50] D. Howell. *Fundamental Statistics for the Behavioural Sciences.* Brooks/Cole, 2^{nd} edition edition, 1989.

[51] J. Hsu, C. Babbs, D. Chelberg, Z. Pizlo, and E. Delp. Pre-clinical ROC Studies of Digital Stereomammography. *IEEE Transactions on Medical Imaging*, 14(2):318–327,

June 1995. A version is also available as TR-EE-93-47, Purdue University, November 1993.

[52] J. Hsu, Z. Pizlo, C. Babbs, D. Chelberg, and E. Delp. Design of studies to test the effectiveness of stereo imaging, truth of dare: is stereo viewing really better. *in Proc. SPIE 1994*, 2177:211–220, 1994.

[53] Jean Hsu, Ke Shen, Frank B. Venezia, Jr., David M. Chelberg, Leslie A. Geddes, Charles F. Babbs, and Edward J. Delp. Application of Stereo Techniques to Angiography: Qualitative and Quantitative Approaches. Technical report Unknown, School of Electrical Engineering, Biomedical Engineering Center, Purdue University, Purdue University, West Lfayette, IN 47907, 1994.

[54] Jian Huang, Naeem Shareef Klaus Mueller, and Roger Crawfis. VOXBLT: An Efficient and High Quality Splat Primitive. In *IEEE Visualiyation 1999 Late Breaking Hot Topics Proceedings*, pages 1–4. IEEE, October 1999.

[55] Alex Huk. Seeing in 3d. World wide web, 7 1999. `http://www-psych.stanford.edu/~lera/psych115s/notes/lecture8/` — Still Available 20/09/2001.

[56] Immersive Autostereoscopic Display for Mutual Telexistence: TWISTER I (Telexustence Wide-angle Immersive STEReoscopic model I. Yutaka kunita and naoko ogawa and atsushi sakuma and masahiko inami and taro maeda and susumu tachi. *Proc. IEEE VR 2001, Yokohama, Japan*, pages 31–36, 2001.

[57] Kitware Inc. Kitware VTK. World wide web. `http://www.kitware.com/vtk/index.html`.

[58] T. Inoue and H. Ohzu. Measurement of human factors of 3d images on a large screen. *Large Screen Projection Displays II, SPIE*, 255, 1990.

[59] V. Interrante, H. Fuchs, and S. Pizer. Conveying the 3D Shape of Smoothly Curving Transparent Surfaces via Texture. In *IEEE Transactions on Visualization and Computer Graphics*, volume 3, pages ISSN 1077–2626, April-June 1997.

[60] J. Johnson. Analysis of Image Forming Systems. In *Image Intensifier Symposium*, pages 249–273, 1958.

[61] B Julesz. *Foundation of cyclopedian perception.* University of Chicago Press, Chicago, 1971.

[62] James T. Kajiya. The rendering equation. *Computer Graphics (SIGGRAPH '86 Proceedings)*, 20:143–150, August 1986.

[63] J.T. Kajiya and B.P. Von Herzen. Ray tracing volume densities. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 165–174, 1984.

[64] Adam Kalai. Improved rendering of parallax panoramagrams for a time-multiplexed autostereoscopic display. citeseer.nj.nec.com/15063.html.

[65] Karl-Heinz Hoehne, et al. Voxel-Man 3D Navigator Inner Organs. Springer, Electronic Media Eddition, 2000.

[66] A. Kaufman. Real-time accelerators for volume rendering (panel). In *Proceedings of the conference on Visualization '96*, pages 445–447. IEEE Computer Society Press, 1996.

[67] A.E. Kaufman, M. Brady, W.E. Lorensen, F.L. Kitson, and H. Pfister. Why is real-time volume rendering no longer a year away? In *IEEE Visualization 1998*, pages 497–499, 1998.

[68] Gordon Kindlmann and James W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *IEEE Symposium on Volume Visualization*, pages 79–86. ACM/SIGGRAPH, IEEE Computer Society Press, October 1998.

[69] Gordon Kindlmann, Joe Kniss, and Charles Hansen. Interactive Volume Rendering Using Multi-Dimensional Transfer Functions and Direct Manipulation Widgets. In *IEEE Symposium on Volume Visualization*, pages 79–86. ACM/SIGGRAPH, IEEE Computer Society Press, October 1998.

[70] Joe Kniss, Patrick McCormick, Allen McPherson, James Ahrens, Allan Keahey, and Charles Hansen. Interactive Texture-Based Volume Rendering for Large Dsata Sets. *IEEE Computer Graphics and Applications*, pages 52–61, July/August 2001.

[71] Yun-Mo Koo, Cheol-Hi Lee, and Yeong Gil Shin. Stereoscopic volume rendering using templates. World wide web.

[72] Joe Krahn, Flavio Villanustre, and Andreas Stenglein. stereogl documentation. SourceForge Project: `http://stereogl.sourceforge.net/`, December 2001.

[73] Kevin Kreeger and Arie Kaufman. Mixing translucent polygons with volumes. In *Proceedings Visualization '99*, pages 191–198. ACM SIGGRAPH, IEEE Computer Society Press, October 1999.

[74] Graphics Lab. The VolPack Volume Rendering Library. World wide web, January 1995. `http://graphics.stanford.edu/software/volpack/`.

[75] Philippe Lacroute. Fast volume rendering using a shear-warp factorization of the viewing tranformation. Thesis CSL-TR-95-678, Standford University, Computer Systems Laboratory, Department of Electrical Engineering and Computer Science, Stndford University, Stanford, CA 94305-4055, September 1995.

[76] R. Ledley and R. Frye. Processing of Stereo Image Pairs: Eliminating Depth Planes using the 'Cut-Plane' Procedure. In *Proc. SPIE — Stereoscopic Displays and Virtual Reality Systems*, volume 2177, pages 66–77, 1994.

[77] M. Levoy. Display of Surfaces from Volume Data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

[78] M. Levoy. Volume Rendering by Adaptive Refinement. *The Visual Computer*, 6(1):2–7, 1990.

[79] M. Levoy and R. Whitaker. Gaze-Direted Volume Rendering. *ACM Siggraph, speical ussue on 1990 Symposium on Interactive 3D Graphics*, 24(2):217–223, March 1990.

[80] Marc Levoy. A hybrid ray tracer for rendering polygon and volume data. In *IEEE Computer Graphics and Applications*, pages 34–40. ACM SIGGRAPH, March 1990.

[81] Marc Levoy. Volume Rendering using the Fourier Projection-Slice Theorem. In *Proceedings of Graphics Interfaces '92*, pages 61–69, 1992.

[82] Mark Levoy. Efficient ray tracing of volume data. *ACM Transactions on Graphics*, 3(9):245–261, 1990.

[83] H.S. Lim and T.O. Binford. Curved surface reconstruction using stereo correspondances. *Proceedings Image understanding workshop*, DARPA:808–819, April 1988.

[84] J. Lipscomb and W. Wooten. Reducing crosstalk between stereoscopic views. World wide web, December 1995. `http://www.research.ibm.com/people/l/lipscomb/www-ctalk.html` — Still available 23/05/2002.

[85] L. Lipton and M. Feldman. A new autostereoscopic display technology, The SynthaGram^{TM}. Word wide web, 2000. `http://www.stereographics.com/Downloads/whitepapers/SPIE2002b.pdf` — Still available 22/05/2002.

[86] William E. Lorensen and Harvey E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 163–169, 1987.

[87] Caleb Lyness and Edwin Blake. Real time isosurface browsing. In *"AFRIGRAPH 2001"*, pages 143–146. ACM SIGGRAPH, 2001.

[88] D.L MacAdam. Stereoscopic perception of size, shape, distance and direction. In *J. Soc. Motion Picture and Television Engineers 62*, pages 271–293, April 1954.

[89] Kathleen McCarthy. Chest Radiography Assessment in Nursing. World wide web, March 2002. `http://www.nursingceu.com/NCEU/courses/chest/` — Still Available 10/06/2004.

[90] H.C. McKay. *Three Dimensional Photography; Principles of Stereoscopy.* American Photography Book Department, New York, 1953.

[91] Michael Meißner, Ulrich Hoffmann, and Wofigang Straßer. Enabling Classificationa and Shading for 3D Texture Mappping based Volume rendering using OpenGL and Extensions. In *IEEE Visualization 1999*, pages 207–214, 1999.

[92] M.Meißner, U.Kanas, and W.Straßer. VIZARD II: A PCI card for Real-Time Volume Rendering. In *SIGGRAPH/EuroGraphics Workshop on Graphics Hardware*, pages 61–67, 1998.

[93] Torsten Moller, Raghu Machiraju, Klaus Mueller, and Roni Yagel. A comparison of normal estimation schemes. In *IEEE Visualization '97*, pages 19–26, 1997.

[94] Klaus Mueller, Naeem Shareef, Jian Huang, and Roger Crawfis. Ibr-assisted volume rendering. In Hans Hagen Amitabh Varshney, Craig M. Wittenbrink, editor, *IEEE Visualization 1999, Late Breaking Hot Topics*, pages 5–8. ACM/SIGGRAPH, IEEE Computer Society Press, October 1999.

[95] Paul Ning and Jules Bloomenthal. An evaluation of implicit surface tilers. *IEEE Computer Graphics and Applications*, 13(6):33–41, 1993.

[96] Nvidia Corp. Nvidia 3D Stereo Technical Brief. World Wide Web, 2002. `http://www.nvidia.com/docs/lo/1404/SUPP/TB-00271-3D_Stereo.pdf`.

[97] National Medical Library of Medicine. The Visible Human Project. Available online: `http://www.nlm.nih.gov/research/visible`, 1996.

[98] W. Pasman. Perceptual requirements and proposals for the ubicom augmented reality display. World wide web, 1997.

[99] J.D. Pfautz. *Depth Perception in Computer Graphics.* Degree of doctor of philosophy, Trinity College, University of Cambridge, England, May 2000.

[100] Hanspeter Pfister, Jan Hardenbergh, Jim Knittel, Hugh Lauer, and Larry Seiler. The volumepro real-time ray-casting system. In *Proceedging of the Siggraph 1999 annual conference on Computer graphics*, pages 251–260, Los Angeles, CA USA, August 1999. IEEE Computer Society, IEEE Computer Society Press.

[101] Hanspeter Pfister, Bill Lorensen, Chandrajit Bajaj, Gordon Kindlmann, Will Schroeder, List Sobierajski Avila, Ken Martin, Raghu Machiraju, and Jinho Lee. Visualisation viewpoints: The Transfer Function Bake-Off. *IEEE Computer Graphics and Applications*, 22:16–22, May/June 2001.

[102] Thomas Porter and Tom Duff. Compositing Digital Images. *Computer Graphics (SIGGRAPH '84 Proceedings*, 18:253–259, July 1984.

[103] PSY280F. DEPTH PERCEPTION. World wide web, 8 1999. `http://www.cquest.utoronto.ca/psych/psy280f/ch7/chapter7.html` — Still Available 12/10/2001.

[104] A Kaufman R. Yagel. Accelerating volume animation by space leaping. In Gregory M. Nielson and Dan Bergeron, editors, *Proceedings of the Visaualisation '93 Conference*, pages 62–69, San Jose, California, October 1993. IEEE Computer Society Press.

[105] Harvey Ray and Deborah Silver. RACEII: A State-Of-The-Art Volume Graphics And Volume Visualization Accelerator for PCs. In *IEEE Visualiyation 1999 Late Breaking Hot Topics Proceedings*, pages 9–12. IEEE, October 1999.

[106] Martin Reddy. Perceptually Optimised 3D Graphics. *IEEE Computer Graphics and Applications*, 21(5):68–75, September/October 2001.

[107] Christof Rezk-Salama. *Volume Rendering Techniques for General Purpose Graphics Hardware.* Phd thesis, University Erlangen-Nürnberg, Germany, December 2001.

[108] Stefan Roettger, Martin Kraus, and Thomas Ertl. Hardware-accelerated volume and isosurface rendering based on cell-projection. In *"Proc. IEEE Vis 2000"*, pages 109–116. IEEE Computer Society Press, 2000.

[109] B. Rogers and M. Graham. Similarities between motion parallax and stereopsis in human depth perception. *Vision Research*, 22:261–270, 1979.

[110] Detlef Runde. How to realize a Natural Image Production using Stereoscopic Display with Motion Parallax. *IEEE Trans. On Circuits and Systems for Video Technology*, 10(3):376–386, April 2000.

[111] Chris Schoenenman. Converting Frequency to RGB. World wide web, November 1991. `http://www.research.microsoft.com/~hollasch/cgindex/color/freq-rgb.html%`.

[112] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit.* Kitware, Inc. Publisher, 2^{nd} edition, 1998.

[113] S.Clarence, A. Ellis, and G. Rein. Groupware: Some Issues and Experiences. *Communications of the ACM*, 34(1):38–58, January 1991.

[114] H. Sedgwick. *Handbook of perception and human performance*, chapter Space Perception (Chapter 21). John Wiley & Sons, 1986.

[115] H.E. Seemann. *Physical and Photographic Principles of Medical Radiography.* Wiley, 1968.

[116] Steven Senger. Visualisation and segmenting large volumetric data sets. *IEEE Computer Graphics and Applications*, pages 32–37, May/June 1999.

[117] B.N Shand. Analysis of Vengrinovich's proposal: Using the maximum entropy method for 3D tomographic LODOX reconstruction. Technical report: Digital image processing, University of Cape Town, DIP Lab, Department of Electrical Engineering, University of Cape Town, Cape Town, Rondebosch, 1998.

[118] B.N Shand, P.L Starke, and G. de Jager. Limited Angle Tomography with LODOX. Technical report: Digital image processing, University of Cape Town, DIP Lab, Department of Electrical Engineering, University of Cape Town, Cape Town, Rondebosch, 1998.

[119] David Southard. Transformations for stereoscopic visual simulation. *Comput. & Graphics*, 16(4):401–410, 1992.

[120] P.L Starke and G. de Jager. Further work on limited angle tomography with LODOX. Technical Report: Digital Image Processing NA, University of Cape Town, DIP Lab, Department of Electrical Engineering, University of Cape Town, Cape Town, Rondebosch, 1998.

[121] H.M. Stauffer, G.C. Henny, and A.W. Blackstone. Stereoscopic televised flouroscopy. *Radiography*, 79:30–34, July 1962.

[122] Richard Steenblik. Cromadepth frequently asked questions. World wide web, 4 2002. `http://www.chromatek.com/FAQ/faq.shtml` — Still Available 30/04/2002.

[123] Eric Stollnitz, Tony Derose, and David Salesin. *Wavelets for Computer Graphics: Theory and Applications.* Morgan Kaufmann Publishers, San Francisco, California, 1996.

[124] R. Surdick, E. Davis, R. King, G. Corso, A. Shapiro, L. Hodges, and K. Elliot. Relevant Cues for the Visual Perception of Depth: Is it where you see it? In *Proc. HFES* 38^{th} *Annual Meeting*, pages 1305–1309, 1994.

[125] Troll tech. QT: The multi-platform C++ GUI/API. World wide web. `http://www.trolltech.com`.

[126] UK The Nottingham Trent University. 3D Imaging Group. World wide web, 2002. `http://www.eee.ntu.ac.uk/research/vision/projects.htm`.

[127] H.B. Tilton. Nineteen-inch parallactiscope. *in Proc SPIE*, 902:17–23, 1988.

[128] Takashi Totsuka and Marc Levoy. Frequency Domain Volume Rendering. *Computer Graphics*, 27(Annual conference series):271–278, 1993.

[129] K. Hohne U. Tiede, T. Schiemann. High quality rendering of attributed volume data. In Holly Rushmeier David Ebert, Hans Hagen, editor, *Proceedings of Visualization 1998*, pages 255–262, Los Alamitos, California, October 1998. IEEE Computer Society, IEEE Computer Society Press.

[130] N. Valyus. *Stereoscopy.* Focal Press, London, 1966.

[131] Allen van Gelder and Kwansik Kim. DIRECT VOLUME RENDERING WITH SHADING VIA THREE-DIMENSIONAL TEXTURES. Technical Report UCSC-CRL-96-16, Univesity of California, Santa Cruz, 1996.

[132] C.L. Vaughan, A. Nicol, P. Marais, and G. de Jager. Ami-lodox group meeting and discussions, April/May 2002.

[133] B. Vettermann, J. Hesser, and R. Manner. Solving the Hazard Problem for Algorithimcally Optimized Real-Time Volume Rendering. In *International Workshop on Volume Graphics*, March 1999.

[134] R.D. Walk and E.J. Gibson. A comparitive and analytical study of visual depth perception. *Psychological Monographs*, 75, 1961.

[135] C. Ware. Dynamic Stereo Displays. In *CHI '95 Conference Proceedings, Human Factors in Computing*, pages 310–316, May 1995. Available online: `http://www.acm.org/turing/sigs/sigchi/chi95/Electronic/documents/papers%/cw_bdy.htm`.

[136] Colin Ware. *Information Visualisation: Perception for Design.* Morgan Kaufmann Publishers, 2000.

[137] Allan Watt and Mark Watt. *Advanced Animation and Rendering Technique: Theory and Practice.* Addison-Wesley, Reading, Massachusetts, 1st edition edition, 1992.

[138] Jane Wilhelms and Allen Van Gelder. Topological considerations in isosurface generation, extended abstract. *Computer Graphics, Special Issue on San Diego Workshop on Volume Visualization*, 24(5):79–86, 1990.

[139] C.M. Witternbrink, T. Malzbender, and M.E. Gross. Opacity-Weighted Color Interpolation For Volume Sampling. In *Symposium on Volume Visualization*, pages 135–142, Research Trianlge Park, NC, October 1998.

[140] R. Yagel. Towards Real Time Volume Rendering. In *GRAPHICON '96*, pages 230–241, July 1996.

[141] Roni Yagel and Arie Kaufman. Template-based volume viewing. *Computer Graphics Forum, Proceedings Eurographs*, 11(3):153–167, 1992.

[142] M. Young, M.S. Landy, and L.T. Maloney. A perturbation analysis of depth perception from combinations of texture and motion cues. *Vision Research*, 33:2685–2696, 1993. `http://www.cns.nyu.edu/~msl/publ.html`.

[143] S. Zhang, P.H. King, and X. Pang. Recent progress in x-ray stereoscopes. *Medical Physics*, vol. 10(no. 5):574–578, September/October 1983.