Visual, Spatial and Temporal Quality in Video-Based Reconstruction of People: Achieving, Prototyping and Evaluating

Robert Andrew Aspin

University of Salford

School of Computing, Science and Engineering

2014

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Contents

1 Introduction 1

1.1 Subject of Research 3

1.2 Motivation and Setting 5

1.2.1 Related Research 9

1.2.2 The oCtAVE system 10

1.2.3 Issues and challenges with the oCtAVE system 13

1.3 Timeliness 15

1.4 Aim and Objectives 17

1.5 Scope 19

1.6 Methodology 22

1.7 Impact of Literature Review on Research 24

1.8 Chapter Summary 26

2 Background 30

2.1 Introduction 31

2.2 Case for 3D Reconstruction of a Person 32

2.3 Techniques for 3D reconstructions 35

2.4 Analysis of 3D reconstruction 40

2.5 Rational for research direction 42

2.6 Chapter Summary 44

3 An Initial Study into Voxel Based Evaluation and Rendering of 3D Reconstructed Human-like Forms 46

3.1 Introduction 48

3.2 Approach 49

3.2.1 Requirements 52

3.2.2 First Generation Simulator 53

3.2.3 Second Generation Simulator 58

3.3 Experimentation and Evaluation 61

3.3.1 First Generation Simulator 61

3.3.2 Second Generation Simulator 64

3.4 Discussion 71
3.5 Conclusion 73
3.6 Chapter Summary 74

4 An Approach for GPU Aligned Integrated Rendering and Reconstruction 75
4.1 Introduction 77
4.2 Requirements 80
4.3 Theoretical Description 81
4.4 Approach 85
4.5 Implementation 89
4.5.1 Sampling 94
4.5.2 Fragment Evaluation 101
4.6 Evaluation 103
4.6.1 Initial evaluation of performance 104
4.6.2 Evaluation of performance using real-world capture data compared with simulated capture data 118
4.7 Discussion 129
4.8 Conclusion 139
4.9 Chapter Summary 141

5 Validating the Approach 143
5.1 Introduction 144
5.2 Approach 147
5.2.1 Analysis 150
5.3 Case Study 160
5.3.1 Impact of Number of image sources (cameras) on reconstruction quality 164
5.3.2 Impact of Texture resolution on reconstruction/rendering performance 169
5.3.3 Impact of sampling interval on reconstruction/rendering quality 175
5.4 Discussion 183
5.4.1 Case Study 185
5.4.2 Wider impact of the objective analysis approach 187
5.5 Conclusion 188
5.6 Chapter Summary 189

6 Discussion and Conclusion 191
6.1 Summary of Work Done 191
6.2 Point of Departure 193

6.3 Review of Aim and Objectives 194
6.4 Review of Methodology 198
6.5 Contributions of Research to Literature 199
6.6 Conclusion 201
6.6.1 Short Comings of the Research 207
6.6.2 Future Research Directions 209

A First Generation: Code samples 230
A.1 Volume Rendering: GLSL Vertex Shader Code 230
A.2 Volume Rendering: GLSL Fragment Shader Code 232

B Third Generation: Code samples 240
B.1 Slice Plane Sampling GLSL Code 240
B.1.1 Slice Plane Sampling Vertex Shader 240
B.1.2 Slice Plane Sampling Fragment Shader 241
B.1.3 Slice Plane Sampling Fragment Shader with Occlusion Testing 247
B.2 Ray Cast Sampling GLSL Code 254
B.2.1 Ray Cast Sampling Vertex Shader 254
B.2.2 Ray Cast Sampling Fragment Shader 255

List of Figures

1.1 The oCtAVE system: Overview 11
1.2 The oCtAVE system: CAVE configuration 12
1.3 The oCtAVE system: Schematic 14
1.4 3D Video based reconstruction pipeline 20

3.1 First Generation Simulator: Proof of Concept 50
3.2 Second Generation Simulator: A configurable platform 50
3.3 Voxel Occupancy Model 54
3.4 Camera positioning 63
3.5 Performance Test: 2nd Generation, Occupancy 66
3.6 Performance Test: 2nd Generation, Image Size 68
3.7 Performance Test: 2nd Generation, Volume Size 69
3.8 Performance Test: 2nd Generation, Multiprocessing 70

4.1 Third Generation Simulator: GPU Shape and Texture 79
4.2 Model of Walking Person used for testing 88
4.3 Comparison of 3D reconstructions against source models . . . 90
4.4 Intersection of projected images forming reconstruction 93
4.5 Sampling planes used to intersect reconstruction volume . . . 96
4.6 Reconstruction formed by evaluating projected images with sampling planes 102
4.7 Performance Test: 3rd Generation, Number of source cameras 106
4.8 Performance Test: 3rd Generation, Image Size 107
4.9 Performance Test: 3rd Generation, Occupancy 109
4.10 Performance Test: New Approach, Number of Source Cameras impact on reconstruction and rendering times 112
4.11 Performance Test: New Approach, Number of Source Cameras impact on application framerate 112
4.12 Performance Test: New Approach, Source image size impact on reconstruction and rendering times 115
4.13 Performance Test: New Approach, Source image size impact on application framerate 115

4.14 Performance Test: New Approach, Occupancy impact on reconstruction and rendering times 117
4.15 Performance Test: New Approach, Occupancy impact on application framerate 117
4.16 Datasets used for testing: Sample Reconstructions 120
4.17 Performance Test: New Approach, Sampling density impact on performance 121
4.18 Performance Test: New Approach, Number fo source's impact on performance 125
4.19 Performance Test: New Approach, Impact if differing graphics hardware on performance 127
4.20 Artefacts (shadows) created by reconstruction process 131
4.21 3D reconstruction without and without occlusion testing . . . 137

5.1 Spot the Difference: Effects of sampling variation on reconstruction 144
5.2 Simulated capture tool configurations 147
5.3 Evaluation images taken from virtual surround camera 149
5.4 Outline of applications and data flows 153
5.5 Image pair undergoing evaluation of quality 154
5.6 Mercator projected plot of analysis of form errors 156
5.7 Mercator projected plot of analysis of colour (greyscale) discrepancy 158
5.8 Two different variations of volume space sampling 161
5.9 Analysis of colour (greyscale) for differing numbers of reconstruction source camera 166
5.10 Central (view) band analysis of colour (greyscale) for differing numbers of reconstruction source cameras 167
5.11 Analysis of form for differing numbers of reconstruction source cameras 169
5.12 Central (view) band analysis of form for differing numbers of reconstruction source cameras 170
5.13 Analysis of colour (greyscale) for differing source image resolutions 173
5.14 Central (view) band analysis of colour (greyscale) for differing source image resolutions 174
5.15 Analysis of form for differing source image resolutions 175
5.16 Central (view) band analysis of form for differing source image resolutions 176
5.17 Analysis of colour (greyscale) for differing sample intervals . . 179

5.18 Central (view) band analysis of colour (greyscale) for differing sample intervals 180
5.19 Analysis of colour (greyscale) for differing sample intervals . . 181
5.20 Central (view) band analysis of colour (greyscale) for differing sample intervals 182

Abstract

Capturing, recreating and representing a high fidelity virtual representation of the dynamic human form has long been a target for a diverse range of applications including tele-presence, games, film and TV special effects. The complexity of the challenge, to achieve a lifelike, faithful and believable representation, is such that a wide range of techniques and approaches have been developed. These are both due to research lead curiosity and requirements to address specific objective for particular problems.

This work starts from a novel standpoint: that the processes of surfacing, tessellation and texturing, commonly used in 3D reconstruction, are computationally expensive and un-necessary. This work argues that by integrating the reconstruction and rendering processes into a single process that is aligned with the architecture of modern graphics hardware, a lightweight component solution can be achieved that is suitable for application on the end user systems within the many application domains.

In order to achieve this aim the research undertaken seeks to both define an appropriate technique and develop detailed understanding of the reconstruction process pipeline and impacting factors. This is achieved through a complementary investigation of the tools and frameworks that are necessary to support iterative development of the approach with reliable, repeatable objective assessment. This reasons that by understanding the nature of the capture, reconstruction and presentation pipeline and by objective evaluation of the emerging reconstruction techniques this research will define an approach for 3D video based reconstruction that effectively utilises the processing potential of a single system to deliver acceptable levels of performance (speed) and fidelity (visual quality) for a componentised, multi-purpose 3D reconstruction and rendering solution.

This thesis describes the research that has driven the evolution of technique and documents the iterations made. It presents a novel framework for experimentation and evaluation of the techniques and demonstrates how the use of these tools has enabled both rapid prototyping of approach and objective evaluation of improvement. The work concludes with a review of the approach taken and identifies approaches for evaluation of performance (speed) and fidelity (visual quality) that enable both repeatable experimentation within the research pipeline and reliable comparison of the end-to-end process against other techniques.

Chapter 1

Introduction

This thesis presents a novel simulation framework for objective exploration of video based 3D reconstruction approaches. This offers repeatable experimentation and reliable objective comparison as part of an iterative research process. The framework is used to explore a case study which seeks to map the reconstruction/rendering processes to the features and functionality of the GPU system and define frameworks and tool sets for the evaluation and objective evolution of these techniques. This both demonstrates the approach taken, and establishes metrics for controlled objective evaluation against alternate approaches reported by other researchers. The case study of an evolving technique is used to question the need for model forming processes (surface fitting, tessellation, decimation, and texturing), in real-time generation of animated 3D forms from video based data. The standpoint is that these processes are slow and un-necessary. This argues that by moving to a more direct, combined reconstruction and rendering process, which utilises multiple video streams directly, a similar visual quality of reconstruction can be *achieved entirely within the programmable graphics hardware pipeline. This offers future scalability within a single system architecture.*

3D video based reconstruction, of the human form, is widely seen as advantageous for application in tele-present communication and collaborative virtual environment systems. The approach combines the advantageous features of traditional collaborative virtual environment user tracking (body posture and position) with the fidelity of video conferencing (appearance) to present the observer with a human representation that is spatially integrated and of sufficient quality to portray visual emotion. In essence, the observer can both see the virtual participant and see what they are looking at, whilst also engaging with none verbal communication queues to enhance communication

The original contribution to knowledge of this work is, therefore, the investigation of lightweight approaches to 3D reconstruction aligned to consumer type hardware platforms. Supporting this will be investigation into simulation frameworks to support development of understanding of the conditions for video based 3D reconstruction as part of a toolkit for continual objective analysis of the evolving process for reconstruction and rendering.

This introduction establishes the rational for the research described in this thesis. Fundamentally this chapter will seek to inform the reader of the rational for the research and present an overview of the work. This is presented by answering the following set of key questions:

- *What is the subject of this research?*
- *What is the motivation for the research?*

- *Why is it timely to consider this now?*
- *Who potentially benefits from the research presented here?*
- *How will the research be conducted?*
- *How does this work relate to the research group?*

1.1 Subject of Research

The general subject of this research is 3D video based reconstruction and is targeted at the rendering and reconstruction of a person captured by multiple, synchronised video streams. This is a wide domain, with multiple active researchers and a wide variety of techniques and approaches [44, 32, 24, 56]. However, the general principles common to the majority of these, are that a surface is formed, or fitted, around the intersection of segmented video images denoting the subject of reconstruction. This reconstructed surface is then, typically, tessellated, optimized and textured, before being passed to the rendering pipeline. The objective is to present a lifelike, faithful and believable dynamic representation of a person with sufficient fidelity to convey the nuances of gesture, expression and posture unique to each individual. In achieving this, these reconstructions seek to enhance communication by engaging the emotional, non-verbal effects that initiate and steer conversation fragments, communicate empathy and establish trust. The 3D representations of people captured, reconstructed and presented using such methods have use in tele-present and collaborative systems, both immersive and non-immersive, in which the faithful representation of a live person offers significant improvements over traditional tracked avatar models that cannot recreate the fidelity and detail of human expression [71].

This work concentrates on a specific objective; the reconstruction and rendering of a dynamic entity from multiple segmented and synchronised video streams. This is similar to the aims of others, however the work takes a novel standpoint; that the formation of 3D models from the video stream data is computationally expensive and un-necessary. By understanding the nature of the capture, reconstruction and presentation pipeline and by objective evaluation of the emerging approach against truth data this research will define an approach for 3D video based reconstruction that effectively utilises the processing potential of a single system to deliver acceptable levels of performance (speed) and fidelity (visual quality) for a componentised, multi-purpose 3D reconstruction and rendering solution.

The stepping off point for this research is based on previous investigations into 3D medical imaging which demonstrated techniques for real-time 3D volumetric visualisation [5, 4]. These applied an alignment of the data structure and operations with the architecture and functionality of consumer graphics hardware to deliver both performance gains and interactive functionality. Applying this approach to the problem of 3D reconstruction will enable the goal of the research to be met by removing the need for polygon forming through the direct rendering of the volumetric data set formed by the initial process in space carved reconstruction approaches.

To support the investigation into reconstruction processes understanding of the reconstruction pipeline, and its impact on the reconstruction process is required. It was, therefore necessary to define and create a framework for simulation, prototyping of data and process and objective evaluation of technique. This both supports the development process with dependable, repeatable analysis and allows objective measures between variants of technique to be assessed as evolutions in approach are made. This enabled rapid reconfiguration of a virtual capture environment which allowed refined synthetic data sets to be produced whilst also offering the ability to undertake accurate performance evaluation and highly detailed holistic objective analysis of the resultant reconstructions against truth data gained from the virtualised capture source.

The combination of both novel technique and a novel framework for investigation, prototyping and comparative objective evaluation form the substantial contributions of this work.

1.2 Motivation and Setting

Video Conferencing and Collaborative Virtual Environments (CVEs) both offer the facility to display a tele-present representation of a remotely located participant within a communication platform. However, both approaches present issues which hinder natural conversation. Video Conferencing enables the participants to see what each other looks like and recognise facial gesture and expression which form none verbal queues to support natural conversation. However, the lack of spatial alignment can prevent these from being accurately focused and identified. Put simply, when the person on the screen appears to be making eye contact, they are looking directly into the camera, not the the observer's image on their own screen [69]. CVEs use tracking mechanisms, commonly linked to an artificial representation, an avatar, which is presented within a common spatial reference mediated by the shared virtual environment. In effect the participants are sharing the same space, with the tracked avatars displaying the gross movements and virtual position of the participants. However, the fidelity of the avatar cannot capture the detail of video, meaning that expression is lost. This is seen in the increased use of corrective discussion which fragments conversation and inhibits natural communication [33]. While Video Conferencing supports temporal interaction supporting conversation, it is poor at facilitating spatial references for effective communication. Conversely, CVEs strongly support spatial referencing, but lack the fine, none verbal queues that support conversational flow. Both spatial and temporal elements are seen as part of the key requirements for effective collaboration [78], supporting social interaction which is critical to seamless collaboration [31]. Provided this integration can be achieved with sufficient real-time performance and visual fidelity, it is hoped that significant reductions in the ambiguity and fragmentation in communication experienced within other tele-present and mixed media collaborative systems can be achieved [33].

Replicating a dynamic human in a virtual form offers advantages to many domains, particularly those concerned with virtual communication such as distributed collaborative systems and emotional engagement. These include games, film and TV, Video Conferencing (tele-presence) and Collaborative Virtual Environments. Currently film/video based media, including video conferencing, can convey with great fidelity subtle nuances such as non-verbal expression, eye gaze and skin tone that are important for human communication. However, the fixed viewpoint these media present imagery does not consider the spatial relationship between the virtual and real world participants as important. As a result these paradigms struggle to support the communication cues of individual trust, engagement and empathy relationships that need to cross the boundary of the spatial domains. Research shows that immersive collaborative virtual environments can combine the features of these two domains to address the weaknesses of each in supporting enhanced communication at a distance [71].

Video based 3D reconstruction seeks to address the recreation and representation of a dynamic human within a medium in which temporal and spatial factors can be aligned to deliver effective distributed communication, collaboration, empathy and engagement. Many different approaches are reported for the general problem of 3D reconstruction. However, the specific problems of fidelity and performance necessary to support communication techniques based on shape from silhouette [7], for the formation of a visual hull [44], or image based rendering approaches [89] are common. The first two of these model forming processes, aim to solve the holistic problem, enabling free-viewpoint observation, but impose higher computational load requirements with the need for complex and extensive computing resources to achieve real-time rates. Image based rendering (IBR) approaches offer a relatively lightweight approach with little computational load, but are constrained in that the representation is only accurate for a single specific point of observation.

This research seeks an approach to 3D reconstruction that combines the advantages of both techniques; the computational load of image based rendering approach coupled with the holistic integrity of the model forming processes.

Graphics inspired processes work best when each element can be processed independently of all others as part of a massively parallel pipeline. As an example consider the difference between local and global illumination models commonly used in graphics rendering. In a local model only the drawn pixel (fragment) is considered against the configuration of lights and viewpoint. This approach is inherently very fast and ideally suited to the underlying hardware. In a global lighting model the propagation of a ray of light, from source to eye, including all the events of refraction, reflection, diffraction, etc. that this ray encounters against all of the objects in the scene must be evaluated. This approach is inherently accurate, but does not translate well to graphics hardware and is traditionally implemented as a non-real-time system, such as ray-tracing or radiosity, enacted on a more general purpose processor; the CPU. The core argument of this research is that current processes are akin to global illumination methods. Precise, but slow; some acceleration can be gained by optimization and most recently migrating appropriate sub-processes to graphics co-processing hardware (GPU), however, there will always be bottlenecks. This research is seeking the equivalent of the local illumination model; a technique that is ideally suited to the underlying hardware, which can leverage this infrastructure effectively and due to its inherit structure will scale uniformly with future evolutions of that hardware.

This research is therefore looking for an approach that delivers acceptable levels of performance and fidelity, but is also attuned to the types of computing resources end-users are likely to have rather than the facilities and resources research centres might utilise.

1.2.1 Related Research

This research has been undertaken within a research centre with a wider agenda in this area. During the time this research was undertaken related research was also undertaken by others.

Duckworth [19] undertook related research. This followed a similar aim, in terms of reconstructing people, but investigated the use of heterogeneous processor clusters to gain performance through parallel processing in the solution of more traditional surface based reconstruction techniques.

Moore [61] investigated the process of capturing and segmenting image data from multiple synchronised video streams with the aim of delivering low latency image data, from an integrated capture and immersive display system, in a manner than could support the related research of both this work and that of Duckworth.

During the course of this research, collaboration between the independent researchers was supported by their common supervisor to define and enact experimentation evaluating the impact of the generated reconstructions. These experiments enacted subjective evaluation of the reconstructed forms, through the matching of the participants eye gaze to that of the reconstructed form's, to establish the quality of representation of non verbal communication queues within the reconstructions generated [72].

1.2.2 The oCtAVE system

While this research is seeking a lightweight, componentisable 3D reconstruction and rendering solution, it is supported by a unique research facility, the oCtAVE, that offers both data capture, in the form of synchronised video streams, and large scale immersive visualisation [64]. At the time of starting the research this facility was under construction. This was one influencing factor in the use of simulation frameworks for initial investigations and the use of these also enabled prototyping of issues, such as camera placement, that were integrated into the facility as construction progressed. As the oCtAVE developed successive iterations of the simulation framework were constructed which accurately reflected the configuration and parameters of the oCtAVE system to ensure simulation environments and the physical system had matching characteristics.

The oCtAVE system is a mixed reality/multi modal system that offers both visualisation and capture functionality. The display system comprises 14 projected surfaces which can be re-configured into multiple display formats enabling it to replicate a variety of conventional systems such as power-walls and CAVEs [14, 15]. 8 surfaces are constructed as moveable, back projected rigs vertical screens forming the walls of the system, with the remaining 6 surfaces being formed by fixed roof mounted projectors forming the floor surfaces. For the majority of time this is configured as an 8 sided, 5.5m diameter octagonal system with a projected floor (Figure: 1.1).

The display system can operate in a similar manner to traditional CAVE systems, using a Vicom tracking system to locate the user's head position to

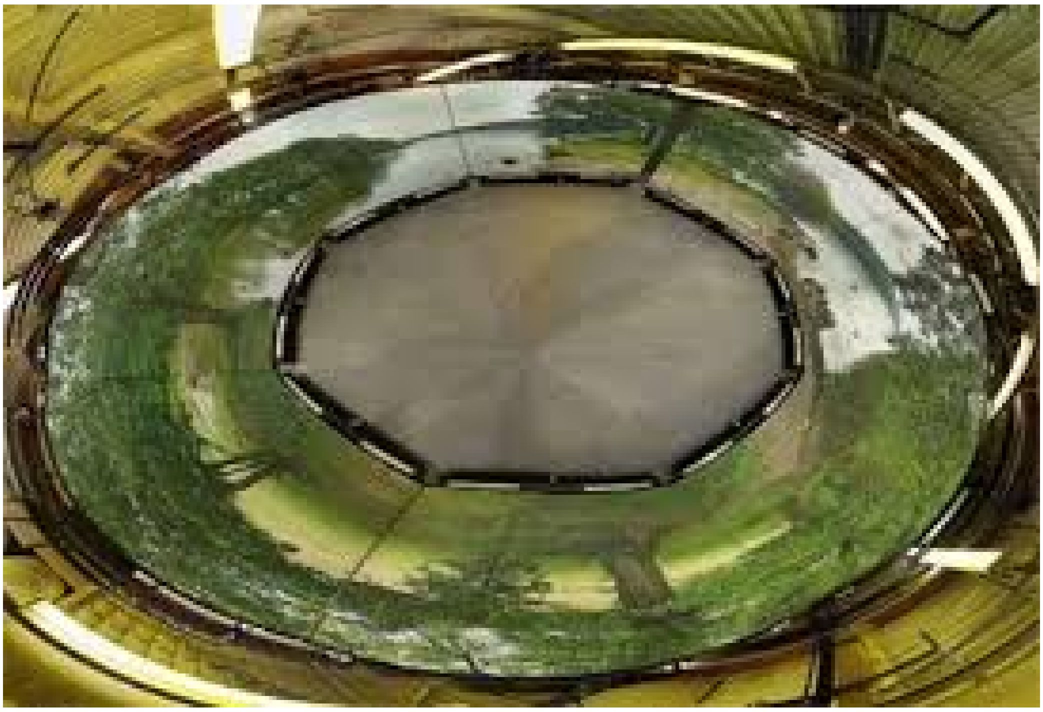


Figure 1.1: The oCtAVE system: View from above in normal configuration without floor in place (Source: University of Salford)

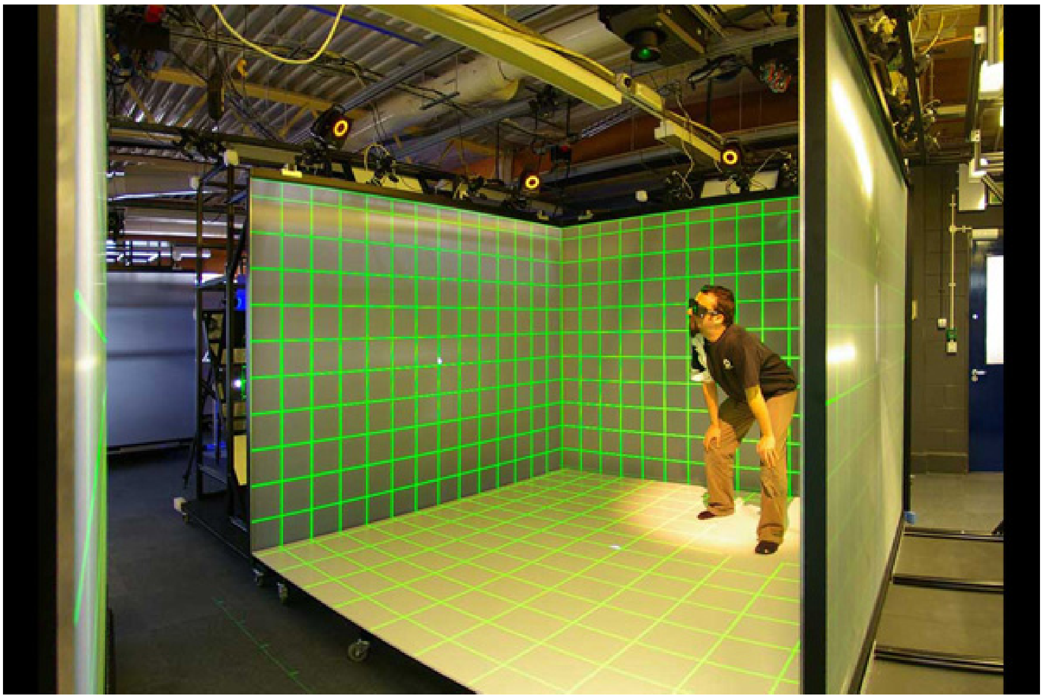


Figure 1.2: The oCtAVE system: CAVE configuration, stereo display (Source: University of Salford)

provide head tracked stereo display to all screens for an immersive experience 1.2, or each screen can be used independently as a single large screen display.

Tracking of the user within the system is enacted using a Vicon MX-F40 system. This is supported by an array of Basler ethernet cameras which provide video capture. Cameras are mounted to the infrastructure supporting the display screens and can be moved/relocated as needed.

Rendering (display) is provided by a cluster of Sun Ultra40 systems, equipped with nVidia FX-5600 graphics cards, each attached to a single Christie S3K projector to form a single rendering pipeline. Additional computation is provided by a SunFire x4600 system. All computation resources are connected by a dedicated local network (Cisco Catalyst 4900M @10Gb/s) 1.3.

Audio facilities within the system were not utilised in this research, however these are provided by an experimental waveform synthesis system rendered through an array of 256 speakers to provided spatially localised sound sources [65].

1.2.3 Issues and challenges with the oCtAVE system

The oCtAVE system is an experimental research platform supporting a wide range of research activities. As such, it is constantly evolving and changing and has an element of instability associated with any research platform. It is also a significantly larger system than this research was targeting for the platform on which the rendering/reconstruction approach would operate on. However, each rendering pipeline within the system is equivalent to a

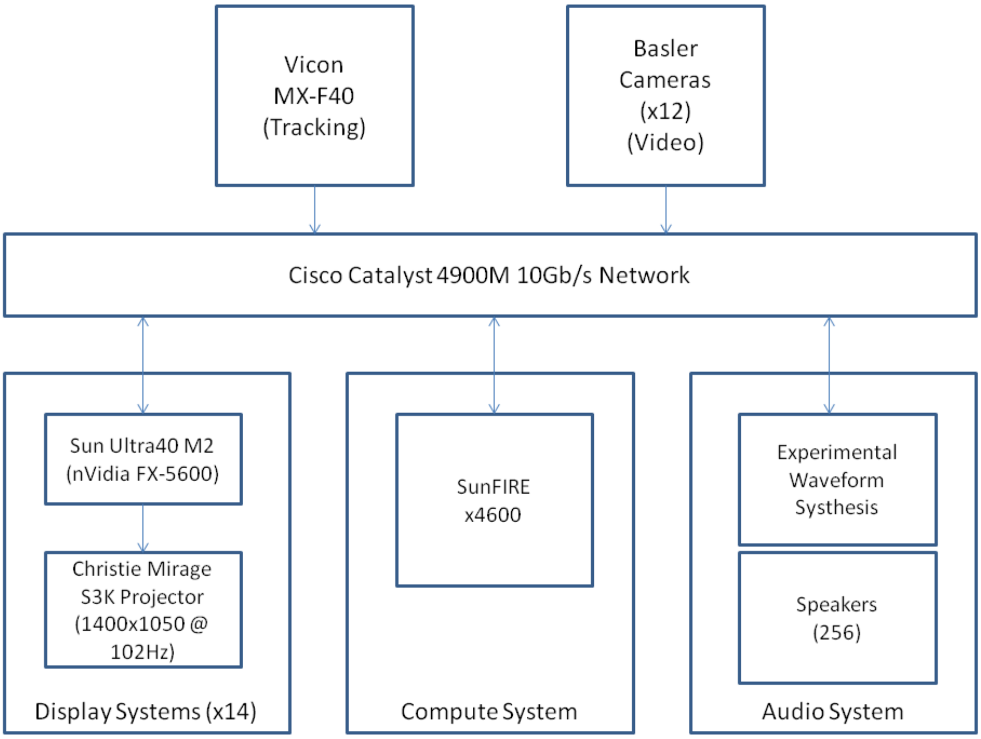


Figure 1.3: The oCtAVE system: Schematic outline of computational resources

typical high end consumer system with a large screen display, rather than a monitor, attached. These systems were used as one of the test platforms for performance testing and were similar in performance to the laptop systems used to develop the techniques and the simulator. The screen size of the system approx 2.5m x 2.2m was also used as the basis for calculation of occupancy, used in performance evaluation, where the proportion of the screen occupied by a 1:1 scale reconstruction of a person was considered as one of the independent factors influencing reconstruction/rendering speed.

The primary use of the oCtave system was for capture of data sets for experimental reconstruction (referred to as "Salford" datasets). These were captured using the Basler camera system to record multiple video streams of an actor within the system. These streams were synchronised, segmented and prepared as part of a fellow student's research activity [61]. The image sets and segmentation data gained though this research were used as test data within this research and other collaborative research within the wider group [19, 72]. In this camera placement for the capture system was informed by the initial studies performed within this research (Chapter: 3).

1.3 Timeliness

Achieving believable and faithful 3D reconstruction reconstructions aims to cross the boundary presented by the Uncanny Valley [75] and engender virtualised human forms with the visual features necessary to effectively communicate trust, emotion, and conversational interaction across the virtual/real divide. In recent years computational processing power and the availability of high speed, high resolution cameras have enabled researchers to make significant progress towards this goal [29, 55, 58, 44, 24].

While the performance increase of traditional CPU based computation shows signs of slowing, the processing power available from graphics hardware, through the reprogrammable GPU pipeline is increasing at an accelerating rate. This is primarily achieved through the exploitation of massive parallelism in both the processing units and the memory architecture that feeds them [52].

The view, presented by this thesis, is that to achieve such reconstructions in an accessible and usable format, that can be delivered to end-user systems, approaches for the reconstruction and rendering process need to be aligned to these rapidly evolving GPU architecture, rather than though the utilisation of computing clusters or CPU resources. This requires a re-consideration of the algorithmic approaches taken. Rather than seeking localised gains within sequential stages of the process, where many sub-processes need to be co-ordinated and controlled, creating bottlenecks, an approach in which each independent fragment of the reconstruction can be evaluated independently is required. This should both maximise the use of the parallel computation hardware and offer future scalability with the relentless increase in GPU units available within a single hardware device.

This research is therefore seeking a solution in which each screen pixel, or rather the fragment processing pipeline controlling it, can be considered an independent processing unit for the calculation of the 3D reconstructed form. This, therefore, seeks an approach that matches the current state of the art in terms of performance and visual fidelity, but offers greater potential for increases, on current and emerging hardware, both as a scalable system directly dependent on the number and performance of GPU units available within the rendering hardware. The assumption is that by combining the reconstruction and rendering process, and enacting both within the GPU system as a process that aligns to the graphics card architecture, current state of the art performance can be achieved within the limitations imposed by a single consumer type system. This further assumes that as future graphics card performance increases, the performance of the reconstruction technique will similarly improve to a point where it is viable for the majority of end user systems.

1.4 Aim and Objectives

Aim: To improve upon the temporal and visual quality of 3D video based reconstruction through new approaches to developing, doing and measuring it.

This work will study and understand the input characteristics, process pipeline and quantifiable objective measures of 3D video based reconstruction that are necessary to deliver a framework for targeted reconstruction/rendering technique development.

Specifically the objectives for this research are:

- **O1:** *Develop novel approaches to holistically measure the visual, spatial and temporal qualities of video based reconstruction while varying key impacting factors. This will establish developmental frameworks* *and objective measurement approaches that support development of 3D video based reconstruction processes towards specific goals of performance (speed) and fidelity (visual quality).*

- **O2:** *Develop a novel reconstruction/rendering technique that improves the visual, spatial and temporal qualities of video based reconstruction. This will determine algorithms and approaches and techniques that can be successfully utilised in the creation of an efficient and effective reconstruction/rendering technique*

- **O3:** *Assess the novel technique for improving the balance of visual and spatial qualities by applying a novel approach to objectively measure its holistic performance. This seeks to determine through repeatable objective analysis that the novel technique for 3D reconstruction aligns to the standards set by current state of the art.*

These can be articulated into a single research question: Can 3D video based reconstruction and rendering be combined and reduced to remove computationally expensive processes and deliver a GPU aligned approach that matches current state of the art performance and quality metrics within a componentisable form suitable for future consumer type systems.

These objectives and their relationship to the overall aim of the work can bee seen as a set of overlapping domains which each contribute to the answering of the research question. The inherent interdependency of the objectives suggests an approach that iteratively refines understanding for each objective as the work refines it's solution and validates the approaches taken.

1.5 Scope

This work is primarily concerned with the reconstruction of the dynamic objects from multiple, synchronised video streams. Within this the work concentrates specifically on the reconstruction and rendering of the human form, from the point at which segmented video images are delivered to the host system through to the presentation of a rendered form within a free-view point application. Capture, synchronisation and segmentation of the video streams is a significant challenge outside of this research 1.4 and is actively pursued by others [61], and while the findings of this study have informed related works [62] this is not a fundamental goal of the research. In recent years the use of depth based imaging systems has also become increasingly relevant [97, 98], however these devices are in their infancy with resolutions and speeds that cannot currently offer the levels of fidelity required across an entire human form [88].

Within the technological platform the work seeks a solution to the reconstruction/rendering process that does not require external computational resources. Therefore the platform boundaries for the approach are that it should deliver levels of performance and fidelity similar to established approaches within the confines of a single computer system and the graphical processing units contained by it. This will be assessed against *a priori* published findings, which report achieved performance in the range of 12-15 frames per second (fps) [102, 13, 90] and present images of the reconstruction used in subjective evaluation [58, 57].

Development of the technique for reconstruction and rendering is antic-

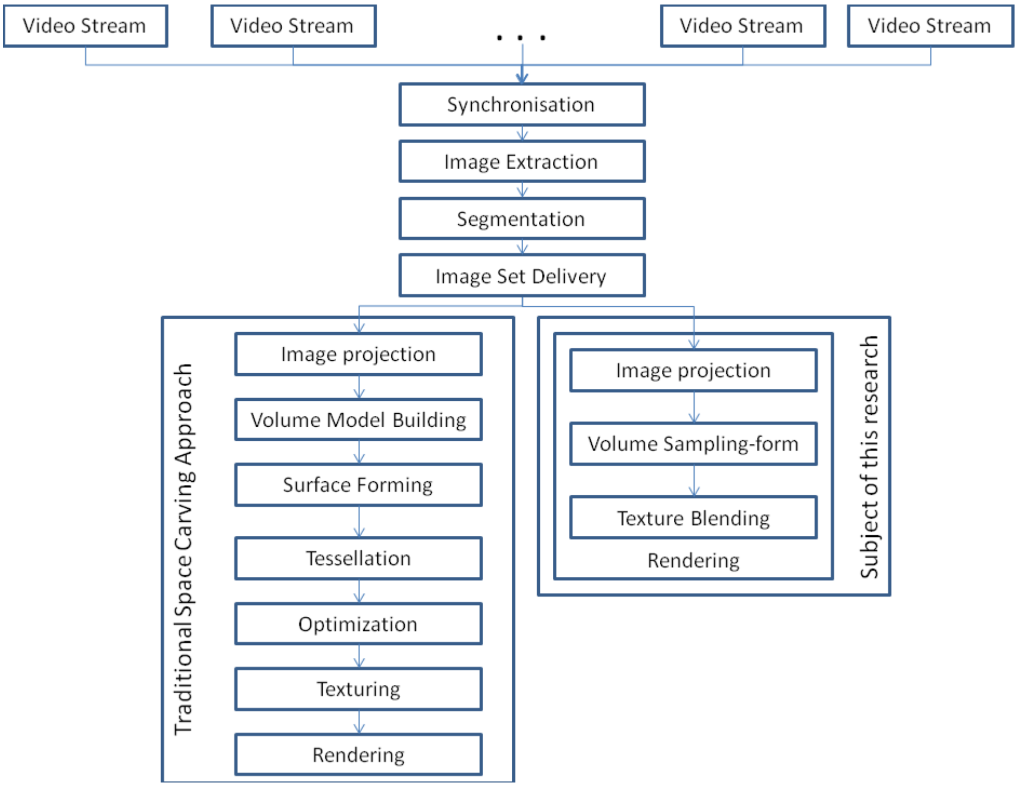


Figure 1.4: 3D Video based reconstruction pipeline, showing both stages for traditional approaches and the approach taken by this research and highlighting the scope of the research undertaken

ipated to be a process of iterative refinement. Fundamental to this is the establishment of a robust platform for objective experimentation. This will allow evaluation of, and how, the evolving techniques used for reconstruction/rendering approach the goals of the research. To this end it is necessary to consider the capture of source data and additional measurement of the reconstruction subject for objective analysis in a dependable and repeatable experimental context.

Physical configuration, and re-configuration of a camera network comprising the capture environment is a time consuming occupation, dependent on research outside the scope of this work, and open to sources of error (for example: signal noise, calibration timing, segmentation inconsistencies.), that reduce the objectivity of the quantifiable analysis. Holistic, objective analysis of fidelity of reconstruction, requires the capture and evaluation of additional datasets which are physically impracticable due to the limitations of cameras placement and dynamism of the source object. Understanding the nature of the input data configuration, and its impact on the resultant form is essential to achieve focused research towards the objectives set. Therefore the scope of the research is defined to include tools, and techniques that aid understanding of the capture data and its impact on the resultant form with the intention of formulating a framework for exploration of the capture system, understanding of the reconstruction/rendering pipeline, and objective analysis, against source data, as part of a iterative development process which enables objective evaluation of variants of technique against both alternate iterations and reported data on external research. This aims to deliver a platform for focused reconstruction technique development based on quantifiable objective measures which assess the accuracy of the reconstructed form to that of the source object and evaluate the matching of the surface texture to that of the source object's appearance.

1.6 Methodology

The research methods used reflect common practices within the computer science domain. The work is predominately formative, being concerned with the definition of process/method/algorithm, concept, and framework. Research methods adopted reflect common approaches, in this domain, of literature review, concept implementation, simulation and case study [70]. This reflects a design and modelling approach common to a high proportion of computer science literature [96].

Common practice, within the 3D reconstruction community, is that implemented algorithms are tested within a simulated environment and reported in terms of raw performance and visual evaluation of imagery generated. Within the VR/Tele-presence community it is common for these algorithms to be incorporated into prototype systems for subjective evaluation of the resultant forms impact on human factors such as conversational flow, emotive engagement and trust. Whilst this work is more closely aligned with the former, the fine grained iterative nature of technique refinement would, it was felt, make subjective evaluation of fidelity, between two approaches with small variation in operation, unreliable.

The process for research was devised to enable rapid prototyping through objective evaluation, whilst minimising the impact of sources of error from outside the scope of the research and reducing the dependence on physical infrastructure and resources that would impose time and logistical constraints. In essence simulation frameworks supporting conceptual implementations that are reported as case studies in the 3 key chapters of the thesis (Chapters: 3-5).

Iterative evolution of technique and simulation toolkit, within a framework of objective analysis, though research informed enhancements is used to enable refinement towards the research aim. Refinement of approach and step changes made, informed by continual literature review in which features of published approaches are considered against the parameters of the technique being evolved and incorporated into new variants for objective evaluation against the previous benchmark. Repeatable quantifiable objective analysis is, therefore, seen as an important factor in the research process. Fundamental to this is the establishment of tools and mechanisms for gathering reconstruction data sets that are capable of being related to a truth data set. This allows the quality of the reconstructed form to be objectively compared to the original source form. Obviously using video acquisition of a real person simply provides the data set for reconstruction without any ability to capture a truth based form. Using a simulated virtual capture space allows this reconstruction data set to be obtained from a 3D geometric model which itself forms the definitive base for comparison. Research into methods of comparative evolution of the 3D reconstruction against the base model are then used deliver a stable and consistent measure of quality that can be used to evaluate the fidelity of the different variants of reconstruction approaches, thereby informing selection of approach for further iteration. Each evolutionary cycle is inspired by established theory and adapted into the principles of the approach. Assessment of performance and quality for successive evolutions are required at each stage to ensure that the impact of the evolution is fully understood and to map the progress made both in terms of performance and quality. This mapping of alternate approaches enables a clear investigation of improvement with the ability to discard approaches that, while theoretically sound, do not assist in achieving the overall goal of the research.

This mapping of evolution in approach is essential to determine the matching of approach to goal. However, the ultimate aim of this research is to define an approach that is comparable to current techniques, while offering the potential for future enhancement. In order to achieve this the evaluation of the emerging technique must be grounded in comparison to the established approaches. In order to achieve this the final solution needs to be evaluated to determine both how it compares to other approaches and how it matches the metrics required for tele-presence in collaborative communication systems. Ultimately this will enable the profiling of the final approach, delivering metrics determining appropriate application and configuration as a mapping to problem domain.

1.7 Impact of Literature Review on Research

The initial prototype was constructed using tacit knowledge of the problem area as a test-bed to understand the problem domain. Prior understanding of 3D medical imaging of volumetric datasets [5, 4, 8] was combined with a basic approach to space carving [44, 42] to formulate a 3D reconstruction and rendering technique that fulfilled the general principles of free-viewpoint rendering [28] without the need for tessellation and surface forming which were common to this approach [30, 6, 27].

Following the initial study a more informed standpoint was taken in which the problems and issues arising from the initial prototype were investigated. In particular the issue of improving the believability and faithfulness of the reconstruction was highlighted as a key target [63, 71, 80] to move towards achieving the levels of fidelity necessary to replicate face-to-face meetings [77]. Fundamentally this was a concern to move from a simple reconstructed form, to a reconstruction in which the appearance of the subject was mapped to the shape generated through reconstruction [100, 30]. Given the nature of the reconstruction, which had no tessellated surfaces onto which texture could be mapped [66], approaches for projecting textures and evaluating them at the surface intersection were sought [23]. This lead to a significant change in the reconstruction approach. It was realised that while texture from the capture cameras could be projected onto the volume surface, this was essentially the same technique that was being used as a pre-process to for the space carved volume. With this realisation the reconstruction technique under development was re-created to integrate both the space-carving and surface texturing within a single integrated process that operated within the GPU resources of the graphics hardware.

In refining the approach to 3D reconstruction emphasis was placed on achieving reconstruction performance (speed) comparable to state of the art approaches. In this the performance techniques across the 3D video based reconstruction domain were considered [1, 24, 26, 30, 16]. From these estimates of state of the art performance could be drawn [102, 13, 90] which indicated that within the system specification being used a target of 12-15 fps was required. However in all these cases this was defined only as the reconstruction time, not an end-to-end cycle time. Further refinements to the reconstruction algorithm were also made by applying the principles of 'floating textures' for weighting the texture blending operations within the surface projections [20].

Validating the approach sought to introduce objective measures for assessing the fidelity of the reconstruction. Specifically this sought to complement traditional subjective evaluation processes [87, 56, 102] by introducing direct comparison of the reconstruction against the 'truth' subject. In achieving this the work aimed to improve the development process by enabling objective, evidence based decision making during the iterative refinement cycle. While approaches exist for assessing the video quality of the reconstruction, ie inter-frame analysis, [22, 18], the only reference to objective analysis against source data [39] was limited to utilising 'live' capture information and could not holistically assess the overall fidelity of the reconstruction.

1.8 Chapter Summary

This is formative research which is seeking an alignment of reconstruction/rendering processes, for 3D video based reconstruction of a person, with the architectures of modern GPU based graphics systems. To structure the research a scope has been defined with a single aim. This is further decomposed into objectives which address a research question which has been defined to provide coverage of the research scope and define a programme of work by which the aim will be addressed.

The work seeks to develop greater understanding of the process and influencing factors surrounding the general problem of 3D video based reconstruction of a dynamic human. The starting position is that current approaches utilise un-necessary and computationally expensive stages in the formation of their reconstructions. In particular these relate to the surfacing, tessellation and texturing of the generated reconstructed form. This research argues that these stages can be removed by defining an integrated reconstruction and rendering approach that is aligned to GPU based systems. This should deliver acceptable quality and speed performance within a component solution that matches typical, all be it currently high end, end user systems. To enhance the research process a simulation framework will be used to investigate influencing factors, improve developmental research prototypes and enable objective analysis of the evolving solution.

The thesis is organised into the following chapters which establish the context for the research (Chapter 2), present the core contributions (Chapters 3-5), and review the outcomes of the research against the aim, objectives and research question (Chapter 6)

- Chapter 2: Background, undertakes a literature review of the state of the art in the related fields of research.

- Chapter 3: An Initial Study into Voxel Based Evaluation and Rendering of 3D Reconstructed Human-Like Forms, reviews the stepping off point for the research and early evolutions of technique and simulation in which previous understanding of medically inspired volumetric visualisation techniques are explored as a basis for 3D reconstruction and rendering without the need for model forming.

- Chapter 4: An approach for GPU aligned integrated rendering and reconstruction, presents the inspirational jump between the 'massively brutal' and naive voxel solution toward a more GPU aligned approach which utilises features of the rendering pipeline to deliver a working, integrated reconstruction/rendering prototype. This also covers a substantial change in the nature of the simulation framework towards a platform for repeatable experimentation which supports the development process and establishes methods of objective evaluation of performance (speed) and fidelity(visual quality)
- Chapter 5: Validating the Approach, undertakes objective evaluation of the final reconstruction/rendering approach. In this quantitative analysis of the holistic reconstruction form is undertaken to establish benchmarks for visual quality and accurate measurement of performance (speed) are made to assess the matching against other approaches and characterise the scalability of the approach with evolving GPU systems.
- Chapter 6: Discussion and Conclusion, reflects on the methods developed and approaches taken and contextualises the work against the identified state of the art and more recent developments in the field. This cumulates in the conclusions and identifies directions for future

research.

Chapter 2

Background

This chapter seeks to establish the context of the research. This is presented through literature review, and provides a description of the current state of the art for both the domains of real-time 3D reconstruction of people and a wider review of tele-presence and collaborative virtual environments as a mechanism for supporting effective communication.

These reviews establish the state of the art of current approaches that are addressed by the research presented here and define the metrics for evaluation of the work. In essence this will review the foundational principles for the research undertaken; establishing the theoretical underpinnings of each and exploring the scientific assumptions made.

This chapter concludes with a rational for the research direction which provides contextual background to the approach taken.

2.1 Introduction

The starting point for the research is to utilise previous understanding into the display of volume based data [5, 4] to establish a basic prototype for developing understanding of the factors and issues that impact upon reconstruction fidelity and performance. This literature review is used to establish the state of the art, in terms of reconstruction approaches, common features of the capture/reconstruction pipeline and standards of performance and fidelity, including the approaches by which these were evaluated. This is an active field of research, with multiple approaches to the reconstruction process proposed and implemented. Initial understanding of these are sought, though the literature review, to identify general categories of approach and position the starting point. Of particular interest is the overlap between model forming approaches and image based rendering techniques as the initial conceptual implementation draws on features of both. There is a significant corpus of material for this field, with many of the published articles also presenting some analysis of reconstruction performance and subjective, visual presentation of quality. However, in initial literature reviews evidence of consistent approaches to speed measurement and objective evaluation of fidelity do not have strong representation. Reported used of a simulation framework for supporting developmental research is virtually non-existent. Therefore the specifications, definition and realisation of the simulation framework and evaluation metrics, while guided by the literature review, are established from first principles.

2.2 Case for 3D Reconstruction of a Person

Reconstructing the face-to-face meeting has long been a goal for tele-presence and collaborative virtual environments [77]. The most important factor in this is the representation of the participants within the shared collaborative environment. Traditionally this has been enacted through the use of 3D geometric representations of a person (an avatar) which mirrors the participant's actions though body tracking [12]. While it is important to ensure holistically congruent virtual characters, especially in immersive settings [94], the avatars in question are not synthetic virtual characters, but 'iconic' representations linked to the actions of real people.

To ensure that communication and collaboration are as effective as possible it is desirable that the human participant has a similar empathetic engagement with the virtual avatar to that they would experience if they were face-to-face with the other participant [71]. However, the human being is a complex system, presenting high levels of expression in non-verbal cues that need to be captured and remotely represented. These are not portrayed by traditional pre-constructed, geometrically modelled, avatars. As computational power has increased and with the development of newer higher quality, multipoint tracking systems, the approach has remained similar, but enacted with a finer level of detail. Connecting these user tracked avatars to animation models further expands the range and scale of the avatar features that can be portrayed by determining appropriate animation sequences that can be triggered to represent identified emotional states [63]. However, the result is an iconic virtual representation of the tracked user that bases gross feature on actual tracked data and fine detail on implied simulation [91]. Furthermore, in order to create an effective, interactive 3D environment non-verbal gestures must be continually updated to reflect the person and enable low latency communication and collaboration between the real and virtual occupants of the immersive environment. Failure to meet these conditions leads to an abnormal representation that debilitates the real occupant as they cannot emotionally relate to their virtual colleague [75].

Attempts to incorporate a live captured, and reconstructed, person, as opposed to a synthetic avatar, within interactive 3D environments are not new and have their origins in the early VirtualPlace experiments where a single silhouette of the participant was projected into the graphical environment [41].

The recent availability of high quality video capture, and support for network, and computational infrastructures that can effectively utilise these, increases fidelity of source data. Given this, researchers have made attempts to directly capture the animated form of the real user and use this information to create accurate representations of them within a synthetic environment [80]. Approaches, such as the free viewpoint video system, pioneered by the BBC [28], uses multiple images to compose a 3D image that can be navigated in the same manner as any other 3D environment. However, the typical approach is to carve a volume, prior to surfacing with a tessellated geometric topology, which is subsequently textured with the image data captured from the original sources [100, 30]. While effective, the algorithmic complexity of this approach makes it ill suited to the real-time requirements of tele-immersion. In addition, the requirement for an uncluttered environment, in which a large number of cameras can be positioned, is difficult to achieve in an immersive virtual environment where the user must occupy an enclosed space that forms the projection surfaces for the display system itself.

The Blue-C display system [79] attempted to resolve some of these issues. This adaptation of an immersive projection system utilises transparent screens that can be switched from an opaque state to a transparent state to enable synchronised cameras to capture imagery through the screen material, while also enabling projected imagery to be displayed on the screens during the opaque phase of operation. While this approach enabled the research and development of a real-time free viewpoint solution, issues still exist with respect to the performance and accuracy of the reconstruction approach and the compromises that must be made between the immersive projection system and the capture systems [83].

The Cisco Tele presence system [76] presents an alternative vision for how tele-presence could be achieved. In this case physical infrastructure establishes fixed seating positions and screen placements to create the impression of an integrated space; one half real, the other virtual. While this offers a high quality tele-presence experience [101], it does so without attempting reconstruction, instead relaying on fixed alignment of spaces within a traditional video conferencing environment. However, this approach does indicate quality metrics for the visual form which present the presentation of a remote participant in 1:1 scale using HD (1920x1200) resolution displays at 30hz refresh.

The European Integrated Beaming project (FP7-ICT: 248620) is seeking to expand further on the enaction of tele-presence by transcending the virtual/physical barrier to enable remote embodiment of a participant which is able to physically interact with the displaced location [82]. In this the use of mixed reality, multi modal displays enables participants to interact with others through a range of capture and display devices that enable multi-sensory embodiment and communication mediated through a range of technologies including spatialised audio, haptics, mobile embodiments, etc. [85]. While the use of tracked participants linked to a 3D avatar is commonly used within the project, the inclusion of a additional spatially located devices utilises video and depth imaging to locate embodiments in the physical environment [68, 86].

2.3 Techniques for 3D reconstructions

Traditional approaches to representing a virtual user within an immersive collaborative virtual environment have sought to connect an artistically generated character (avatar) to a tracked human body. As computational power has increased, and with the development of newer, higher quality, multipoint tracking systems, the approach has remained similar, but refined to ever finer levels of detail. Connecting these user tracked avatars to animation models further expands the range and scale of the avatar features that can be portrayed by determining appropriate animation sequences that can be triggered to represent identified emotional states [63]. However, the result is an iconic virtual representation of the tracked user that bases the gross features on actual tracked data and fine detail on implied simulation.

Video-Based 3D Reconstruction (VBR) is a set of techniques used to recreate the 3D visual appearance of a moving thing. The use of such techniques is growing in applications such as film and TV production, sports commentary and collaborative virtual environment applications/tele-presence, particularly as a way of capturing and representing participating actors. A virtual human created by such means is fundamentally different to a conventional CGI avatar, most notably because it approximates a faithful reproduction rather than an iconic one. Of particular interest was the use of VBR to reconstruct people in a remote space to support real time non-verbal interaction in collaborative, interactive 3D environments. While it is important to ensure virtual characters are recognizable and engaging, especially in immersive settings [94], conventional avatars are not synthetic virtual characters, but iconic representations of real people. To ensure that communication and collaboration are as effective as possible it is thought desirable that the human participant within the immersive environment has a similar empathetic engagement, with the virtual character, to that they would experience if face-to-face with the other participant. Fundamentally, the capture of features, such as facial expression, eye-gaze, body positioning, etc, impart important visual cues that enhance the flow of communication between participants [84]. Failure to meet these conditions leads to an abnormal representation that debilitates the real occupant as they cannot emotionally relate to their virtual colleague [75].

In conventional approaches each image is projected into the reconstruction space, either as a segmented silhouette, or a profile curve derived from it. The geometric shell is then defined as the surface of the volume enclosed by the sum of each projected shape. This process is commonly referred to as space carving [44, 42]. The resultant shell is tessellated to form a set of 3D vertices and indices denoting draw-able triangles, in some cases these are optimized into efficient, render-able geometries [100, 30, 58]. Further evaluation of the vertices maps texture data from the camera images to the form before the reconstruction is passed into a 3D rendering process for graphical display from any viewpoint [56]. In theory, if this can be achieved fast enough, with the computational processing taking place between each frame refresh, a new geometric form and texture set can be displayed for each rendered frame, thereby achieving the goal of real-time animated 3D reconstruction [1]. This creation of an approximation to the visual hull is the most established method of video based reconstruction. Laurentini [44] and Matusik [58] demonstrate how projection of multiple, vectorized silhouette edges enables the computation and shading of visual hulls (geometric shells denoting the form surface) which reconstruct 3D humans. Yue [103], further expands these approaches to incorporate human body part segmentation, thereby overcoming the difficulties within visual hull approaches to construct concave regions and applying the techniques to reconstruction of articulated human reconstruction. These polyhedral approaches are further developed by Franco and Boyer [1, 24] to demonstrate how, with sufficient performance, these reconstructions can be utilized in mixed reality environments.

These geometric approaches seek to resolve the 3D reconstruction as a holistic whole which generates a 3D model that can be viewed from any angle. As such they are the most popular techniques for human reconstruction. However, these approaches impose high computational complexity inherent in the back-projection of points to all silhouettes. Attempts to overcome this have predominantly sought to adapt or optimize heavy weight CPU based processes to utilize cluster networks or GPU processing for speedup of critical sub-processes [102]. However, this approach does not inherently scale as more computational resources are added and significantly increases system costs [56, 93]. In addition, the formation of a 3D form, enacted prior to the texturing and display, disconnects two stages of the process that are effectively utilising the same data. The result of this is that discrepancies in the generated form, either due to mis-matched camera calibrations, or localised depth artefacts, are further emphasised during the 3D rendering process when source images are applied to the surface within a texturing process. Floating Textures [20] seeks to reduce this problem by applying texture through view-point weighted, projective texturing which seeks to match the best image source to the viewer position. Further elaboration of this technique also allows for depth based assessment of projective texture occlusion to reduce additional effects such as ghosting within the textured form.

A common alternative to polyhedral hull forming is to use projected source images to populate a volume space comprised of voxels [13]. This is typically enacted as a pre-processing stage, delivering a 3D texture (voxel map) containing the reconstruction form's occupancy, and in some cases, localized colour values. In a few instances, this volume model is rendered directly, with the populated colour voxels denoting the surface texture of the reconstruction form [66]. However, it is more common to skin a binary voxel set, which simply denotes occupancy, typically using a derivation of the marching cubes algorithm and apply texture to the resultant geometric form [13, 25, 56, 30, 26, 90, 102]. In recent years the emphasis of research in these techniques appears to have shifted to concentrate more on the mapping of texture to a surface of sufficient quality [20, 57]. Yue further expands these approaches to incorporate human body part segmentation, thereby overcoming the difficulties within visual hull approaches to construct concave regions and applying the techniques to reconstruction of articulated human reconstruction [103].

Depth Based Image Rendering (DIBR) techniques seek to minimize the computational load by attempting to create a reconstruction that is valid from a single viewpoint, rather than solving a complete model [89]. Typically these approaches project source image(s) into the 3D environment where they are mapped onto an intermediate geometry located at the approximate spatial position of the reconstruction target [16]. Whilst effective these are often limited solutions that constrain the motion of the observer within tight proximity to the position of image capture [21]. These approaches have an inherent lower computational overhead than view-independent geometric solutions with high texture fidelity, but generally lack the three dimensional depth of geometric solutions.

High Dynamic Range (HDR) imaging [53] is a potentially enhancing technology that could improve the fidelity of reconstruction. The extended range of image fidelity, which enhances contrast in shadowed regions, offers a significant improvement in the potential problem areas for person reconstruction such as eye sockets, where shadows hinder the capture of features such as eye (gaze) direction. Benefits have already been demonstrated for narrow baseline stereoscopic imaging [54] and recent innovations are seeing adaptations of the techniques to support video capture [92] which would support video based 3D reconstruction. 3D reconstruction based on the combination of depth and HDR imaging is reported [36], suggesting that processing of the HDR composition is possible within the latency constraints for real-time reconstruction for a single HDR imaging device. However, it is uncertain if this would be possible for multiple video streams. The capture and initial processing of imagery is outside the scope of this research, but the delivered imagery would be direct replacements for the imagery current acquired through traditional video cameras.

2.4 Analysis of 3D reconstruction

Objective, quantitative analysis of 3D representation is rare in published literature. Authors of the many reconstruction techniques typically present little more than mathematical explanations of their approaches which are more concerned with the core principles of hull forming, processing architecture, etc [56, 59]. With most of the research raw performance (i.e. speed of reconstruction) is seen as important, however this typically considers only the reconstruction time and does not include the additional rendering and associated data preparation time. All the published approaches utilise different hardware configurations and platforms and leverage these to achieve broadly similar performance (speed of calculation) rates. In most cases these are close to that considered acceptable in video (30fps), however, the wide range of differences in approach, system architecture and hardware configuration make objective comparison of the quality and performance of different techniques extremely challenging [39].

Whilst basic measures of performance are commonly presented, little other than the inclusion of example images is typically presented to demonstrate quality. Stroia-Williams [87] and Matsuyama [56, 102] present comparative images of their resultant form and discuss how these compare against the source of their reconstruction. Similar demonstration of quality is also presented for recent developments in the application of texture (Floating Textures), where a set of comparative images are presented, denoting the implications of differing approaches on the resultant textured form, which relate the differing approaches to a ground truth image for subjective review [20]. Kepplinger [38] argues that adequate approaches for robust subjective assessment of free viewpoint video is lacking and more needs to be done to define the quality influencing factors that might formulate a framework for evaluation. In Depth Image Based Rendering Techniques (DIBR) techniques such as Peak-Signal-to-Noise-Ratio (PSNR) and Video Quality Metric (VQM) are commonly used to assess the resultant representation, from a single viewpoint, of 3D video delivery [22] and these form the basis for identifying quality improving techniques [18]. Kilner [39] presents a framework for objective quality assessment in free-viewpoint video production. In this dissatisfaction with subjective analysis and conventional pixel-wise evaluation is expressed, arguing that these measures do not reflect the quality of view synthesis as perceived by the user. In this model, an evaluation of image pairs, based on a leave-one-out test, is used to measure reconstruction quality rather than fidelity, presenting measures in quality of reproduction of shape and appearance rather than PSNR and VQM. Berger identifies this framework as being more suited to the measurement of quality of 3D reconstruction in image space, pointing out that this is only applicable to geometry based 3D reconstruction techniques of the type presented by the research [9].

The recognition that a truly objective assessment framework for video-based 3D reconstruction is lacking is evident. While attempts have been made to address this the techniques applied are either subjective, or specific to the DIBR approach and concentrate on sources of error within the capture/reconstruction pipeline rather than targeting the specific process of 3D reconstruction itself. While Kilner does present a framework for objective assessment that attempts to isolate the quality of the reconstruction process this is still embedded in the live capture pipeline and highly specific to the viewpoint chosen.

2.5 Rational for research direction

While grounded in previous research this work has a novel standpoint. Given previous experience in volumetric rendering and the associated graphics shader rendering processes [4] it was felt that any reconstruction process that required the computationally expensive, and traditionally utilised, processes of surface formation, tessellation, decimation and rendering [30, 6, 27] would always struggle with performance and scalability. However, by using approaches inspired by the previous volumetric rendering work, it was felt that the basis volumetric technique for rendering could be adapted to directly render the typical pre-calculated volume dataset evaluated by space carving, rather than performing an additional tessellation step to obtain a renderable polygonal surface. In the final iteration of the simulator system the need for pre-calculating the 3D volumetric image is removed by utilising the projective texturing and multi-textureing features available through the use of graphics shader programming.

This differs to recent published approaches, which have sought to apply hardware parallelism and processes networks to improve performance. Typically these attempt a view-dependent plane-sweeping strategy [50] or a modified version of a plane based volume intersection algorithm [47, 48] and improve performance by optimizing sub-sections within the sequential pipeline. Some GPU Accelerated Visual Hull approaches do exist [43]. The approach researched here is similar to these, in that it shifts much of the calculation onto the GPU. Early investigations (Chapter 3) calculated a 3D volumetric texture [42], on the CPU, which was then rendered through hardware-accelerated plane/volume intersections [8], enacted on the GPU, to display a lit visual hull. In this approach's evolution a projective texturing solution was originally sought, which could be applied to the already generated visual hull to texture the visual form. However, in seeking this solution an alternate approach became apparent. Projective texturing has been utilised to project an image onto a aligned view plane for multiple image reconstruction with a significant gain in performance when enacted using the GPU [73]. This only partially resolves the visual form. The realisation made was that this approach could be applied directly to the slice planes already generated to render the 3D volumetric image, as part of a multi-textured environment where the shader routine is provided with images for each virtual camera and their appropriate transformation matrices.

In principle, this is much the same as the general approach taken in the Floating Textures [20] technique for applying multiple textured images to a pre-processed geometric form. However, the Floating Textures utilises this approach to minimise the visual artefacts generated by attempting to texture an ill fitting geometric form. In this case the potential for this problem is reduced as both the form and the texture application come from a single stage process in which both are considered against a consistent calibration set. This means that the texture projection is evaluated against a form calculated from the same source data and camera calibration set, thereby reducing the possibility of error.

2.6 Chapter Summary

This work seeks to define approaches for the real-time 3D reconstruction of a human for application within collaborative tele-presence communication systems. The inspiration for the research comes from both a long term interest in collaborative virtual environment systems, and an observation that the vast majority of current approaches seek to adapt techniques drawn from Film and TV industries which are originally designed to maximise quality in a none real-time context. These approaches essentially undertake a shape from silhouette, space carving technique. This progressively builds up form through processes akin to computational solid geometry (CSG) summation of the projected silhouettes, tessellation (and decimation) of the resultant surface and evaluation of texture mapping to apply surface detail. Adoption of these essentially sequential approaches, through the parallelisation of key operations within their pipeline offers performance gains, especially when the rapidly evolving computational power of GPU processing is utilised. However, each stage is essentially a sequential activity that requires cross referencing of data within the evolving data set. This is a short term approach that does not effectively exploit the potential offered by the evolving GPU processing domain.

Chapter 3

An Initial Study into Voxel Based Evaluation and Rendering of 3D Reconstructed Human-like Forms

The ultimate aim of this work is to allow two people in different places to walk around the same virtual place whilst seeing what each other looks like and is looking at. In the medium term, the approach to this is to link two immersive display and capture spaces so that the holographic form of the occupant, in each system, is displayed within the other. This approach is nothing new and was pioneered by Blue-C [29]. However attempts to date have failed to reproduce facial features to a quality where either gaze or expression could be accurately determined. This research contributes to work that is trying to reconstruct a good likeness of the head and face from all sides in real time. This *is important where people can move around a virtually shared space. Specifically this work contributes towards this by building and using simulators that allows experimentation with camera placement, reconstruction algorithms and rendering processes. In this way exploration of the potential solutions can be made without having to rely on temperamental research infrastructure. Virtual cameras can be configured and adjusted in seconds without clambering on ladders trying to fit expensive cameras above costly, delicate screens. Algorithms for reconstruction and rendering processes can be evolved rapidly within a repeatable and stable experimental environment.*

In essence this is an initial study into development of technique from a naive standpoint. However, the most important element of this chapter is the formation of the first iterations of the simulation platform that enables exploration of the reconstruction pipeline. These tools, while crude, enable exploration of the capture configuration and develop understanding of the pipeline sequence. Within this approaches to assessing performance are also presented. This is initial work in defining measures that provide objective analysis to support future iterative development. The conclusions of this chapter show that while the approach to reconstruction is flawed, the framework for developing and analysing the reconstruction is, even in this initial state, a powerful tool for supporting the exploration, discovery of knowledge and development of these reconstruction/rendering techniques.

This is drawn from an article presented at CVMP 2010 [2]

3.1 Introduction

Real-time 3D reconstruction of people promises to combine the winning characteristics of video conferencing and immersive collaborative virtual environments. However, it remains difficult to obtain a reconstruction of a person with recognisable facial features in real-time, as the person moves around. Three highly related issues that require further work are camera placement, reconstruction algorithm and rendering processes. In order to speed up development and concentrate on core principles a series of software simulators have been created in which all three can be varied without the need to reconfigure a physical system. These are described and used to identify and report evaluation of camera placement, 3D reconstruction algorithms, and texturing. This also presents the initial 'brute force' approach to 3D reconstruction without tessellation. The approach taken is not to create a surface on which to texture, but to directly use image data for shape throughout and specific graphics shader functionality to render, light and texture it. The acid test of performance was comparing what the simulated approach could achieve in terms of frame rate and pixel density and comparing this to state of the art commercial tele-present video conferencing.

The work described applies a simple principle: Tessellation (surface forming) is a computationally expensive and un-necessary process. The advent of access to fast processing performance directly located within the rendering pipeline has enabled the significant redefinition of the rendering processes and techniques for real-time computer graphics [37, 99]. Utilising this approach the 3D volumetric data, generated through a projective space carving approach [42], was directly rendered against an orthogonally aligned 3D planar space, applying lighting and texture through modified shader routines.

This initial study tests this starting point concept, that 3D video based reconstruction is possible without surfacing, by adapting techniques inspired by previous research into medical imaging towards the display of a 3D volumetric form.

3.2 Approach

Given the complexity of both the hardware and software, investigations were started by building a simulator that would allow experimentation with issues relating to the reconstruction and rendering process in an easily reconfigurable and accessible way. Additionally this would enable experiments to explore camera positioning without the laborious process of physically moving and re-configuring a complex hardware system, thereby allowing much more efficient use of time.

The simulators have undergone two iterations. These started with a simple proof of concept solution that enabled free positioning of virtual cameras around a geometric model, and one shot 3D reconstruction based on the images captured by each virtual camera and the transformation matrices of each giving a quick proof of concept demonstrator (Figure: 3.1). The second generation simulator was created to provide a platform for repeatable experimentation into the reconstruction and rendering processes by enabling the definition of set configurations of camera position and model (Figure: 3.2). Plugin modules were developed for the reconstruction and rendering

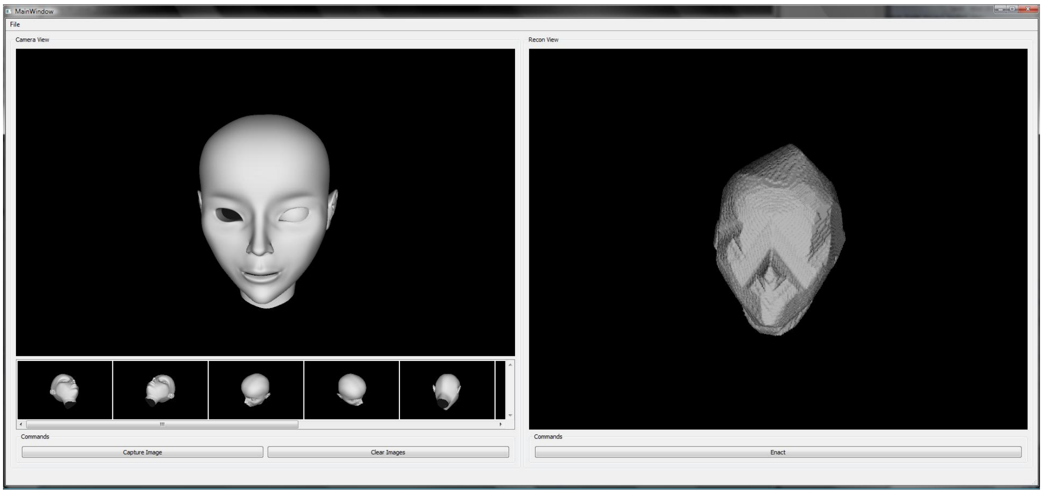


Figure 3.1: First Generation Simulator: Proof of concept stage (Images taken from 1st person perspective)

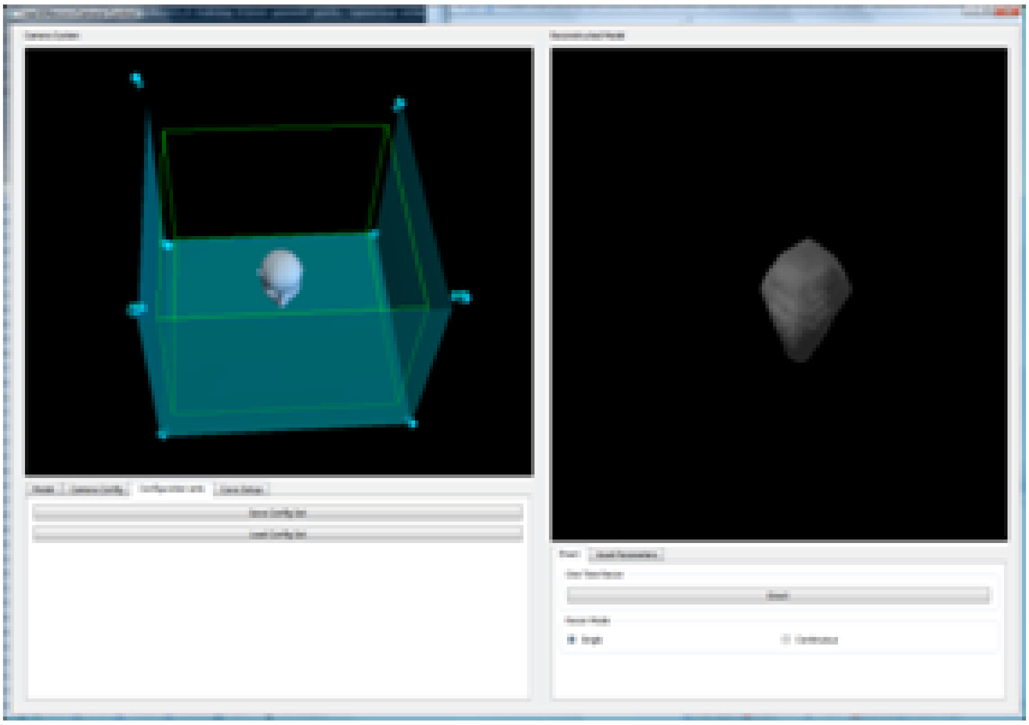


Figure 3.2: Second Generation Simulator: A configurable platform for experimentation into reconstruction and rendering

processes. Side-by-side visual comparison of the reconstructed form against the original model provided subjective evaluation. Unlike the first simulator, cameras were configurable entities that could be manipulated within the space. Models of physical objects such as display walls were also added to help understand the space and where the cameras can be placed. This enabled refinement and evolution of the reconstruction and rendering processes. Reconstruction was enacted by projecting each camera image into a volumetric voxel space, enacted on the main CPU of the host computer. The rendering process utilised a technique adapted from research into medical visualisation [4] to directly render the 3D image without tessellation of the reconstructed form through the use of bespoke graphics shader routines. It was felt that any reconstruction process that required the computationally expensive, and traditionally utilised, processes of surface formation, tessellation, decimation and rendering [30, 6, 27] would always struggle with performance and scalability. The approach used here applied the base volumetric description, necessary to start the surface formation process, to directly render the resultant form, rather than performing an additional tessellation step to obtain a polygonal surface. In the final iteration of the simulator system the need for pre-calculating the 3D volumetric image is removed by utilising the projective texturing and multi-textureing features available through the use of graphics shader programming.

3.2.1 Requirements

The reconstruction approach must be scalable both in terms of input data and reconstruction. The complexity of input data is in terms of size of image, number of images and the size and occupancy of the subject in the reconstruction volume. The complexity of reconstruction is in terms of spatial accuracy of reconstruction and calculation time. A scalable solution would offer consistent performance regardless of the occupancy of the reconstruction volume and the format of the source data. Realistically, the number of input sources (cameras) will always have an effect, as each source stream requires processing, however, the relationship between calculation time and number of input sources should be no more than linear. A realistic goal of the output is a linear relationship between computation time and number of spatial samples (voxels) to be computed. This differs to the traditional consideration of direct volume rendering as a $O(N^2)$ complexity problem as the spatial sample points (voxels) are independently processed without reference to others. Therefore, in order to optimise the use of parallelism, there should be a direct relationship between processing power and computation time.

State of the art video conferencing, such as Cisco Tele-presence, runs at 30Hz in HD resolution. At frame rates below 15Hz smoothly moving objects appear to jump. It is important, however, to consider the impact of tying the viewpoint to the observers eyes, which is necessary if interpersonal gaze is to be accurately determined [71]. Motion sickness can be caused by latency in viewpoint update when the viewpoint is controlled from head tracking. No more than 40ms latency and at least 30Hz update is necessary to guard against motion sickness. Measures of quality of experience including presence have pointed to 50Hz being preferable in tracked viewpoint systems. The update rate of the display should be between 60Hz for pal and 100Hz for HD. There are not as yet any tele-immersive system approaching 30Hz update at anything approaching HD resolution. However, this is what is needed if it is to become a reliable form of communication. In the experiment interest was in capturing a person within a 3m cube, which is a typical size for a large screen cubic immersive display device.

3.2.2 First Generation Simulator

The initial simulator was devised as a proof of concept demonstrator to see if a projective space carving approach previously developed for medical imaging was suitable for reconstruction of a human head, and to play with camera placement. This simple simulator captured a single image, and the associated transformation matrix, from the current viewpoint. Once enough images had been captured, an occupancy model projects the source image data into a 3D voxel volume (Figure: 3.3). In this initial iteration each voxel is traversed and its location used in an inverse projection of each source image's intrinsic matrix to determine the texel projected into that voxel. The source images, captured from the simulation tools, have a fixed background colour which provides basic segmentation. Any voxel evaluated as having 1 or more projected source background texels is evaluated as being empty (0) opposed to the default voxel value (255) (Algorithm: 3.2.1).

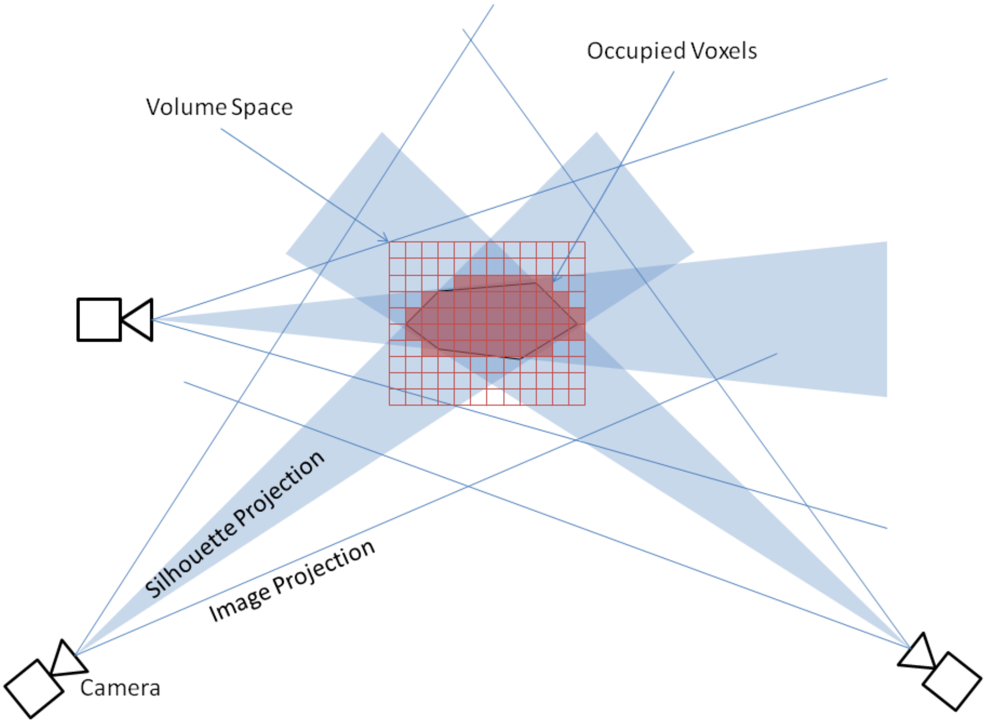


Figure 3.3: Voxel Occupancy Model: Three camera positions with projected silhouette images carving a form at the region of intersection, sampled by a voxel volume

Algorithm 3.2.1: First Generation Simulator: Voxel Occupancy()

```
global $voxels[numVoxels]$
global $voxelLocation[numVoxels]$
global $sourceImage[numImages]$
global $sourceImageInverseProjection[numImages]$

procedure evaluateVoxel($n$)
 for $i \leftarrow 0$ to $numImages$
   do {
     if reverseTextureProjection($voxelLocation[n]$,
     $sourceImage[i]$,
     $sourceImageInverseProjection[i]$)) $= background$
       then return ($false$)
   }
 return ($true$)

procedure populateVolume()
 for $i \leftarrow 0$ to $numVoxels$
   do {
     if evaluateVoxel($i$)
       then $voxels[i] \leftarrow 255$
       else $voxels[i] \leftarrow 0$
   }
```

In the case of the first source image all voxels are tested and either enabled or disabled depending on their occupancy of the reconstruction entity's profile within the source image. For each following image evaluation, only enabled voxels are tested for occupancy of the camera image, with zero occupancy disabling the state of the voxel. In this manner the reconstructed form utilises a reduction occupancy model that offers increasing refinement proportional to the number of images used. An adapted form of the MC slicing algorithm [8] was then applied to generate view direction aligned planes for evaluating the 3D volumetric texture to render the volume image, thus displaying the form. A crude shader routine was also implemented to test each processed fragment to determine if it occurred on the boundary of the form and, it so, calculate a surface normal and apply basic lighting calculations. This enabled the display of a 3D form, rather than a fixed colour shape which would appear as a silhouette without any ability to determine the surface detail (Algorithm 3.2.2 & 3.2.3, Code: Appendix A.1 & A.2).

Algorithm 3.2.2: First Generation Simulator: Vertex Shader()

```
global $vStep$
global $vAmbient$
global $vDiffuse$
global $vLightDirection$
main
  $vStep \leftarrow samplestepsize$
  $vLightDirection \leftarrow normalised(modelViewMatrix * lightSourcePosition)$
  $vDiffuse \leftarrow materialDiffuse * lightSourceDiffuse$
  $vAmbient \leftarrow materialAmbient * (lightModelAmbient + LightSourceAmbient)$
  $gl\_Position = ftransform()$
```

Algorithm 3.2.3: First Generation Simulator: Fragment Shader()

```
global $vStep$
global $vAmbient$
global $vDiffuse$
global $vLightDirection$
global $vNormal$
procedure isEdge()
  $bEdge \leftarrow false$
  $vNormal \leftarrow 0.0, 0.0, 0.0$
  comment: Repeated for vStep = 26 surrounding positions of current fragment
    $vPos \leftarrow current(vStep)$
    $vCol \leftarrow$ texture3D$(volume, textureCoords + vPos)$
    if $vColour.Alpha! = background$
      then $vNormal \leftarrow vPos$
      else $bEdge \leftarrow true$
  if $bEdge$
    then normalise$(vNormal)$
  return $(bEdge)$

main
  $vColour \leftarrow$ texture3D$(volume, textureCoords)$
  if $vColour.Alpha > 0$
    then
      $gl\_FragColour \leftarrow vAmbient$
      if isEdge()
        then
          if dot$(vNormal, vLightDirection) > 0$
            then $gl\_FragColour \leftarrow vDiffuse*$
          dot$(vNormal, vLightDirection)$
    else $discard$
```

This approach proved robust and reliable, proving the approach worked but little more. The free hand positioning of camera for capturing images meant that, while this was a quick and simple demonstration, repeatability was poor. In addition the calculation time, for projecting each of the images into the 3D volumetric texture space, was high (20-25 seconds). Rendering efficiency was not considered important, as with comparable research by others, and was not included in the time measurement. To explore this approach further a more dependable and configurable platform for experimentation was required.

3.2.3 Second Generation Simulator

In order to explore the reconstruction and rendering processes more deeply a second generation simulator was constructed. Instead of simply capturing an image from the current viewpoint this evolved system provided a set of virtual cameras which could be placed within the environment with their configuration stored for reuse in repetitive trials of the evolving reconstruction and rendering systems. These stored configurations also included details of the geometric model used and facility was also provided to import camera position and characteristics from the calibration data of the physical oCtAVE system to enable matching with the real-world environment.

In evaluating the initial reconstruction approach the most significant time element affecting reconstruction time was determined as the enaction of the projective texture transformation for mapping texels to voxels. As the camera positions and the volume region would remain constant during the reconstruction loop removal of this could be pre-calculated before reconstruction processing commenced and the mapping accessed during the reconstruction cycle. To address this a direct mapping of the pixels, comprising imaging plane to the voxels occupying the pixel projected ray within the reconstruction volume, is maintained for each camera. This mapping is contained within a 3 dimensional array, of the same parameters as the reconstruction volume, with each entry being the texel index for the image projection into the voxel space. Each source image requires a unique mapping array. While computationally expensive, and requiring a high memory overhead, this mapping is only performed once, based on the assumption the neither the reconstruction volume, nor the cameras, will move during the capture and reconstruction runtime. As a result of this pre-calculated mapping, reconstruction of sequential frames of source images need only traverse each and every voxel within the reconstruction volume and iterate through all of the camera images for each voxel, performing the same voxel evaluation as in the previous iteration of the system (Algorithm 3.2.4).

Algorithm 3.2.4: Second Generation Simulator: Voxel Occupancy()

```
global $voxels[numVoxels]$
global $sourceImage[numImages]$
global $sourceImageMap[numImages]$
procedure populateVolume()
 for $i \leftarrow 0$ to $numVoxels$
   do {
     $bOccupied \leftarrow true$
     for $n \leftarrow 0$ to $numImages$
       do {
         if $sourceImage[n][sourceImageMap[i]] = background$
           then $bOccupied \leftarrow false$
       }
     if $bOccupied$
       then $voxels[i] \leftarrow 255$
       else $voxels[i] \leftarrow 0$
   }
```

Each voxel within the reconstruction volume is considered independently of all others. Due to this the algorithm offers significant potential for parallelisation. For performance trials a variant of the algorithm has been created which uses OpenMP multiprocessing to utilise the multiple cores of modern processors.

The resultant volumetric data set is directly rendered using a forward mapped orthogonal slice plane approach derived from previous medical imaging research [75]. Post processing of the voxel data is undertaken at render time using GLSL shader routines to both smooth the resultant form and apply a per-pixel Phong illumination calculation [40]. These processes are similar to those used in the first simulator. This approach offers sufficient rendering performance for real-time immersive applications (approx 100 fps) while also removing the need for computationally expensive surface tessellation that is needed for traditional polygon rendering approaches. The use of the GLSL shader routines facilitated the integration of image data as surface texture onto the geometric form [51].

3.3 Experimentation and Evaluation

All tests were performed on the same laptop computer (DELL XPS M1730, Intel X9000 Dual Core processor, 4GB Ram, dual nVidia 8800 GTX, Vista 32bit). This provided fast processors and adequate available memory for testing purposes; however, memory read/write speed was low compared to many graphics workstations. This system was chosen as it would provide a stable, dependable platform for development and testing, while the systems used within the reconfigurable laboratory would undergo many configuration changes over this period. While this is relatively modest laptop, the results shed light on the potential hardware requirement for an operational system capable of enacting a continuous reconstruction at the frame rates required for interactive immersive environments.

3.3.1 First Generation Simulator

While a crude prototype, the first generation of the simulator did enable rudimentary experimentation with camera placement. Fundamentally the requirement was to identify if the physical limitations of the large scale visualisation facility would present significant problems to the arrangement of cameras. Evaluation of the resultant form was highly subjective, based on a visual comparison of the original geometric form against the rendered reconstructed one.

The immersive display and capture facility has eight movable display walls. Initial simulated tests have concentrated on the classic cubic configuration typically used in immersive collaborative virtual environments [81]. In this case cameras cannot be placed so that they are often in the line of sight between the occupant and the display screens, while the occupant looks around the environment and into the reconstructed face of a remote participant. Such occlusion not only reduces their feeling of presence, but precludes someone looking into the projected face of a remote participant while being filmed from directly in front of their own face.

Initial trials placed the cameras at the eight corners of the cube. This, however, reproduced heads that looked more like a classic alien than a human as both the top and bottom were stretched into a cone, (Figure: 3.4, top right). This was due to the projected intersection between the oblique angle camera positions rather than any perspective distortion. In contrast to the first configuration, forgetting about the restriction of occluding the display, placing the cameras in an evenly spaced ring around the head offers a better reconstruction of the head profile. However, without off axis imagery any protuberances in the facial features are extruded horizontally round the reconstructed form (Figure: 3.4, bottom right). An approximation to an ideal placement of cameras requires both a ring of cameras surrounding the head and also cameras located both under the chin and above the head to accurately represent the more pointed features of the face, such as the nose, chin and mouth (Figure: 3.4, bottom left). In addition, an ideal configuration of cameras also requires two cameras to be located on each side of the head,

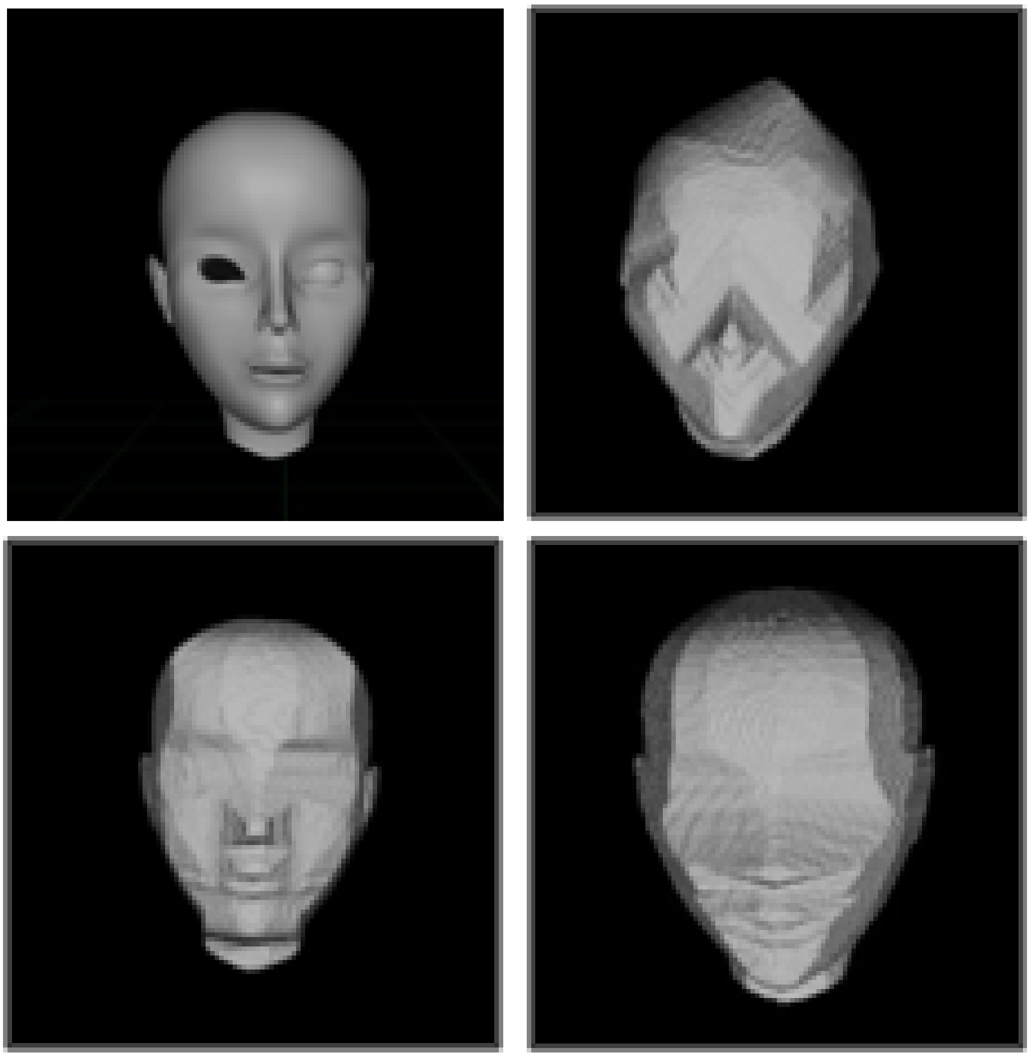


Figure 3.4: Camera positioning, (Top left: Source Geometry, Top Right: CAVE corners, Bottom Right: eye-level, Bottom Left : ideal

ideally behind the ears, with their central view axis aimed at the bridge of the subject's nose. These final two cameras enable the eye sockets to the carved into the reconstructed form, thereby generating a geometric structure that is recognisable as a face. Clearly this is not a realistic solution, while arranging a ring of cameras is potentially possible, by filming through the display systems screens [29], any arrangement that requires cameras to be located relative to the reconstruction object will not work in a system intended for free movement.

3.3.2 Second Generation Simulator

The second generation simulator offered significantly more advanced control over the virtual cameras and was used to create a repeatable configuration for assessing the performance and visual quality of the reconstruction and rendering processes. Each evolution of the processes was assessed against 4 key requirements which sought to determine a solution that was scalable and computationally consistent. This aimed to refine delivery of a rendered image within a consistent time frame, regardless of the occupancy of the reconstruction volume, ideally independent of source image resolution and which would scale to the multi-processor environment within our facility. All tests measured the actual calculation time of the algorithm, by querying the difference in the processor performance counter before and after the execution of the algorithm and multiplying the result by the processor frequency count to obtain an execution time in seconds. For all, apart from the multi-processor test, the algorithm operated in a single thread of execution and

No. Cameras	Low Occupancy	Medium Occupancy	High Occupancy
1	0.29	0.29	0.29
2	0.57	0.57	0.59
3	1.08	1.1	1.06
4	1.16	1.17	1.15
5	1.457	1.449	1.463
6	1.799	1.781	1.756
7	2.067	2.089	2.091

Table 3.1: Occupancy Test Results: Calculation time (seconds) for differing occupancy levels of the reconstructed form within the reconstruction volume against number of cameras (images) used for reconstruction: Low Occupancy=1.6%, Medium Occupancy=15%, High Occupancy=100%

was locked to a single processor. In each test the average calculation time for 3 consecutive computations was recorded, with this process repeated 10 times, with a restart of the application between each. Results quoted are the average times for each set of 10 executions.

Ideally within a real-time reconstruction process the time for each reconstruction frame should be constant, regardless of the composition or configuration of the reconstruction form. This will ensure there are no temporal anomalies between frames of reconstruction removing the chance of slowdowns and stutters in performance as the reconstruction subject changes. To assess this an occupancy test is used. In this test the geometric model, occupying the virtual reconstruction volume was modified in size to occupy a differing proportion of the volume, thereby assessing reconstruction time with different configurations of the model. Theoretically, each sample point (voxel) within the volume should be operating the same test, assessing the inclusion of the voxel within the reconstruction form as a product of the projected camera (source) image pixel's alpha value. This means that re-

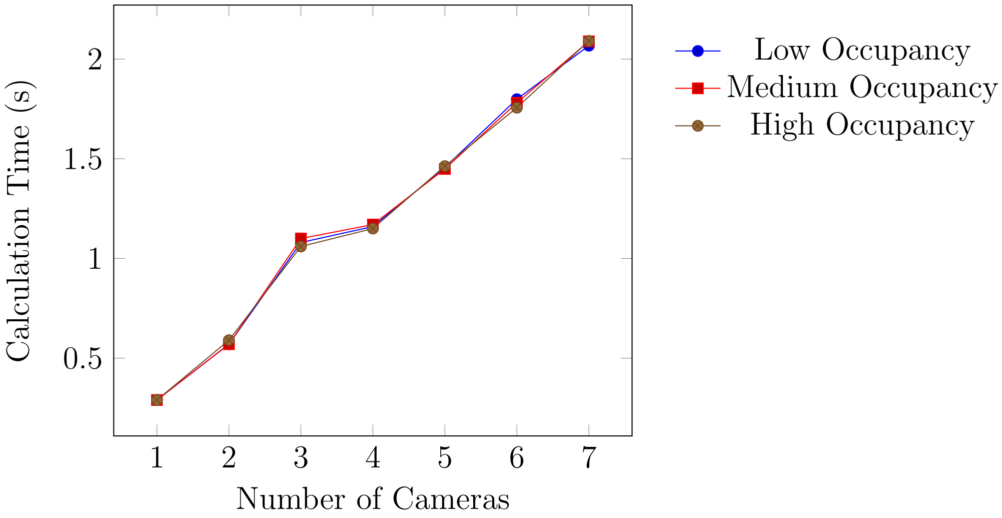


Figure 3.5: Plot of occupancy tests (Table: 3.1), showing calculation time is independent of form's occupancy of the reconstruction volume

construction performance should be constant with variation occupancy and have a linear relationship with the number of source images used.

Trials were enacted at 3 occupancy levels; 100% (High Occupancy), @15% (Medium Occupancy) and @1.6% (Low Occupancy), with the low occupancy variant representing the occupancy level of a normal head in a real (rather than simulated) CAVE, and the medium occupancy level equating the occupancy of the head model at 1:1 scale on a typical large scale monitor (60" HD display). Each test was enacted for between 1 and 8 cameras. The resolution of the voxel space was 512^3 voxels and that of the input images 1024^2 pixels.

From the plot of the results (Figure: 3.5) we see that in the case of our algorithm the occupancy of the reconstruction volume has little effect on the calculation time; however the number of cameras used, and therefore images to be evaluated, has a significant impact. The linear relationship between number of cameras used and reconstruction time demonstrates efficient im-

No. Cameras	128^2 Pixels	256^2 Pixels	512^2 Pixels
2	0.566	0.587	0.582
4	1.188	1.202	1.177
6	1.749	1.751	1.806

Table 3.2: Source image resolution tests showing reconstruction time (seconds).

plementation of the algorithm and reflects the $0(N)$ relationship predicted. The consistent reconstruction time also shows that the approach is not prone to temporal jitter as an animated reconstruction form changes occupancy of the reconstruction volume.

A minor inconstancy occurs with the use of 3 cameras. While this is constant for all 3 occupancy levels, the reasons are uncertain. Investigation suggests this is most likely related to memory utilisation within the test system.

The second test seeks to understand the relationship between image resolution and reconstruction time. Ideally the magnitude of the source imagery (image resolution) should have no effect on the calculation time, as the backward projection used in the algorithm, casting sample voxel back to the source image pixel, should be dependent on the number of sample points (voxels) rather then the source image resolution. In this test the dimensions of the captured images were modified and evaluated against a range of different camera numbers.

The results (Figure: 3.6) suggest that the algorithm's calculation time is not dependent on the quantity of the size of each image frame but is

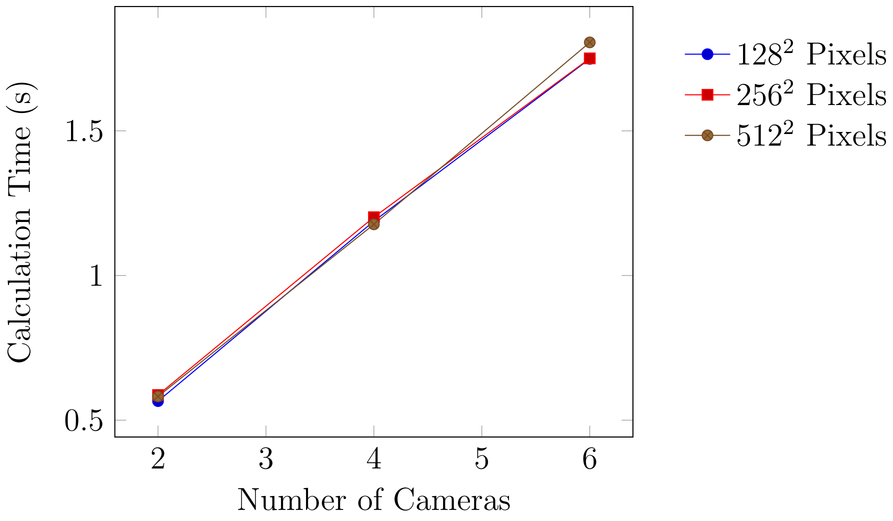


Figure 3.6: Plot of source image resolution tests (Table: 3.2), showing calculation time is independent of source image resolution and linearly dependent on the number of cameras used

again linearly proportional to the number of input sources. This confirms the findings in the Occupancy tests (Figure: 3.5) and demonstrates that the algorithms performance is predictably scalable with respect to number of input sources.

The greater the number of samples (voxels) taken within the reconstruction volume, the greater the precision of the reconstructed form. While memory limitations prevent testing with high resolution reconstruction volumes, testing could be undertaken with lower resolutions to determine the scalability characteristics. At this stage, the reconstruction algorithm is intended for use within the typically cubic display environment occupying a space of approx $3m^3$. Our anticipated resolution for the reconstructed volume within this space is 1024^3 voxels which will give a spatial accuracy of approximately $3mm^3$.

For all resolution, the processing time was directly linearly proportional

No. Cameras	32^3 Voxels	64^3 Voxels	128^3 Voxels	256^3 Voxels
2	$1 \cdot 10^{-4}$	$9 \cdot 10^{-4}$	0.074	0.586
4	$2 \cdot 10^{-4}$	0.02	0.145	1.201
6	$3 \cdot 10^{-4}$	0.028	0.222	1.75
8	$6 \cdot 10^{-4}$	0.042	0.321	2.618

Table 3.3: Reconstruction volume resolution tests showing how reconstruction time (seconds) varies with the resolution of the voxel set used for reconstruction

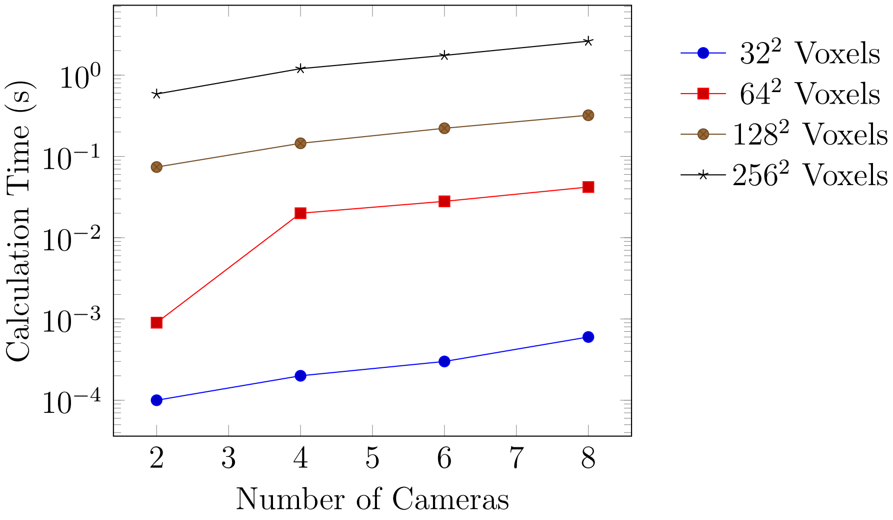


Figure 3.7: Plot of reconstruction volume resolution tests (Table: 3.3), showing relationship between differing resolution of the volume set and number of cameras (source images) used for reconstruction evaluated by measuring calculation time (second)

No. Cameras	1 Job (Single Proc)	10 Jobs	20 Jobs	40 Jobs
2	0.58696	0.290683	0.289558	0.287681
4	1.201596	0.582306	0.565611	0.539308
6	1.750578	0.84017	0.816253	0.803806

Table 3.4: Multiprocessor tests showing how calculation time (seconds) varies with the number of batch jobs that the processing of the volume space evaluation is divided into

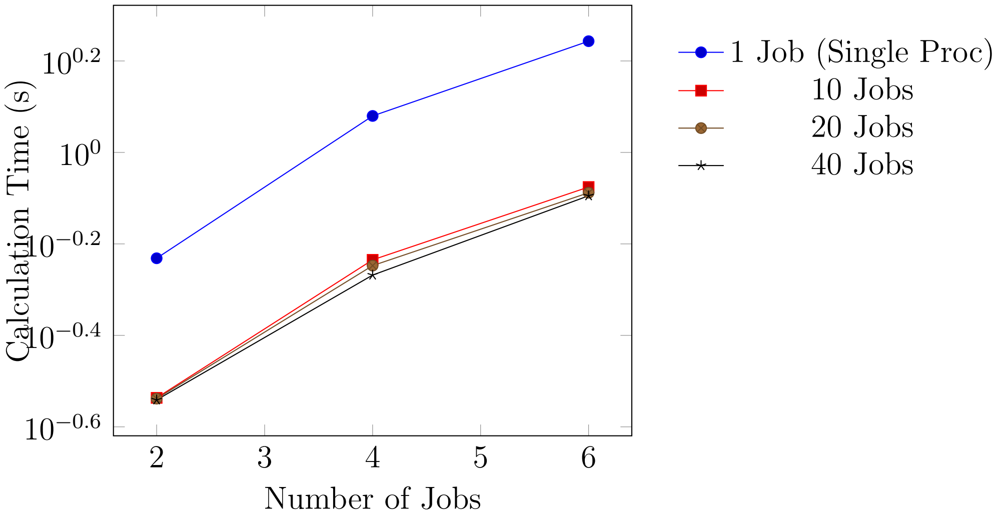


Figure 3.8: Plot of multiprocessor tests (Table: 3.4), showing relationship between batch job subdivisions on the reconstruction calculation time (second)

to the number of cameras (Figure: 3.7), with both the starting point and slope of the graph directly proportional to resolution. For all resolutions, 8 cameras took slightly longer to process than expected if the above linear relationship had continued.

The final element of testing sought to assess how the algorithm would scale to multi-processor systems. In these tests an OpenMP [10] variant of the algorithm was evaluated on the test system with a range of number of cameras. In order to crudely measure impact of load balancing, we used a range of differing job sizes (with the volume space sub-divided into 10, 20 and 40 job units). These results were compared to the case of a single threaded, single process execution. The image size was set to 256^3 pixels.

These results show how the utilisation of multiple processors affects the calculation time of the algorithm (Figure: 3.8). While accurate tests have only been possible for a dual processor system, initial trials have been undertaken on a quad processor system. From this data it can be observed that a halving of the calculation time occurs when two processors are utilised as opposed to performing the calculations on a single core. Observations on a quad core system support this and suggest a linear relationship between calculation time and processor number.

3.4 Discussion

These two simulators has enabled exploration of the 3D reconstruction problem without the need for reconfiguring and manipulating the hardware comprising the physical capture and display environment. While this has obviously avoided some of the issues of large scale system integration, currently being explored by other members of the research centre, it has enabled rapid evolution, development and evaluation tools and techniques for the reconstruction and rendering process in a parallel activity to other complementary research.

The initial simulator enabled a proof of concept, demonstrating that the approach for reconstruction and rendering, drawn from previous medical visualisation research and quickly demonstrating that the, initially naive, concept for camera placement was inadequate. While filming through the side screens might better resolve the facial features, this is a technical challenge that needs to overcome and is outside the scope of this research. While attempts have been made by others, most notably Blue-C [29], the quality of the displayed image is compromised. This exploration enabled ideal camera positioning to be identified, however, it is difficult to see how this can be achieved, even without the constraints imposed by the display facility.

The second simulator enabled refinement of the reconstruction and rendering algorithms used in the initial prototype, however this was insufficient for the end-to-end reconstruction and rendering performance required to enable recreation of the face to face meeting. In addition, the key optimization, the initial mapping of 3D volume voxels to camera image pixels, performed once during the initialisation of the system, was hugely memory hungry. This required a memory address to be recorded for every image that relates voxel to pixel. Therefore in addition to the overhead of maintaining the reconstruction volume data set (512Mb for 512^3 voxels and 4Gb for a 1024^3 voxel set) a matching sized data structure is required for each source image (assuming that the size of a memory address is 32 bits). This results in an overall memory requirement of 5.5Gb for a 512^3 voxel space and 135Gb for a 1024^3 voxel space. While the former is achievable within a normal(ish) hardware system, the latter is pushing the boundaries of even high end server systems.

3.5 Conclusion

The ultimate aim is to build a tele-immersion system that combines wide baseline display and capture to allow people in different places to apparently walk around each other. This is a complex system which will take a few years to perfect. To speed up the process of critical aspects of the research a series of simulators have been built, these are introduced here. These initial investigations enabled identification of camera placements and allowed establishment of whether the desired reconstruction frame rates where feasible. Through simulation it was demonstrated that while six cameras is sufficient for capturing images from which a recognisable reconstruction of a human head can be made, they have to be carefully placed. This placement would not be easy in an immersive display and capture system. It was hoped that by removing the need to tessellate the 3D reconstruction the resultant process would be fast enough for tele-conferencing. A 30Hz update and HD equivalent pixel density was aimed for as a target as this is what is used in todays state of the art video conferencing. A significant disadvantage of the approach is the amount of memory it uses. The test, and very basic, laptop could only support a voxel space of 256^3. When using the main processor on a laptop it took 1.75 seconds to reconstruct the 3D form from 6, 1024^2 pixel, cameras (Figure 3.3.2).

Clearly, while the initial investigation enabled rapid prototyping and experimentation with the capture environment and camera characteristics the naive approach taken to reconstruction was both too slow and too expensive in memory resources to offer a viable solution.

3.6 Chapter Summary

This chapter presented a deeply flawed approach to the problem of 3D reconstruction and rendering, that utilised brute force processing and massive memory overheads to achieve a crude replication of the source object. While it is an approach with a small modicum of novelty there is little that is efficient or effective about it. The key contribution of the chapter is related to the simulator framework. In building these tools and using them to explore not only the reconstruction/rendering approach but also the pipeline and configuration impacts of the capture environment on the resultant form are identified and understood. This has started to establish a novel platform for supporting the evolution of technique through objective measurement that enables rapid prototyping and testing that demonstrates how such tools could support onward design and evolution of technique.

Chapter 4

An Approach for GPU Aligned Integrated Rendering and Reconstruction

This chapter describes and discusses the evolution of approaches investigated and presents the final, refined, novel approach for 3d video-based reconstruction defined by this research. The initial, naive approach originated in previous research for 3D medical imaging and defined what has been described as a massively brutal attempt to map 3D reconstruction in to a non-geometric volume space. While crude the initial approach demonstrated that the principle of moving away from polygonal reconstruction towards a more parallelisable approach offering a direct mapping to rapidly evolving GPU architectures was potentially possible. Informed iterative refinement of this approach offered significant improvements to the generation of a visible representation of a human form, but did little to enhance the visual appearance (textural quality) *of this. This challenge of mapping texture to the representational form lead to a conceptual jump in understanding in which both form and appearance could be evaluated as independent fragments during the graphical rendering cycle. This both eliminates the need to tessellate and better utilises the pipeline architecture of current GPU systems by using a view-point dependent sampling of a view-point independent volumetric space.*

This approach utilises projective multi-texturing to populate a region within the 3D graphical environment. This region is sampled to determine form and surface texture of the 3D reconstructed person using GLSL routines operating on orthogonally aligned sample planes within a single integrated process. The approach removes the need for pre-processing multiple synchronized source images to generate a textured 3D geometric form, thereby addressing common issues relating to computational load, and mis-alignment of form and texture. This delivers an efficient, scalable approach to real-time 3D human reconstruction without the need for external large scale computational processing.

This is a third, more refined iteration of the development of process that builds on understanding developed in Chapter 3. Underpinning this is evolved understanding of the principles of actually doing and measuring 3D reconstruction with further enhancements to this understanding derived by the work in the section.

The theoretical and implementation considerations of this approach are considered, with experiments to measure its performance. These measures assess the approaches viability in terms of magnitude of source data and computational load, against resultant performance, to establish understanding *of scalability with graphical processing power. This evaluation demonstrates the matching of our approach, to current research, and demonstrates the relationship between this approach and evolving GPU processing power.*

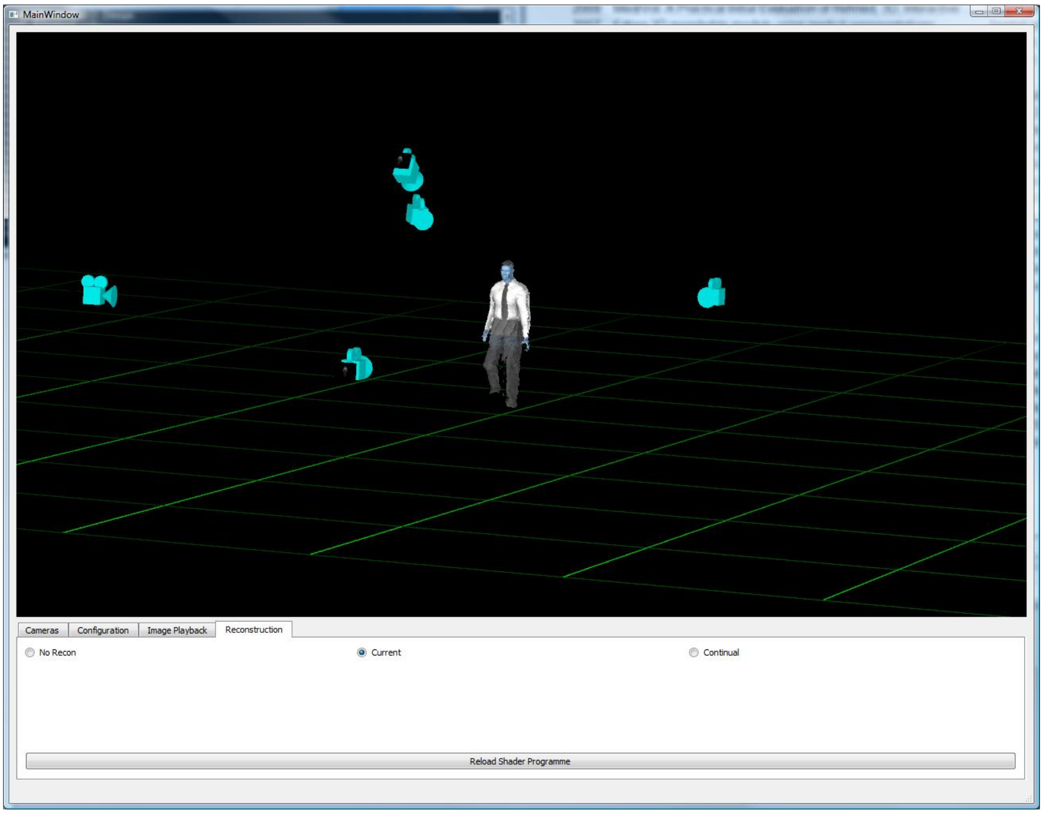
The conclusion of this chapter will review the evolution of the approach and align this with both the core research question and the context of the state of the art. The chapter summary will outline the progress made against the objectives of the research. A significant portion of this chapter has already been published in 'A GPU based, projective multi-texturing approach to reconstructing the 3D human form for application in tele-presence' [3]

4.1 Introduction

This chapter presents this approach and reviews evaluation of the performance and visual improvement of this with respect to our goal of matching the fidelity and performance of high end video conferencing. The previous generation reconstruction and rendering approaches utilized GPU accelerated features, primarily projective multi-texturing, in a 3D medical imaging inspired [4] algorithm, to enact render time space carving of a 3D shape from silhouette images [2]. The most recent development of this is to integrate the texturing of the 3D model surface, with the form generation process, to define a single step process enacted at real-time display rates.

The final (at this stage) iteration of the simulator extends this further, mapping the reconstruction process directly onto the Graphics Processing Unit (GPU) without the need for a CPU pre-processing step (Figure: 4.1). This additionally incorporates viewpoint independent texturing and divides the capture stage of the simulator from the rendering process in preparation for integration with live camera feeds from our array of networked video cameras. In this a method for volume based reconstruction is presented which eliminates the need for the formation of a volumetric model, and the resultant surfacing/texturing associated with rendering this. Instead we utilize existing graphics rendering features to populate a volumetric potential (a volume within the 3D space that has the source data projected into it, but has not been evaluated to determine occupancy) within the screen space of the rendering environment, through projective multi-texturing of the source data images. This removes the need for generation of a computationally, and memory, intensive voxel data set and formation of a geometric surface, whilst also reducing the generation of graphical artefacts coming from misaligned form and texture spaces. The approach applies a technique, inspired from 3D volumetric medical imaging. This creates orthogonally aligned slice planes to sample the volume potential, and applies a GLSL enabled rendering approach to evaluate and render each textured fragment without the need for surface forming or texturing. This view dependent sampling of a view independent volume potential thereby effectively utilizes the available resources of the GPU system to deliver good quality, fast 3D reconstruction, matching the computational performance metrics (reconstruction rates) of established approaches, in a way that can be argued is more scalable with evolving graphics hardware performance.

This is relevant because it illustrates that a bottleneck, that has for many years prevented 3D reconstruction from realizing this potential, has been addressed. Specifically it introduces a new variant of approach that for the first



○ No Recon ● Current ○ Continual

Figure 4.1: Third Generation Simulator: Both shape reconstruction and texturing on the GPU in one cycle

time allows 3D reconstruction algorithms to produce the required temporal and graphical properties to allow non-verbal communication to manage a conversation. It thus supports a representative, rather than severely truncated cross section of non-verbal communication. In doing so it opens the door to people being able to share each other's space with faithful representations of others. Combined with emerging capture and display techniques it could be used to allow people to exchange both a glance and a smile while walking across rooms joined as if through glass, or invite another to join a conversation in spaces shared between seemingly overlapping rooms. The analysis of performance and quality undertaken here is fundamental to understanding the suitability of this approach.

4.2 Requirements

The level of performance of the reconstruction and rendering process is particularly important if we are to reach our goal of matching the fidelity of state of art the video conferencing, such as Cisco Tele-presence, which updates at 30Hz in HD resolution. However, it is also important to consider the impact of tying the viewpoint to the observer's eyes, which is necessary if interpersonal gaze is to be accurately determined [71]. Motion sickness can be caused by latency in viewpoint update when the viewpoint is controlled from head tracking. No more than 40ms latency and at least 30Hz update is necessary to guard against motion sickness, with 50Hz being preferable (100Hz if we consider 3D stereo rendering). Nothing has yet approached a tele-immersive system with 30Hz update at anything supporting HD resolution. However, this is what is needed if it is to become a reliable form of communication.

In addition to the raw performance considerations, the reconstruction approach must be scalable both in terms of input data and reconstruction. The complexity of input data is in terms of size of image, number of images and the size and occupancy of the subject in the reconstruction volume. The complexity of reconstruction is in terms of spatial accuracy of reconstruction and calculation time. Therefore, a scalable solution would offer consistent performance regardless of the occupancy of the reconstruction volume and the format of the source data. Realistically, the number of input sources (cameras) will always have an effect, as each source stream requires processing, however, the relationship between calculation time and number of input sources should be no more than linear. A realistic goal for the output is a linear relationship between computation time and number of spatial samples (voxels) to be computed and in order to optimize the use of parallelism, there should be a direct relationship between processing power and computation time.

4.3 Theoretical Description

The inputs to this reconstruction system are multiple synchronized sequences of video streams from a set of cameras and camera calibration information. Streams from at least five cameras are needed to capture the back and sides of the head and to provide good representations of both left and right side of the face. The cameras also need to be spaced reasonably evenly around a person in order to reconstruct a believable human shape. The camera collaboration data is both intrinsic and extrinsic, denoting camera position and orientation and the lens characteristics of the capture system. Unlike the camera images that need to streamed continuously, the camera calibration data is only needed at start up, provided cameras are not moved. Each image must undergo some pre-processing to correct distortion and segment the reconstruction target from the background. This preparation of the input data is common throughout the domain of 3D reconstruction, and while processes for enacting this may differ the resultant datasets are broadly similar.

In conventional approaches, each image is projected into the reconstruction space, either as a segmented silhouette, or a profile curve derived from it. The geometric shell is then defined as the surface of the volume enclosed by the sum of each projected shape. This process is commonly referred to as space carving [42]. The resultant shell is tessellated to form a set of 3D vertices and indices denoting draw-able triangles, in some cases these are optimized into efficient, render-able geometries [58]. Further evaluation of the vertices maps texture data from the camera images to the form before the reconstruction is passed into a 3D rendering process for graphical display from any viewpoint [56]. In theory, if this can be achieved fast enough, a new geometric form and texture set can be displayed for each rendered frame thereby achieving the goal of real-time animated 3D reconstruction [1].

Conceptually this approach is different. A reconstruction volume is established which contains the space occupied by the intersecting set of projected image frustums from every source camera/projector. In order to render the reconstruction target within this volume, the space must be sampled at discrete intervals. During this process each evaluation point is tested for inclusion in the reconstruction target, determined by identifying if the sample point occurs within the projected silhouette of the reconstruction target for all of the projected images. If a sampled point is determined to be within the reconstruction target, a texture value evaluated by averaging the colour value for each projected source image pixel. Sampling could be achieved in a variety of ways. However, as this is enacted as part of the rendering process, only sample points occurring along the ray from viewpoint, through a rendered screen pixel and projected into the reconstruction volume need to be evaluated for the display of a rendered image. While this approach delivers a viewpoint dependent representation of the reconstruction target, this viewpoint is updated for each rendered frame giving similar rendered visual output to the previous approach based on a rendering of viewpoint independent 3D geometry.

In principle, this is much the same as the general approach taken in the Floating Textures [20] technique for applying multiple textured images to a pre-processed geometric form. However, the Floating Textures utilises this approach to minimise the visual artefacts generated by attempting to texture an ill fitting geometric form. In our case the potential for this problem is reduced as both the form and the texture application come from a single stage process in which both are considered against a consistent calibration set. This means that the texture projection is evaluated against a form calculated from the same source data and camera calibration set, thereby reducing the possibility of error.

The difference between the conventional approach and this approach can be considered as somewhat analogous to the difference between computer graphics global and local illumination models, with similar implications for the mapping of technique to the underlying graphics hardware. In global illumination models, as with traditional 3D reconstruction approaches, each element of the final result is dependent on mapping a complex interplay between all of the elements in the dataset. While each of these pipelines are inherently independent, each requires significant cross referencing of the entire dataset to achieve its goal. This leads to potential conflicts in memory access. Additionally, while parallelism can be achieved within each sub-process, each of the sub-process results must be evaluated and combined to a single coherent whole before passing to the next, thereby creating bottlenecks.

In local illumination, as with this approach, a single element is considered in isolation to all others throughout the entire pipeline. In this a single fragment is evaluated against only the eye position, camera properties, and lighting parameters and the result is displayed as a single pixel. Our approach is much the same.

In terms of mapping to hardware global illumination approaches impose non-uniform pipeline lengths, significant requirements for cross dataset access, and inherent bottlenecks in the overall process. For these reasons it is generally seen as enacted within the general purpose process facilities of CPU, main memory, etc., with any elements that can be optimized through parallelism farmed out to appropriate computational resources. This is common to most ray-tracing system implementations. In contrast local illumination independence of individual pixel calculations, coupled with fixed pipeline processing lengths for each fragment evaluation and minimal need for cross region memory access, lends itself extremely well to modern graphics hardware in which the GPU units comprise a relatively large parallel processing array. Each fragment evaluation can be considered as a batch job, which, given the fixed set of processing operations required, can be enacted in lock-step with all others through the parallel processing pipeline of the GPU. This concept of screen pixels as a processing unit, operating on many same instruction/data independent calculations, is extremely scalable. This is demonstrated by the current evolution of GPU based graphics hardware, where performance is being gained by an increase in number of GPU cores in the parallel array, rather than the actual speed of these individual units.

This aimed to develop an approach for vision based 3D reconstruction that maps directly to the underlying processing architecture of the graphics hardware and scales with the number of GPU cores it contains. The rationale for doing so was to make best use of commonly available graphics hardware and scale with the growth in number of cores of future GPU systems.

4.4 Approach

The second generation simulator enabled refinements in both reconstruction calculation time and rendering performance/quality, however a key element was missing; the texturing of the 3D reconstructed form. As no actual geometry for the form existed, a traditional approach of calculating polygonal texture coordinates, as part of the 3D reconstruction process could not be utilised. Instead projective texturing features available through the GLSL shader language were considered, with the intention of passing a second set of image data to the graphics card to be applied to the rendered form. However, it was quickly realised that this approach offered a significantly enhanced solution to the reconstruction process. Instead of using the CPU to cast the image into a 3D texture space, as a pre-calculated set occurring before rendering took place, this could be achieved at render time. Utilising the projective texturing features available within the GLSL fragment shader routine a 2D image can be projected onto a geometric surface with the projection properties defined by a local transformation matrix incorporating the spatial and perspective characteristics of the projection source. This feature is commonly used for creating the illusion of spot lights and reflection in polygonal rendering applications. In this case the textures relate to the cameras used to capture the original images. Their images are projected onto the view direction aligned planes derived from the MC slicing algorithm with texture coordinates calculated by the GLSL vertex shader and multiple images being combined within the fragment shader. Weighted addition, based on the proximity of the camera view direction to the viewers view direction, of the multiple image RGB values is used to combine the image and a multiplication of the alpha channel component used to enact the carving process for each fragment. Image weightings are used to empathise the texture detail from the camera most closely aligned to the viewers eye direction. This approach enables both the form of the 3D reconstruction to be generated and texture to be applied in a single step.

Further refinements were made to this version of the simulator. Texture coordinates were computed by the projective texturing process, and therefore no longer needed to relate to the parameters of the 3D volumetric space. This meant that the render time evaluation of a set of depth sorted polygons, derived from the intersection of orthogonally aligned planes with the volume region, to determine vertices and texture coordinates was no longer needed. Instead a set of parallel triangles, bill-boarded to remain aligned perpendicular to the viewers eye direction were used as surfaces to evaluate the projected image against, thereby removing the computational overhead of the MC slicing calculations.

Source data is captured form a separate application which allows the positioning of virtual models and cameras within a 3D reconstruction of the capture space. A sequence of images for each camera can then be captured and recorded, along with the intrinsic and extrinsic matrix data for each camera. This enables recording of standardized data sets that can be used, and reused, for texturing and evaluation. This approach allows us to quickly record many variants of our source data with differing characteristics, (eg number of cameras, source image size, etc). Most of our trials are based on the head model, correctly positioned and scaled to reflect the capture configuration we anticipate using in our physical system, which supports evaluation of the refinement of process with particular reference to the eyes and lips. However, any model can be captured and recorded, including animated sequences, allowing trials and tests with a wider variation of source types (Figure: 4.2).

This approach constructs a 3D form using an established feature of the graphics hardware, accessed through the use of the shader routines; projective multi-texturing [23]. In this approach camera images are uploaded to the graphics hardware, with the associated intrinsic and extrinsic matrices for each camera, and, for any drawn fragment compute the product of the

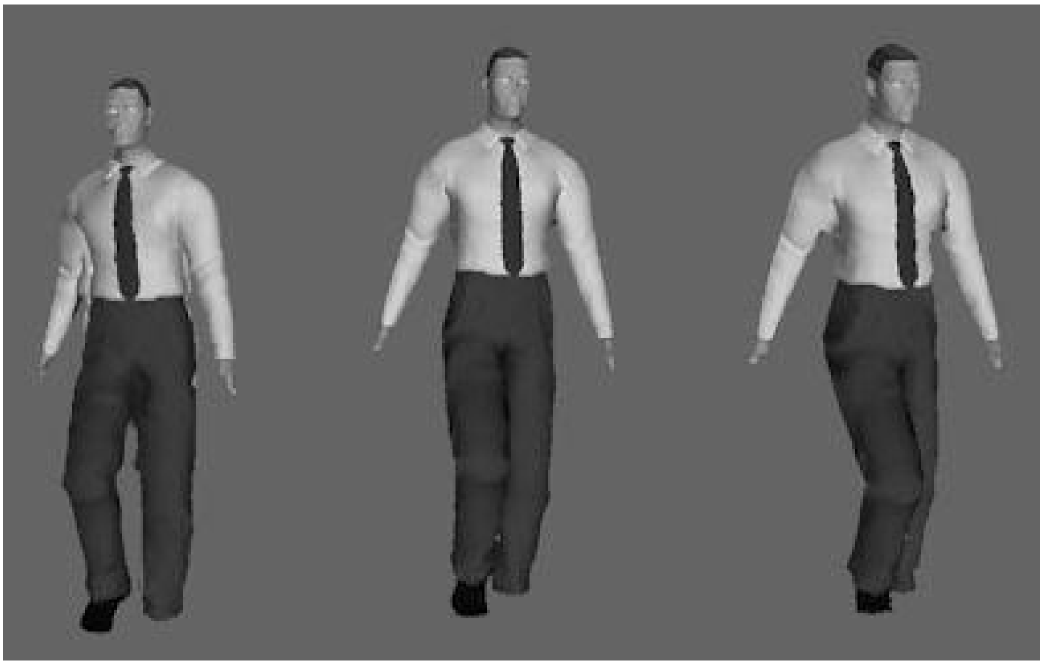


Figure 4.2: Walking model of a person, captured with 8 cameras in an inward facing horizontal ring formation. Reconstructed and rendered using GPU base projective multi-texturing.

projected texels alpha channel (segmentation data), for every image in the multi camera set, to determine the inclusion of a drawn fragment in the reconstructed form.

While this approach delivered a more visually refined 3D form, and significantly high update rates, the nature of the rendered form was still similar to our CPU based approach. A further refinement extends this by applying a blended composition of the camera images onto the 3D form during the render process, thereby texturing the surface. However, while 3D form reconstruction requires a product of every projected texture to be evaluated, a relatively simple operation, the texturing process requires a more subtle blending of the camera images closely aligned to the viewers gaze direction. Colour values for the surface elements of the 3D form are therefore computed as a weighted sum of the projected texel colour components, where the weighting value for each camera image is based on the dot product of the camera's view direction and the viewer's eye gaze direction. These are integrated into a vector and normalized to deliver a unit product vector. The weighting vector is computed for each rendered frame and uploaded as a parameter to the shader routines. The visual results of this approach (Figure: 4.3) offered a considerable improvement over the un-textured form.

4.5 Implementation

The reconstruction technique, described, has been implemented within the research simulation framework. This platform for experimentation is constructed using Trolltechs (Digita's) QT framework, for GUI [17], and Open

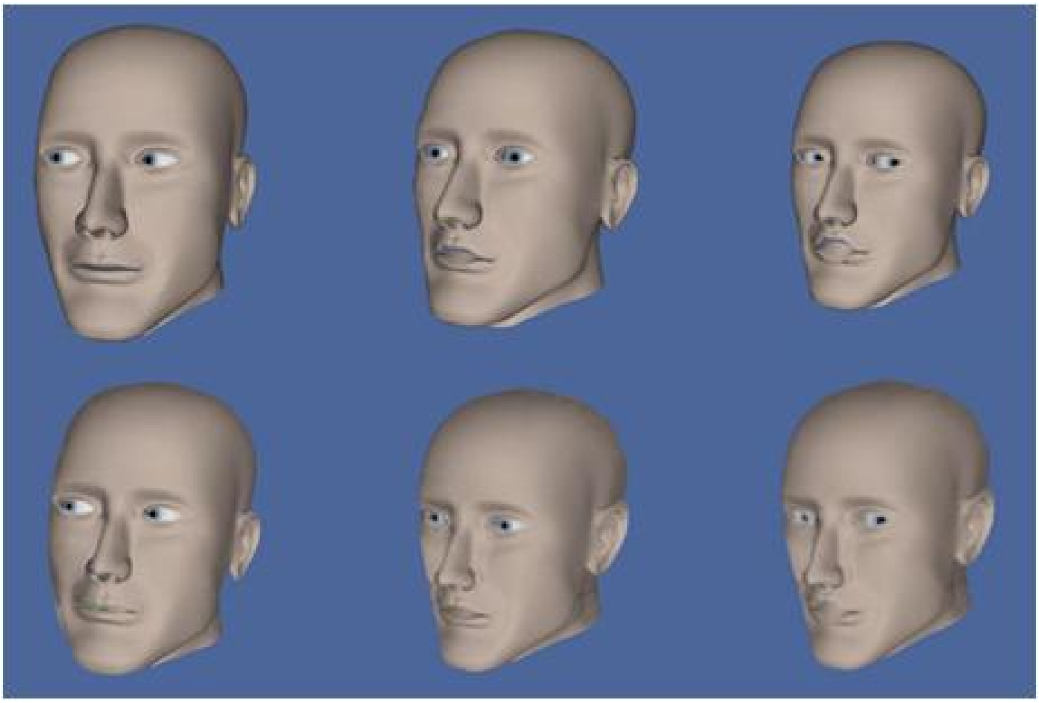


Figure 4.3: 3D reconstruction by projective texturing. Top row, source images taken from a ring of 8 cameras surrounding the animated (eyes & lips) head model. Bottom row: Reconstructed visual form (projective texture method)

Scene Graph (OSG) [67] for 3D rendering support. The entire application is C++ based, with functionality for virtual reconstruction of the physical capture and reconstruction environments; generation of synthetic data sets, from 3D geometric models surrounded by an array of virtual cameras; loading of live capture video sequences and camera calibration data; and evaluation tools for assessment of performance and reconstruction quality. Using this test application, reconstruction techniques can be implemented as plug-in components which can potentially be used within other OSG based applications. Overall the platform provides a stable and consistent toolkit for rapid prototyping and coherent evaluation, both of individual approaches and for comparative evaluation between techniques.

Within the plug-in created to implement this technique several stages comprising the process need to be created. Image and silhouette data, along with camera calibration data are loaded by the test platform, resulting in an array of Camera objects, each with unique intrinsic/extrinsic calibration properties and a list of the images comprising the individual video stream. The first action is to upload these images to the graphics hardware and define the properties for projection source location/orientation and the frustum of propagation. This established a populated reconstruction volume within the graphics hardwares local memory assets (Reconstruction Volume Definition). Next the sampling mechanism needs to be defined, thereby enabling the testing of despite points within the reconstruction volume (Sampling). Finally, the sample point evaluation process needs to be defined for determination of form and texture (Fragment Evaluation). These actions all take place before rendering starts and define the framework within which the reconstruction, and rendering, is performed as a parallel process in the GLSL fragment shader. At runtime only the image data, allowing the display of animated reconstruction, and observer viewpoint are changed.

Reconstruction Volume Definition

Image and silhouette data for each camera is pre-formatted, during the data acquisition phase, to a combined 16Bit per pixel format (R5G5B5A1) and passed to the graphics hardware as part of a multi-texture set. The transformation matrix for each camera position/orientation is extracted from the cameras extrinsic calibration properties and also passed to the graphics hardware as a GLSL shader attribute for inclusion in the evaluation of the sample points. The definition of the projection parameters (field of view, focal length, etc) are extracted from the intrinsic camera calibration and formatted into an OpenGL projection matrix for inclusion as a OpenGL texture attribute.

Both the facility for multi-texturing (OpenGL 1.2 onwards) and projective texturing (OpenGL 1.0 onwards) have long been included in the OpenGL standard and are available on any OpenGL compliant graphics card [23]. Fortunately, OSG offers a simplified mechanism for applying all these parameters from the osg::StateSet class, an instance of which is attached to the drawable geometry defined in the sampling process.

Through the use of projective multi-texturing, a core feature on the graphics hardware, we have, therefore, defined the populated reconstruction volume (Figure: 4.4). Animated sequences of video images can be uploaded within this construct to enable reconstruction of moving subjects to take

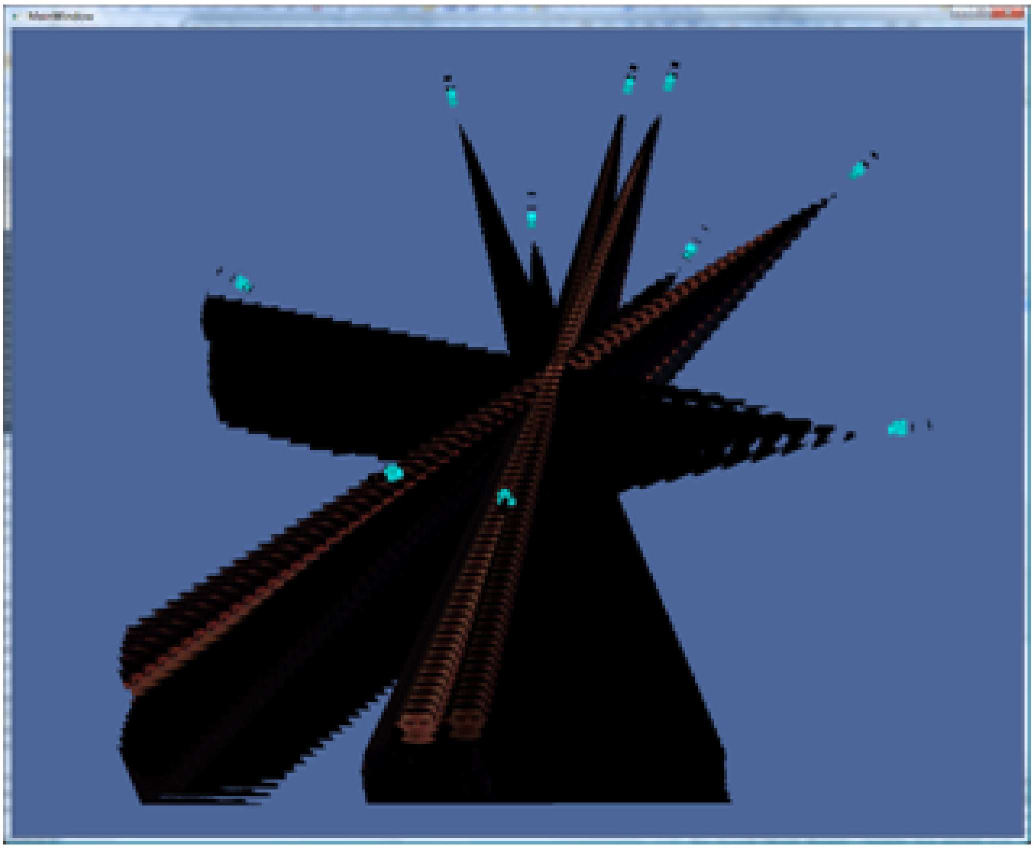


Figure 4.4: Formation of Reconstruction volume with intersected projection of camera images into reconstruction space from source camera positions.

place, with the only penalty being the upload time for each image in the sequence frame. While a 32 bpp image could be used, our trials show that there is little, if any visible difference to the reconstructed form, while use of a 16bit format halves the data bandwidth required.

Currently most OpenGL compliant graphics cards support 8 layers of multi-texture, restricting the technique to 8 camera streams, however, recent versions of the OpenGL standard are now defining support for up to 32 multi-texture layers.

4.5.1 Sampling

The populated reconstruction volume defines a volumetric space within the 3D environment where a 3D form can potentially exist. In order to test for this form a sampling process was adopted which discretely placed points within the volume can be located and evaluated. The traditional approach to sampling this sort of space would be to use a ray-casting technique. However, the usual enactment of this, as a CPU based process, fails to meet our requirement for a GPU based solution and since the introduction of shader based programming with both GLSL and HLSL, alternatives now exist which can utilize GPU processing power for ray-casting.

The basic unit for GPU shader based processing is either a vertex (vertex shader), or fragment (fragment shader). In the case of the latter a fragment is the sub-element of a drawn polygon that corresponds to a single screen pixel. As the drawn polygon exists in the 3D scene, a single fragment within a drawn polygon can be considered as a sample point within the 3D scene itself. By defining a fragment shader programme a function (or set of functions) can be defined to be evaluated for each of the fragments drawn within the polygon to which the shader programme is attached. These are effectively batch jobs, with location specific attributes which are operated on, in parallel, by the array of GPU cores.

The construction of a sampling framework for the reconstruction volume is, therefore, simply a case of determining what 3D geometric primitive to draw, and from this how to evaluate the fragments it generates. In the use of GPU processing for rendering 3D volumetric data sets two approaches are common. Drawing a single polygon orthogonally aligned to the screen, and drawing multiple orthogonally aligned polygons, at increasing screen depths, which slice through the volume. The former approach generates a single fragment per pixel, and allows a ray-casting mechanism to be enacted. In this case a ray is computed, originating at the viewpoint and passing through the fragment location. The fragment shader enacts a loop construct, which walks along the ray, with a set step size, and enacts the fragment evaluation function for each step (Algorithms 4.5.3 & 4.5.4). In the latter case, multiple fragments are generated for each pixel, each effectively corresponding to a single step in the ray casting method. In this latter case the polygonal geometry of the slice primates is commonly re-evaluated for each rendered frame so that polygons whose vertices match the boundary edge of the 3D dataset (usually a 3D texture) can be computed along with corresponding texture co-ordinates [8] (Algorithms 4.5.1 & 4.5.2).

In the evaluation both approaches have been implemented and tested (Appendix B.1.1, B.1.2, B.2.1, B.2.2). For our purposes a modified version of the slicing approach works best. The ray-casting approach generates less fragments for evaluation, however, each fragment defines a larger processing job and the need for a inner loop operator within the fragment shader, to enact the walking process appears to have a detrimental effect on performance. The slice planes method generates many more fragments, but these are much simpler in operation and have no requirement for a loop within the algorithm. In addition, because texture co-ordinates were evaluated based on the fragments spatial relationship to the set of projected textures, rather than mapping a 3D texture to a set of geometric vertices we have no need

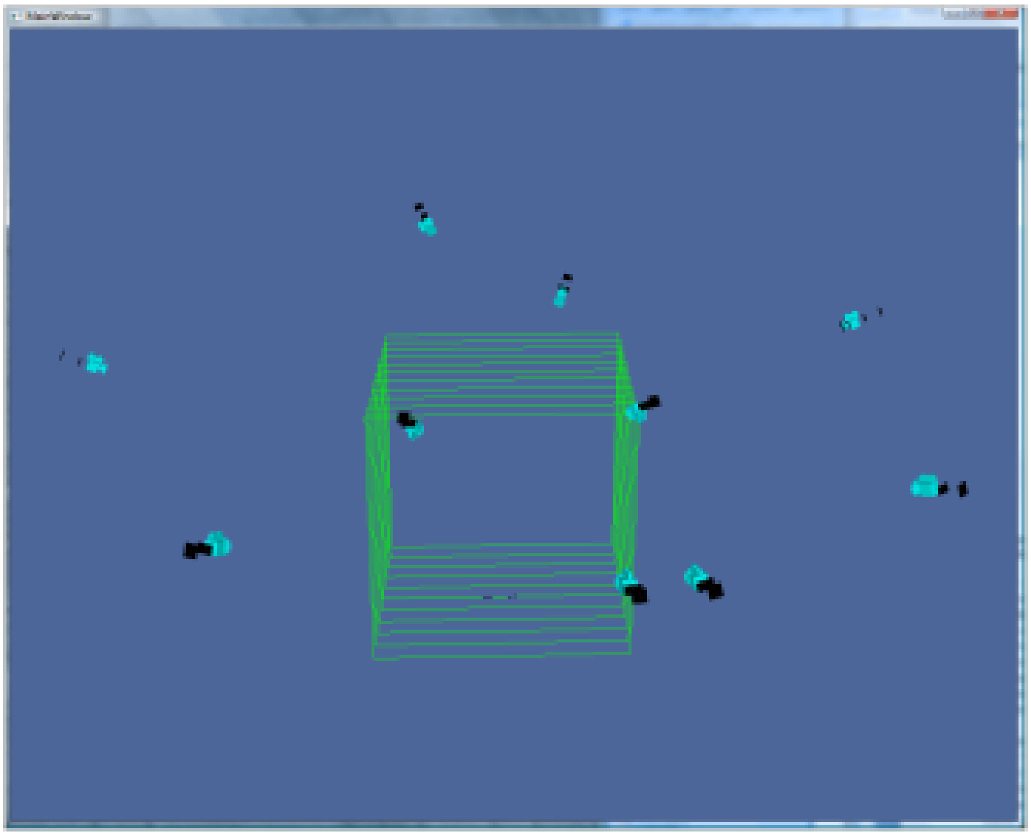


Figure 4.5: Definition of bill boarded slice planes used for sampling framework, camera positions also shown.

to modify the drawn 3D polygons as the viewpoint changes (Figure: 4.5). Therefore our drawn geometry is a set of rectangular polygons, which never change their shape, maintained in constant orientation to the viewpoint with the use of a simple billboard mechanism.

As the polygon fragments are effectively being used as sampling points for the 3D reconstruction volume, and the fragment colour value is solely dependent on the combination of projected texture components no illumination properties are required for these geometries and lighting is therefore disabled. Unlike 3D medical visualization, where the use of volume based rendering is required to display a semi-transparent 3D object, we are seeking to display a non-transparent form. Therefore blending functionality is also disabled.

Algorithm 4.5.1: THIRD GENERATION SIMULATOR: SLICE PLANE VERTEX SHADER ()

```
global $ViewMatrixInverse$
global $ModelViewMatrix$
main
  $Position \leftarrow ViewMatrixInverse * ModelViewMatrix * gl\_Vertex$
  comment: Resolve Texture Coordinates: for each camera image (n)
  $\{gl\_TexCoord[n] \leftarrow gl\_TextureMatrix[n] * Position$
  $gl\_Position \leftarrow$ FTRANSFORM()
```

Algorithm 4.5.2: THIRD GENERATION SIMULATOR: SLICE PLANE FRAGMENT SHADER()

```
global numCameras
global CameraImage[numCameras]; CameraPosition[numCameras]
global CameraWeights[numCameras]
global numSteps
global VertexPosition; EyePosition
procedure CHECKOCCUPANCY(position)
 for i ← 0 to numCameras
   do { alpha ← TEXTUREPROJECTION(CameraImage[i], position)
 return (alpha)

main
 comment: check is fragment inside form
 if CHECKOCCUPANCY(gl_Position)
   then
     comment: resolve camera weights to reflect eye position
     comment: and discard those at greater than 180 degrees to eye position
     for w ← 0 to numCameras
       do
         if CameraWeights[w] > 0
           then activeCameraWeights[numActiveWeights + +] ← w
         NORMALISE(activeCameraWeights)
     comment: evaluate fragment colour
     for f ← 0 to numActiveWeights
       do
         gl_FragColor ← CameraImage[activeCameraWeights[f]]∗
         CameraWeights[activeCameraWeights[f]]
         RETURN()
 DISCARD()
```

Algorithm 4.5.3: THIRD GENERATION SIMULATOR: RAY CAST VERTEX SHADER()

```
global $ViewMatrixInverse$
global $ModelViewMatrix$
global $VertexPosition$
global $EyePosition$

main
  comment: Ray castingis enacted from EyePosition through Vertex Position
  $VertexPosition \leftarrow ViewMatrixInverse * ModelViewMatrix * gl\_Vertex$
  $EyePosition \leftarrow ViewMatrixInverse * (0.0, 0.0, 0.0)$
  $gl\_Position \leftarrow$ FTRANSFORM()
```

Algorithm 4.5.4: Third Generation Simulator: Ray Cast Fragment Shader()

```
global $numCameras$
global $CameraImage[numCameras]; CameraPosition[numCameras]$
global $CameraWeights[numCameras]$
global $numSteps$
global $VertexPosition; EyePosition$
procedure checkOccupancy($position$)
 for $i \leftarrow 0$ to $numCameras$
   do {$alpha \leftarrow$ textureProjection($CameraImage[i], position$)
 return ($alpha$)

main
 comment: Setup ray to traverse along
 $rayDirection \leftarrow VertexPosition - EyePosition; rayDirection \leftarrow$ normalise($rayDirection$)
 $samplePosition \leftarrow gl\_Position; rayStep \leftarrow raaDirection/numSteps$
 comment: Traverse ray sampling at each step
 for $i \leftarrow 0$ to $numSteps$
   do
     comment: check is sample point inside form
     if checkOccupancy($samplePosition$)
       then
         comment: resolve camera weights to reflect eye position
         comment: and discard those at greater than 180 degrees to eye position
         for $w \leftarrow 0$ to $numCameras$
           do
             if $CameraWeights[w] > 0$
               then $activeCameraWeights[numActiveWeights++] \leftarrow w$
             normalise($activeCameraWeights$)
         comment: evaluate fragment colour
         for $f \leftarrow 0$ to $numActiveWeights$
           do
             $gl\_FragColor \leftarrow CameraImage[activeCameraWeights[f]]*$
             $CameraWeights[activeCameraWeights[f]]$
             return()
 discard()
```

4.5.2 Fragment Evaluation

Within the sampling process each sample point needs to be evaluated, both for inclusion in the 3D form and, if included, for what the evaluated pixel colour should be. In our technique this evaluation is performed by GLSL shader programmes. Each GLSL programme comprises two sub-programmes, one which operates on each drawn vertex (vertex shader) and the second operating on each rendered fragment (fragment shader). The OSG library provides a simple set of classes for defining and compiling these, with the resulting compiled shader programme object attached to the OSG geometry node defining the 3D slice planes. The OSG structure is also used to manage the additional attributes which need to be passed to the shader programmes for evaluation. These include each textures unique projection matrix and an array of camera weights used to determine the relative contribution each image make to the texturing of the 3D form.

Within the vertex shader, in addition to the required evaluation of the vertex position to screen space coordinates, a texture coordinate for each of the multi-texture elements is evaluated for the vertex as a product of the screen space vertex coordinate and the appropriate texture matrix. These texture coordinates are then interpolated for each drawn fragment to enable the fragment shader to extract the appropriate texel colour value for each of the projected images, relative to the fragment position. Within the fragment shader these colour values are tested. Each value comprises a 4 component RGBA vector, with the silhouette encoded into the Alpha channel (0: outside silhouette, !0: inside silhouette. Determination of whether or not a

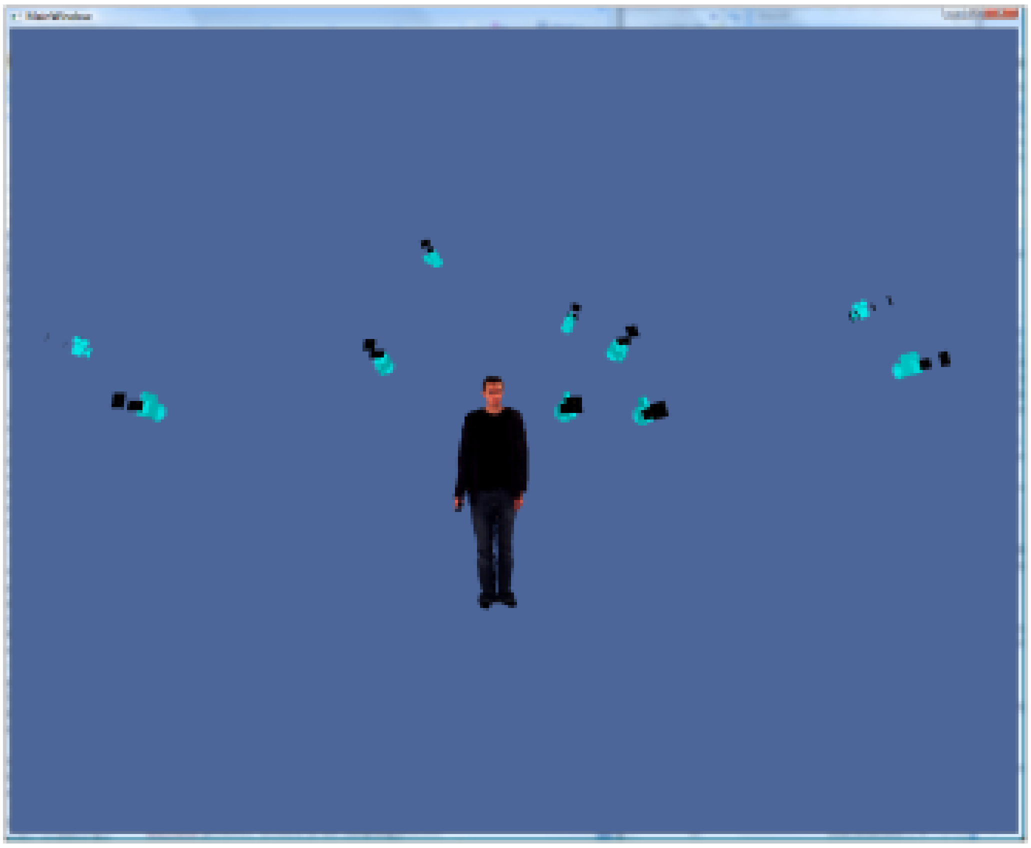


Figure 4.6: Evaluation of fragments within the sampling planes against the projected image set to form 3D reconstruction. Source cameras also shown.

fragment exists within the 3D reconstruction target is achieved by a simple multiplication of all the Alpha values from each of the projected textures (Algorithm 4.5.2 & 4.5.4: checkOccupancy). If this result is greater than 0, the fragment is within the projected silhouette from all of the camera images, and therefore within the 3D reconstruction form. In this case a second evaluation is performed to determine the fragment colour. This is determined as a sum of the product of each individual project image RGB value and camera weighting (Algorithm 4.5.2 & 4.5.4: main). The camera weighting is a normalized N dimensional vector (N being the number of cameras within the reconstruction system) of the dot product of each camera view direction against the viewpoint view direction. By performing this calculation each time the viewpoint is changed, and uploading the resultant vector as an attribute to the rendering system, a simple mechanism is defined to blend the texture contribution from each camera source for the current viewpoint, thereby ensuring that each fragment is generating its colour from the most relevant camera source. This delivers a smooth textured form without the visual ripping associated with polygonal processes that attempt to texture each polygon from a single texture (Figure: 4.6).

4.6 Evaluation

Evaluation of this 3rd generation approach to reconstruction was undertaken in two stages. In the first iteration a 3D geometric head model was used (Figure 4.3). This offered sufficient detail for evaluating the reconstruction of facial features and enabled rapid performance testing and tuning of the system (4.6.1). Once completed the simulator was adapted to enable import of real-world camera configurations and real-world capture data. This enabled evaluation experimentation that was representative of the real-world reconstruction problem (4.6.2). To enable comparison with initial experimentation results, which used simulated capture data, 3 datasets were chosen (Figure: 4.17). Two of these were real world captures of real people; the 'Salford' data set was captured using the oCtAVE system, while the 'Inria' dataset is publicly available and open source. The third is a simulated capture of a 3D human model, matching the basic characteristics of the others, captured within the simulation tools, using the camera configuration from the oCtAVE system. In both sets of evaluation the same Sun Ultra40 system was used.

4.6.1 Initial evaluation of performance

The final generation simulator presented a radical departure from the approaches of the first two. All of the reconstruction and rendering was enacted on the GPUs located within the graphics cards as part of an integrated process that both carved the reconstructed form and textured the resultant shape in a single rendering step. As this operated within the rendering pipeline the process was enacted each time the reconstructed form was rendered. From trials with the second generation simulator it had been seen that the raw rendering process, without pre-calculation of the voxel form, operated at approximately 100fps by evaluating a 3D texture, held in the graphics memory, against a set of 1000 orthogonally aligned polygons re-calculated for each render pass. Therefore, in evaluating this approach the key consideration is

No. Cameras	Mean Calculation Rate (fps)	Std Dev
1	59.2271	14.2228745
2	51.6702071	10.9989041
3	42.357363	7.3214645
4	34.107364	3.6429878
5	31.81052	4.1082862
6	30.02572	1.8843779
7	27.7103259	2.8068184
8	29.1641	1.621907

Table 4.1: Combined reconstruction and rendering time (frames per second)

whether the fully GPU enabled process improves significantly on the combined time for the calculation of the 3D volumetric texture, upload of the texture to the graphics card, and a single pass of the rendering process. Given that this approach also includes texturing of the reconstructed form we would also expect visual quality to be improved as well.

Using 6 to 8 cameras this approach delivers a stable 30fps rendering time (Figure: 4.7) when displaying the reconstructed form, of the same head model used in previous tests, when the camera images and projection texture matrices are held on the graphics card. When refreshing the camera images (1920^2 pixels @ 32 bit colour) and texture matrices for display of an animated sequence this drops to 27-29fps, excluding the time taken to read the subsequent animation frames from the computers hard drive. This suggests that this approach is significantly quicker than the previous direct volume rendering approaches, even when compared to the multiprocessor variant (Figure: 4.8).

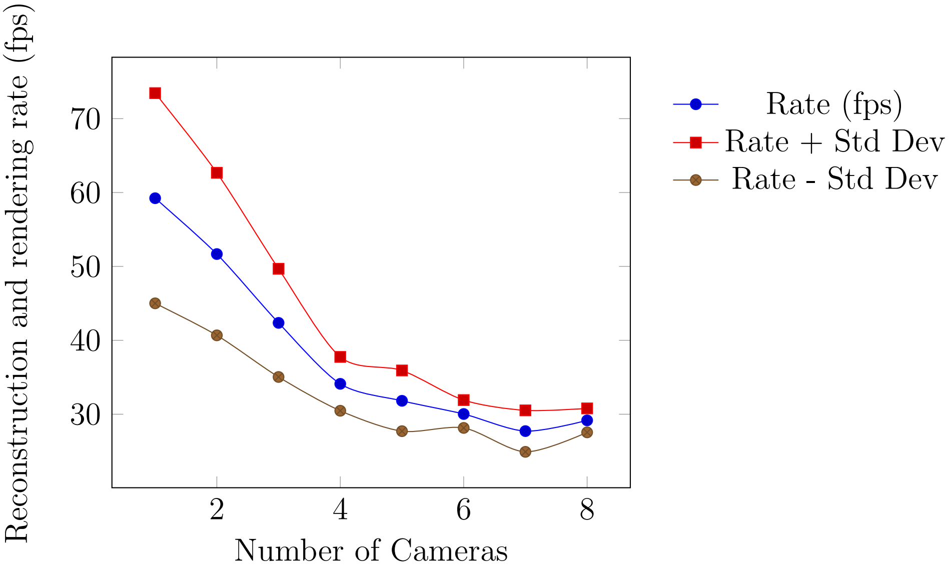


Figure 4.7: Plot of combined reconstruction and render times (Table:4.1), showing relationship overall performance and number of cameras (source images) used

Img Width (pix)	Img Height (pix)	Img Size (pix)	Rate (fps)	Std Dev
128	128	16,384	29.0234	2.5747
256	256	65,536	29.8295	4.9027
512	512	262,144	22.4298	3.6155
1,024	1,024	1,048,576	29.1546	3.7556
1,920	1,080	2,073,600	30.6337	5.54

Table 4.2: Combined reconstruction and rendering time considered against image size (6 cameras)

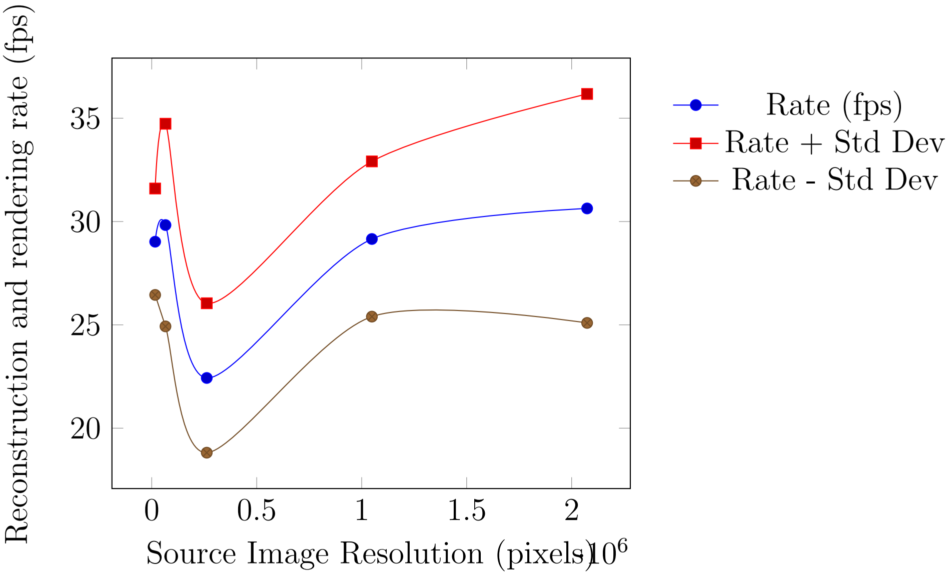


Figure 4.8: Plot of combined reconstruction and rendering time considered against image size (Table:4.2)

As with the CPU based approach, the size of the source images has little effect on the performance of the process (Figure: 4.8). However, given that this was performed for 6 cameras (images) in each case, and the worst case texture upload is @63Mb, this is a significant improvement on the @0.5Gb 3D texture required for the CPU based approach. While both approaches performance scales with texture magnitude in a similar manner, the projective texture approach simply requires less texture to be uploaded for each frame, and therefore imposes a lower overhead.

Occupancy is a more difficult measure to evaluate. Given the changes in the process, and the manner in which CPU based shading routines are applied, all of the shader routine operations are applied to every pixel drawn, the actual number of GPU operations does not vary if the drawn pixel is within the reconstructed form or not. Instead, a more meaningful occupancy test for this variation of the system is to consider the rendering and reconstruction performance against the proportion of the screen occupied by the reconstructed form. In this case performance was measured for differing sizes of rendered forms within the display screen. Interestingly, higher proportions of occupancy initially deliver better performance, possibly due to clipping of the evaluation planes reducing pixel draw operations. In addition a significant increase in performance appears to occur when the occupancy drops below 10% of the screen real-estate (Figure: 3.5). For a typical HD TV a head displayed at 1:1 scale occupies about 13% of the screen, while for our large scale display systems, with screens 2.2m x 2.7 this occupancy drops to @1% of the screen area. Within this range the current hardware platform performs adequately, however, the pixel fill rate of this system, the number of pixels that can be processed per second (pps) is only 12 billion pps. Current generations of consumer graphics cards offer in excess of 30 BPs, while professional level cards deliver fill rates measurable in giga pixels per second. These newer system should, therefore, offer more than adequate performance for both interactive display rates (rendering at more than 60 fps) and stereo rendering rates (120+fps).

While the transfer of the reconstruction process to operate entirely within the graphics cards processing ability has had an impact on the raw rendering performance, this is more than compensated for by the removal of the pre-calculation of the 3D volumetric texture in the second generation simulator approach. Offering a combined reconstruction and rendering time of approximately 0.033 seconds, compared to a best equivalent time of some 1.7 seconds (Figure: 3.7, six cameras, 256^3 voxels) on our test platform. In

% Occupancy of HD screen (1920x1080)	Rate (fps)	Std Dev
33.7	38.4742	4.3271
24.92	31.3147	0.6683
13.45	29.9711	1.3392
3.91	54.5121	3.9108
0.85	1,050.1665	300.7567

Table 4.3: Rendering performance (fps) for reconstruction and rendering process against proportion of screen occupied by the reconstructed form)

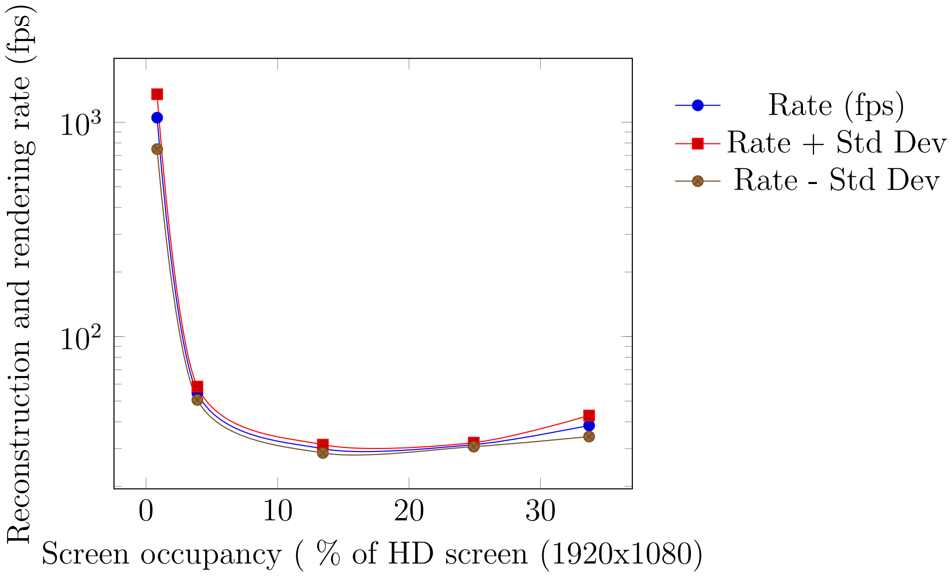


Figure 4.9: Combined reconstruction and rendering time considered against screen occupancy (Table:4.3)

addition, as well as including texture in this process, this approach, using the fragment shader, operates in screen pixel space thereby offering higher resolution to the reconstructed form than the 3D volume texture approach it is compared against (Figure: 4.9).

The improvements in visual fidelity, when a textured form was generated, were dramatic. However, the aim is not just to reconstruct and render a visual representation, but to do it in a real-time 3D environment, possibly with stereo rendering, as part of a collaborative immersive communication system.

These trials used a Sun Ultra 40 with 4x Dual-Core Opteron Processors, 8GB memory, and a single nVidia Quadro FX5600 graphics card. The simulator application was constructed using a Qt [17] framework, embedding an Open Scene Graph (OSG) [67] renderer with a bespoke volumetric rendering node and shader routines. Rendering refresh rates was limited to a maximum of 100fps. In all trials data was captured for the actual draw time of the reconstruction and rendering node, and for the overall frame rate of the application. Both measures were gathered as the overall frame rate would include additional overheads for the rendering process. Data was captured for 2 modes of operation; constant source images, ie the image data was retained on the graphics cards between frames, were used to evaluate the actual performance of the reconstruction and rendering algorithm, while refreshed source images, ie all of the source camera textures were uploaded for each rendered frame, were also used to determine the impact of the upload of the texture data on the overall process.

Our first test was to determine the effect that changing the number of

	Retained Source Image		Refreshed Source Image	
No. Cameras	Recon & Render time (s)	Application Frame Rate (fps)	Recon & Render time (s)	Application Frame Rate (fps)
1	$3.77 \cdot 10^{-4}$	98.23	0.1004	9.86
2	$2.09 \cdot 10^{-4}$	97.7	0.0788	12.58
3	$2.13 \cdot 10^{-4}$	77.87	0.0791	12.53
4	$2.09 \cdot 10^{-4}$	59.54	0.0792	12.52
5	$2.09 \cdot 10^{-4}$	51.96	0.0787	12.6
6	$2.63 \cdot 10^{-4}$	38.51	0.1008	9.82
7	$2.09 \cdot 10^{-4}$	37.42	0.0782	12.68
8	$2.86 \cdot 10^{-4}$	29.24	0.1004	9.86

Table 4.4: Combined reconstruction and rendering timings for differing numbers of source cameras (HD-1080p resolution). Measurements are made for both the actual reconstruction and rendering time and the resultant graphics application display frame rate. Variations are recorded for the raw reconstruction and rendering process, without texture upload to the Graphics Cards (Retained Source Images) and including upload of all the source textures for every rendered frame (Refreshed Source Images).

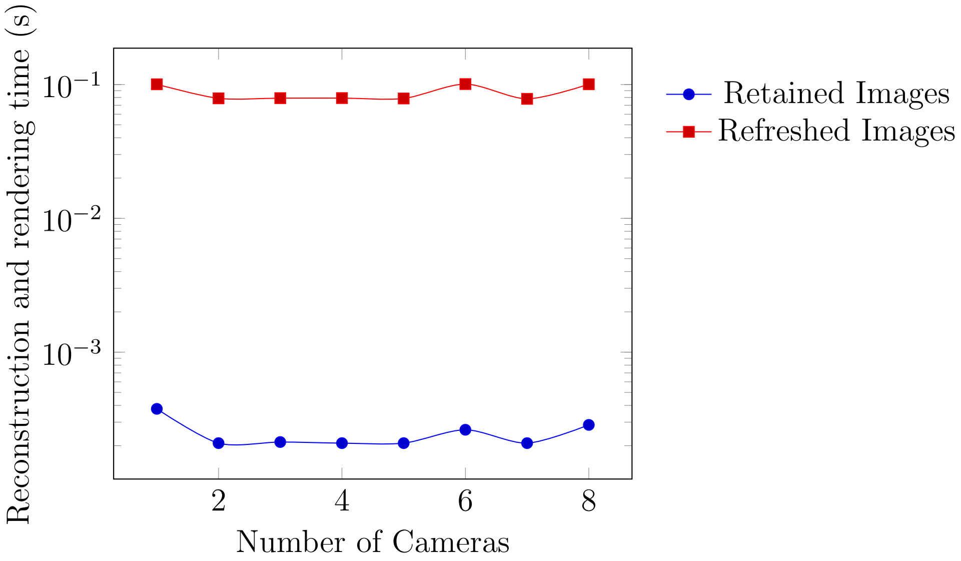


Figure 4.10: Impact on combined reconstruction and rendering time for differing numbers of camera source images.

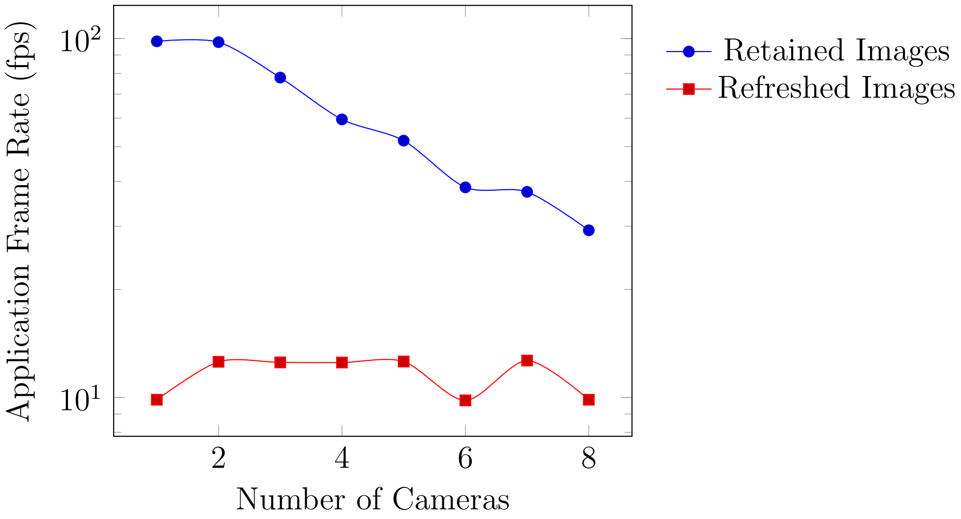


Figure 4.11: Application frame rates for combined reconstruction and rendering for differing numbers of camera sources.

source cameras had on the performance of the application (Figures: 4.7, 4.10, 4.11). In both modes the actual time taken for the reconstruction and rendering process remained relatively constant, all be it slightly slower for refreshed images opposed to the retained image mode, regardless of the number of source images used. However, when using retained images the application frame rate, strangely, progressively increased as the number of images reduced. This may suggest that when the images are retained on the graphics cards there is a significant overhead in managing the textures outside of the actual reconstruction object drawing process. When the images were refreshed for each frame the drop in frame rate suggest that the graphics cards texture bandwidth is limiting the overall application performance. However, it is strange that the application frame rate does not increase as the number of source images reduces. Further investigation is needed to determine why a relatively constant performance is recorded here.

In our second set of trials we considered the impact of image size on the performance of the system. In this case 8 source images were used for each test, with differing resolutions for each run (Figures: 4.8, 4.12, 4.13). As expected, the frame rate performance of the system increased as the texture sizes reduced. Curiously, when using the textures in a retained mode the actual draw time increased while the application frame rate got faster. However, given our target of 30 fps, to match current high end video conferencing, a source image size of 1024^2 pixels should give a close match to our required performance on our current, slightly old graphics hardware.

Our final consideration was how the size of the reconstructed form, proportional to the size of the rendering window, affected performance. This

% Screen Occupancy of reconstructed form	Retained Source Image Recon & Render time (s)	Retained Source Image Application Frame Rate (fps)	Refreshed Source Image Recon & Render time (s)	Refreshed Source Image Application Frame Rate (fps)
33	$2.23 \cdot 10^{-4}$	25.97	0.0797	12.44
24	$2.11 \cdot 10^{-4}$	28.07	0.0784	12.64
13	$2.14 \cdot 10^{-4}$	34.66	0.0785	12.63
4	$2.09 \cdot 10^{-4}$	51.1	0.0786	12.62
0.85	$3.62 \cdot 10^{-4}$	98.11	0.0785	12.62

Table 4.5: Combined reconstruction and rendering timings for differing sizes of source images (8 cameras used).

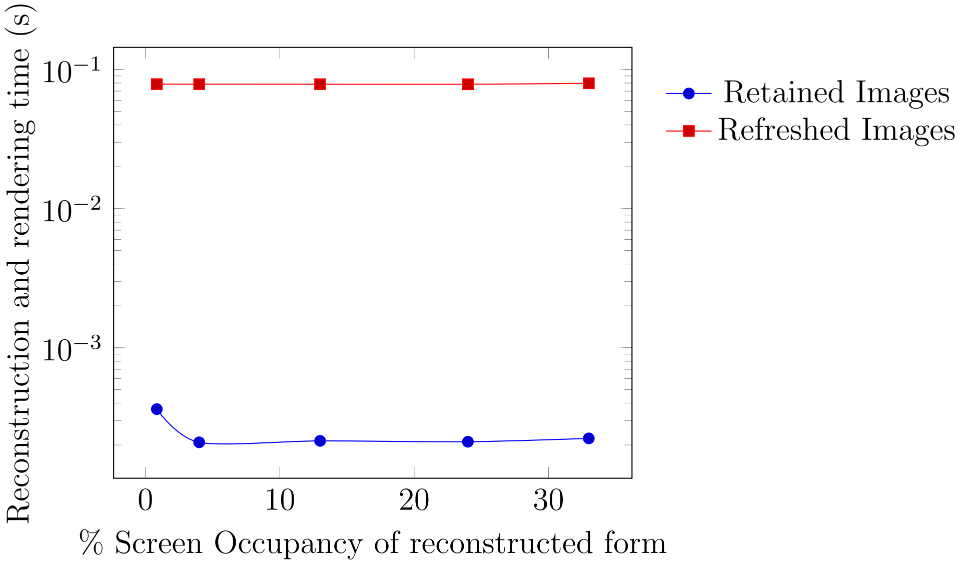


Figure 4.12: Impact of source image size on reconstruction and rendering time

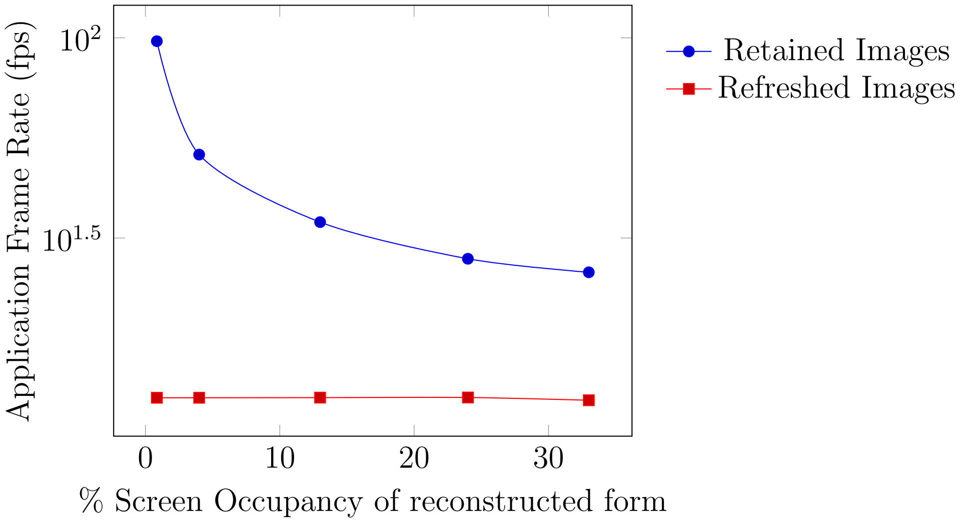


Figure 4.13: Impact of source images size on application frame rate

			Retained Source Image		Refreshed Source Image	
Source image width (pixels)	Source image height (pixels)	Source image size (pixels)	Recon & Render time (s)	Application Frame Rate (fps)	Recon & Render time (s)	Application Frame Rate (fps)
128	128	16,384	$4.09 \cdot 10^{-4}$	98.54	$2.431 \cdot 10^{-3}$	98.01
256	256	65,536	$3.56 \cdot 10^{-4}$	98.16	$5.153 \cdot 10^{-3}$	98.04
512	512	262,144	$4.03 \cdot 10^{-4}$	98.79	0.0143	65.55
1,024	1,024	1,048,576	$2.11 \cdot 10^{-4}$	49.15	0.0403	24.41
1,920	1,080	2,073,600	$2.86 \cdot 10^{-4}$	29.24	0.1004	9.86

Table 4.6: Combined reconstruction and rendering time considered against screen occupancy

occupancy measure is important as the shader processing operates for each screen pixel drawn against the rendered form. As the reconstructed object reduces in size it occupies a reducing number of screen pixels and therefore has a reduced GPU processing load. While the previous two trials considered constant screen occupancy, with the reconstructed form covering @24% of a HD (1080p) resolution rendering window, our final target systems have a differing set of characteristics. For a typical (60 inch) HD TV a head displayed at 1:1 scale occupies about 13% of the screen, while for our large scale display systems, with screens 2.2m x 2.7 this occupancy drops to @1% of the screen area. In this case we measured the performance of the system for differing occupancy levels to determine how this would scale to these displays (Figures: 4.9, 4.14, 4.15).

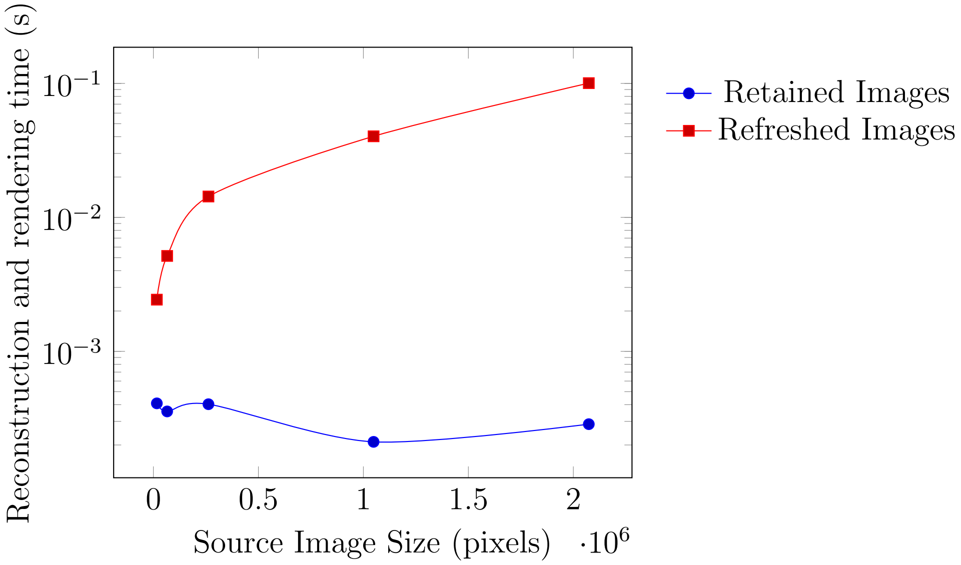


Figure 4.14: Combined reconstruction and rendering time considered against screen occupancy

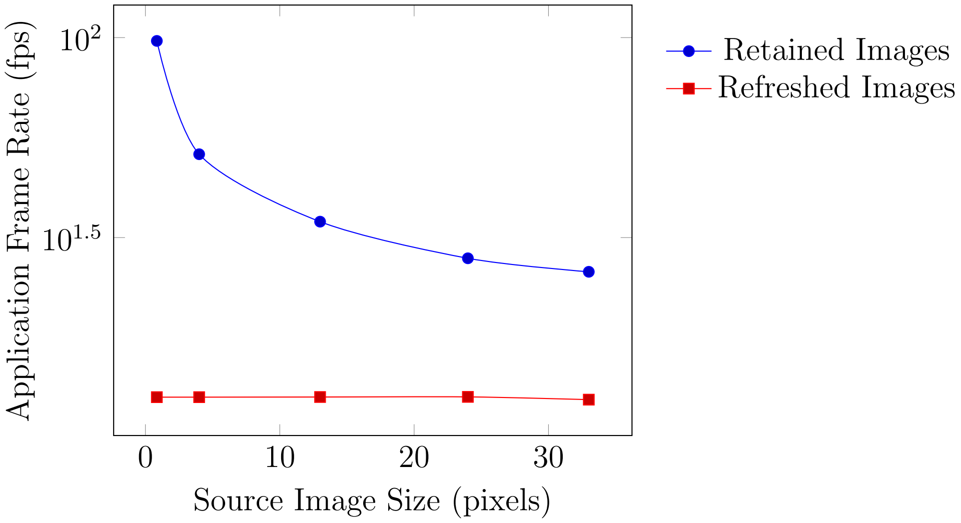


Figure 4.15: Frame rate of combined reconstruction and rendering considered against screen occupancy

For the HDTV display we have a raw reconstruction and rendering time just over the 30fps that we hope to achieve, with a significantly better performance at the large screen occupancy level. However, when the upload of images to the graphics cards are also considered we are still hampered by the texture bus bandwidth.

4.6.2 Evaluation of performance using real-world capture data compared with simulated capture data

In these trials datasets representative of the real-world problem for reconstructing a person were used. Three trials were undertaken to profile the performance of our technique. In the first two performance was compared for the two controllable parameters we have for the technique, the number of slice planes used to sample the volume space, and the number of cameras used to project source data into the volume.

In the final trial performance was assessed against the virtual distance between observer and reconstruction target. As this distance is increased the reconstruction target moves further back into the screen, hence occupying less of the screen space. This reduction in occupancy reduces the number of fragments that need to be evaluated for the reconstructed form (the form occupies less screen pixels as distance increases and within the GLSL fragment shader routine each execution results in a single screen pixel colour value), thus enabling a assessment of scalability.

In the analysis of the new technique the raw performance of the reconstruction and rendering technique is assessed. This this seeks to define and also attempts to profile the scalability of the approach to evolving graphics cards. For this 3 source data sets are used. The first comes from the simulated capture space and comprises 8 camera images, from 8 cameras located round the reconstruction target (Data set: Sim). Each image in this case was captured at a resolution of 512^2 pixels. The second data set comes from the Inria research centre 4D Repository [35], and comprises 8 synchronized video image sequences (with image resolution of 780x582 pixels) of 201 frames (Data set: Inria). The final source data comes from the oCtAVE physical capture space and comprises 10 camera images of resolution 1000^2 pixels (Dataset: Salford). For each dataset images were pre-processed to a 16bit format (5R5G5B1A) with the segmentation data encoded within the alpha channel. Image resolution was maintained for each set (Figure: 4.16).

Analysis of the performance of the technique was undertaken on one of the IPT system rendering computers (Sun Ultra 40, 2xAMD Dual Core Opteron 2210 processors, nVidia FX 5600 Quadro Graphics Card). This was linked to a rear projected display screen (2.6m wide and 1.95m high) running at a screen resolution of 1400x1050 pixels (104Hz refresh). The application view frustum (projection matrix) was defined to present a 1:1 scale representation of the virtual scene for an observer positioned centrally to the screen at a 1m viewing distance.

In each test the performance was measured as the resultant graphical frame rate delivered for 50 cycles of reconstruction and rendering including the upload of texture, to the graphics card, for each frame. In the case of the Sim and Salford datasets the full set of images was re-uploaded for each frame of reconstruction. In the case of the Inria dataset each frame was up-

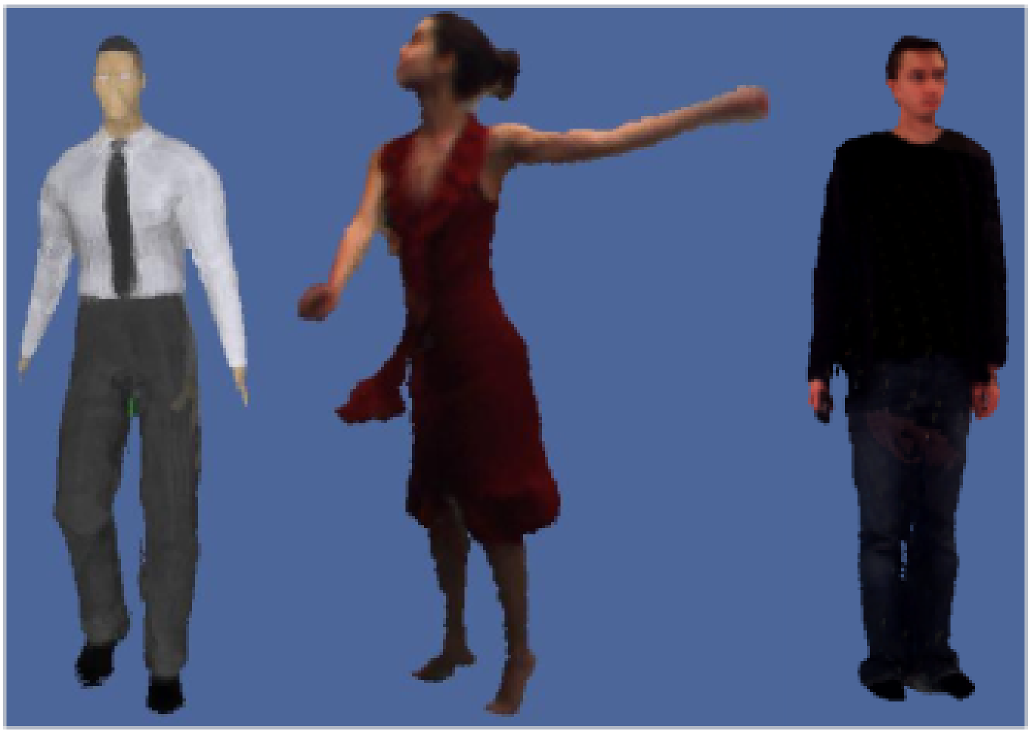


Figure 4.16: Reconstructions of the 3 datasets used for testing. Left: Reconstruction for simulated camera capture of a geometric model (Sim). Middle: Reconstruction from Inria Dancer data set (Inria). Right: reconstruction from the oCtAVE camera array (Salford). Each reconstruction performed with 8 camera sources and 50 slice planes.

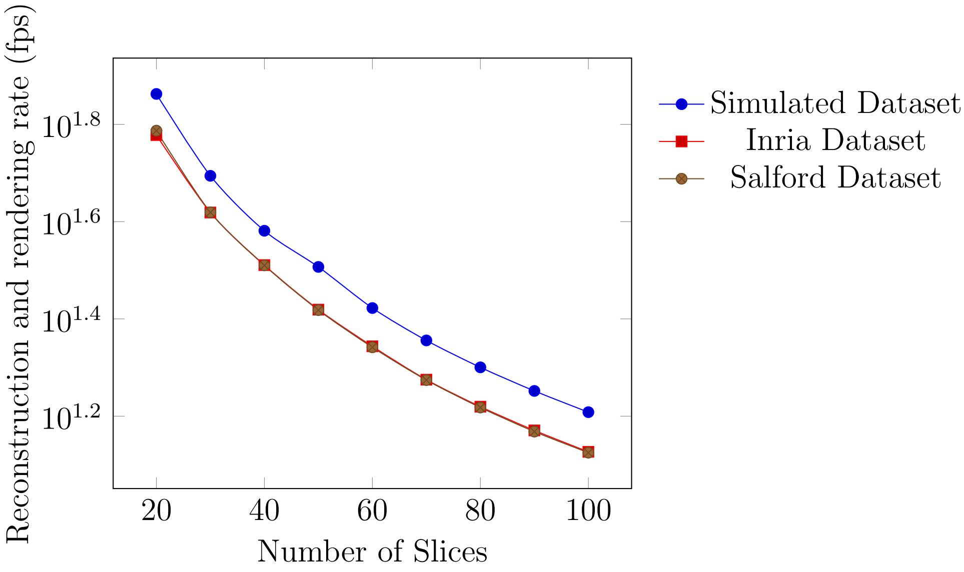


Figure 4.17: Plot of combined reconstruction and render times (Table: 4.6.2), showing relationship between overall performance and number of slice planes used (FX5600 system)

loaded, within a looped animation sequence refreshed for each reconstruction operation. Average frame rates are quoted (in frames per second (fps)) for 5 repetitions of the performance measure.

Evaluation of performance against sampling density (number of slice planes)

As expected, reducing the number of sample planes used in the evaluation of the reconstruction volume has a significant impact on the performance of the technique (Figure: 4.17). It is slightly surprising that this is not linear, as increasing the number of slice planes used is simply adding an additional number of independent fragments to be evaluated, this would suggest a $O(N)$ complexity problem, rather than the traditional view of volume rendering as

Number of slice planes used	Simulated Dataset (fps)	Inria Dataset (fps)	Salford Dataset (fps)
20	72.87304	59.97962	61.2408
30	49.45192	41.56344	41.60194
40	38.12996	32.40662	32.35962
50	32.1289	26.24566	26.18242
60	26.44288	22.06178	21.95912
70	22.6841	18.84178	18.8032
80	19.97182	16.57656	16.51274
90	17.86204	14.8153	14.73214
100	16.1526	13.38664	13.33304

Table 4.7: Combined reconstruction and rendering time (fps) for varying numbers of slice planes used for reconstruction and rendering process, evaluated on FX5600 system

being a $O(N^2)$ complexity problem. However, this may be explained by the potential overhead each slice incurs in terms of its rendering as a geometric primitive and the vertex operations required on that. The simulated data set does show a slightly better performance. This is most likely due to the smaller image size and the fact that these images are a power of 2 resolution (eg: 512^2 pixels) which often results in slightly better texture performance on OpenGL systems.

Using a technique, made possible with our simulation framework, to assess reconstruction fidelity (Chapter 5) it has been determined that 50-60 slice planes are sufficient for a 3D reconstruction of similar quality to that generated by a space carving technique. In this case the performance of this technique, approx 30 fps, is a good match for current high end teleconferencing systems such as Cisco Tele-presence, which updates at 30Hz in HD resolution.

Evaluation of performance against number of cameras used.

Reducing the number of camera sources used to project images into the reconstruction volume should increase reconstruction and rendering performance both because of a reduction in image data needing to be uploaded to the graphics card for each frame, and because each fragment process needs to evaluate less data (Figure: 4.18). This is not a linear relationship, although there is a relatively linear trend within the range 4-8 cameras. 8 cameras, and therefore 8 layers of projected multi-texture, is the maximum supported by our current graphics hardware, and with less than 4 cameras the resultant reconstruction is unrecognisable.

Number Cameras (source Images) used	Simulated Dataset (fps)	Inria Dataset (fps)	Salford Dataset (fps)
1	99.72264	99.7244	99.90856
2	96.93162	78.65694	86.3845
3	74.39144	59.84472	67.82484
4	59.5362	47.86868	52.30754
5	48.95234	39.66454	40.7077
6	42.8829	34.14832	33.78218
7	36.72068	29.47798	29.0644
8	32.1289	26.24566	26.18242

Table 4.8: Analysis of performance of the 3 test data sets with differing numbers of source cameras used, evaluated on FX5600 system with 50 slice planes

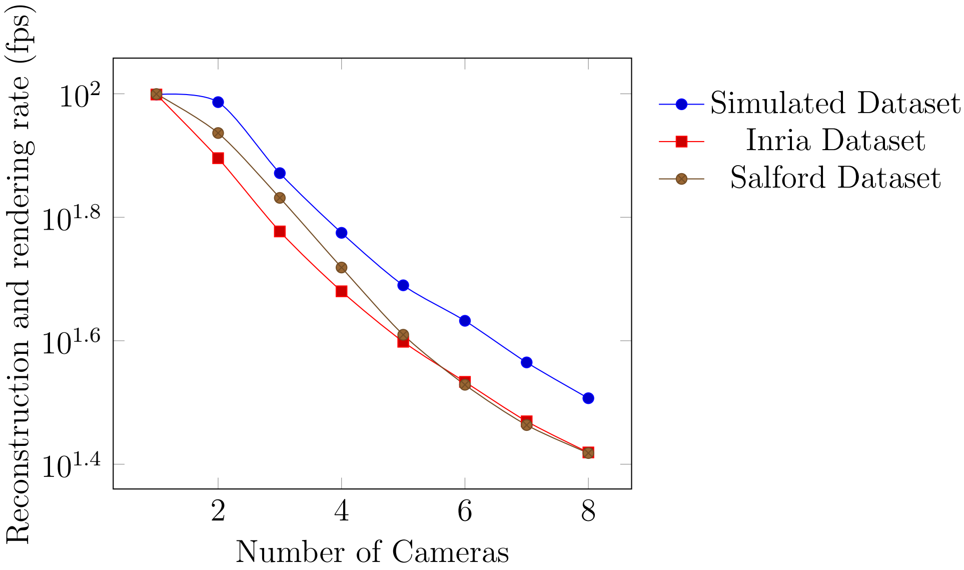


Figure 4.18: Plot of combined reconstruction and render times (Table: 4.6.2), showing relationship between overall performance and number of cameras (source images) used (FX5600 system, with 50 slice planes)

Evaluation of performance against virtual screen depth of the reconstruction target (scalability).

In this test assessing the potential for our approach to scale with evolutions in graphics hardware was assessed. As the distance between the observer and the reconstruction target is increased the proportion of the screen occupied by the reconstruction volume is reduced. This means that the number of screen pixels occupied by the volume is reduced and therefore the number of fragments that need to be processed is reduced. However, as each slice plane still needs to be evaluated, and each screen pixel defines the fragment within the slice polygon to be evaluated, this means that, in the case of this test, 50 slice planes will generate 50 fragments per pixel. Therefore, by evaluating the occupancy of the screen, using the known volume dimensions

		FX5600 System			8800M GTX System			580M GTX System		
occ	Estimated No. of fragments processed	Simulated Dataset (fps)	Inria Dataset (fps)	Salford Dataset (fps)	Simulated Dataset (fps)	Inria Dataset (fps)	Salford Dataset (fps)	Simulated Dataset (fps)	Inria Dataset (fps)	Salford Dataset (fps)
78.9	$5.8 \cdot 10^7$	9.4	7.7	8.1	5.6	4.4	4.7	23.2	19.1	20.1
52.6	$3.9 \cdot 10^7$	19.6	14.5	15.3	12.1	8.3	9.2	38.8	27.6	30.3
39.45	$2.9 \cdot 10^7$	32.1	26.2	26.2	19.8	15.4	16	46.6	38.1	38
31.56	$2.3 \cdot 10^7$	46.2	40.4	37.9	27.8	24	23.5	63.8	55.2	53.9
26.3	$1.9 \cdot 10^7$	54.8	55.1	50.5	36	33.9	31.4	82.1	77.4	70.9
22.54	$1.7 \cdot 10^7$	67.3	71.7	61.7	44.4	45.5	39.7	97.7	102	89
19.72	$1.4 \cdot 10^7$	81.1	87.6	74.6	53.8	56.6	48.1	118.4	128	108.8

Table 4.9: Comparative analysis of performance against estimated number of fragments, averaged for each of the test data sets on three different graphics devices.

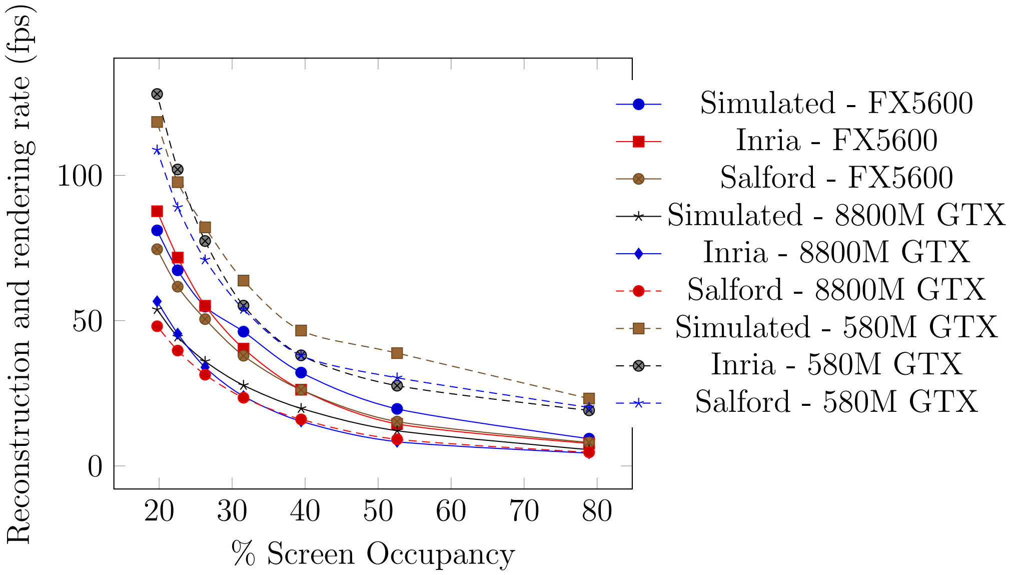


Figure 4.19: Comparative analysis of performance against estimated number of fragments, averaged for each of the test data sets on three different graphics devices. (Table: 4.6.2), showing relationship between overall performance and screen occupancy (50 slice planes)

and the screen projection matrix we can determine the approximate number of fragments evaluated at each screen depth (Figure: 4.19). Given that the same processing is required for each depth, in terms of geometry and texture, this is a fair measure of the actual processing load within the fragment shader pipeline. In this case performance figures quoted are the average performance for all three datasets.

Comparative performance figures are quoted for two additional computing systems; a DELL M1730 laptop equipped with a single nVidia 8800M GTX graphics card and an Alienware M18x with an nVidia 580M GTX graphics card. While the 8800M GTX graphics card is a similar generation to the test machine, the 580M GTX is significantly more advanced. Determining the performance differences of these systems is challenging. However, our reconstruction approach is predominantly comprised of pixel (fragment) operations and texture operations, therefore a comparison of quoted performance for these should give an estimate for the comparative performance of the approach on different graphics solutions, and therefore the scalability of the approach. The 8800M GTX has a quoted pixel fill rate of 8GP(ixels)/s and texture fill rate of 24GT(exels)/s. These compare with 19.8-39.8 for the 580M GTX and 14.4-38.4. This suggest the FX 5600 should be 1.8x faster than the 8800M GTX for pixel fill and 1.6 times faster for texture, with an overall performance 1.65x better. The 580M GTX with the 8800M GTX graphics cards can also be compared, and this suggests that the newer graphics hardware should be 2.5x faster for pixel fill, 1.65x faster for texture fill, and on average 1.86x faster than our baseline 8800M GTX system. If we are to demonstrate that this approach is scalable with the graphics hardware performance, similar differences in the performance of the approach on the three different systems should be expected.

Analysis of the results of this evaluation confirm the estimates for performance increase with graphics computational power (Figure: 4.19). Indeed, the comparative performance evaluations tend towards the pixel fill rate in both cases, suggesting that pixel fill rate is more significant as a measure of performance for this approach than the texture fill rate. Reviewing this data in a graphical form (Fig 4.19) highlights the scalability of the approach, showing that each performance profile curve is an upward shift across the entire range of test values.

4.7 Discussion

Attempting to solve the problem of adding texture to the original volumetric space carved, shape-from-silhouette, CPU based, approach that was naively created has taken the 3D reconstruction and rendering algorithms in a new direction. Considering the application of texture to the 3D voxel based form as a projection problem has not only improved the visual quality of our reconstruction, but also found a way to utilize the GPU processing hardware to deliver a significant performance increase over our previous approach. However, while visual quality and performance have been improved, neither gain is made without issue.

Adaptation of this approach, to operate on the graphics card's GPUs, initially to provide a textured reconstructed form, presented a significantly enhanced approach that utilised the GLSL fragment shaders functionality for projective texturing and logical processing to both carve the spatial form and apply texture to it in a single pass. Fundamentally, this removed the need to maintain massive memory structures for both the voxel form, and the relational linking between voxel data and image pixels, reducing the memory overhead to @63Mb, for HD resolution images, however, this memory requirement needs to be matched on both main system memory and the Graphics Card's own local memory. An additional refinement also weights the proportion of one cameras image contribution to the overall texture, by comparing camera view direction to the viewer's view direction further improves the image fidelity by providing a view dependent texturing process that enhances the visual representation further and may also reduce the need for a highly refined 3D form. While this approach gives adequate performance, within our test hardware system, extrapolation of the performance characteristics and extrapolation to current leading edge hardware suggests that the desired level of performance and fidelity is now achievable (Figure:4.19). Future developments seek to integrate the principles investigated here with the oCtAVE large scale display and capture facility to attempt a live capture and reconstruction of a person within an immersive virtual environment. However, before this adaption to a fully integrated system takes place, further investigation into the practical implications of real-life issues will need to be undertaken (Chapter 5).

In terms of visual quality the mapping of high resolution textures and their integration though a weighted blending process does illicit a significant improvement to the rendering of the 3D form, sufficient for fine features such as eye position and lip formation to be easily identified. However, one purely visual artefact remains. In weighting the images used for texturing the 3D form based on the matching of the view direction vector to the source camera direction vector a single dominate image contributes the majority of the texel contribution for any drawn pixel. Despite this visual artefacts are evident during the midpoint of the transition from one dominate texture to another (Figure: 4.20). While this is only a minor anomaly, the effect is disconcerting when seen. Increasing the number of cameras, thereby reducing the potential maximum angular camera separation against view direction has some mitigating effect, but ultimately better blending and composite image formation is required.

The target performance is to match current high-end video conferencing

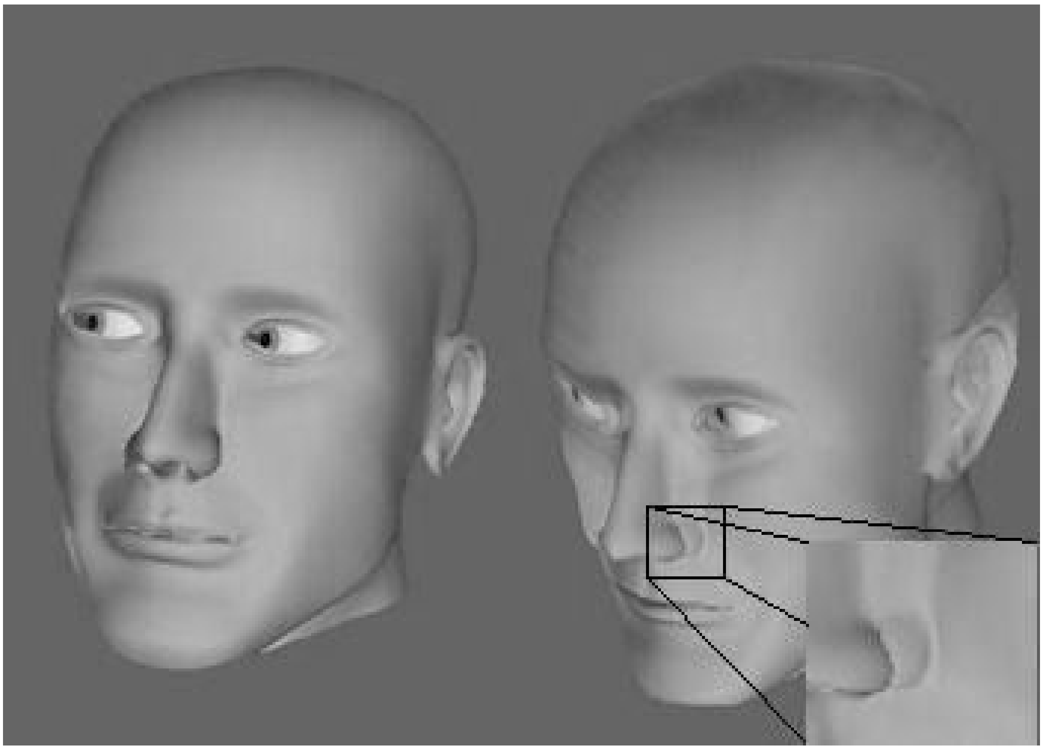


Figure 4.20: Shadow artefact during dominant texture transition. Left image is from a view direction closely aligned to a camera source direction. Right image is generated with view direction mid way between two camera directions (expanded view of shadow artefact)

(ie, HD-1080p resolution updated at 30 fps). While analysis, performed on current graphics hardware, shows that this has not quite achieved yet, the indications are positive. Using 8 cameras achieves close to the target 30fps, but only if without updating all of the images for every rendered frame. Reducing the number of source cameras used has little effect, as the primary issue is that the graphics hardwares texture bus bandwidth has been swamped by the image data (Figure: 4.6.1). Reducing the image size has a significant and dramatic effect on the application performance (Figure: 4.6.1). While reduced screen occupancy also improved the actual reconstruction and rendering time; when the upload of source images for each frame was included in the evaluation any performance increases were negated by the upload bandwidth limit (Figure: 4.6.1).

However, these results indicate several areas for performance improvement. Cutting a sub image of 1024^2 or even 512^2 pixels from the original source image should be relatively simple, although the intrinsic matrix for each camera will need to be updated accordingly. Our figures indicate that this could improve rendering performance to at least 24fps (Figure: 4.6.1).

All these tests assume that all of the images need to be updated for each rendered frame. Given that a typical 3D graphics system aims to render at 60+ fps to deliver smooth animation, but this captured source images are only captured at 30fps, this would also suggest that only half of the source images need to be updated per frame. While this would mean a slight lack of synchronization, research indicates that this would not be a problem [62]. This does not immediately suggest a performance gain (Table 1). Further reduction of the texture bandwidth could also be achieved by reducing the colour encoding of the image (currently the system is using a 32bit per pixel BGRA format) to a format closer to that of MPEG encoded video.

Ultimately, the most significant performance gain may well come from the ever increasing computational power of graphics hardware. The current graphics card for this evaluation is a 3 year old nVidia Quadro FX5600, with 1.5 GB of memory and 76.8 Gb/s bus bandwidth. Current generation ultra high end graphics cards offer 4 times the memory and double the bus bandwidth. Extrapolating the figures from the evaluation indicates that this would be more than adequate.

While this approach would appear technically and visually viable, further research is required to move this from a simulated approach to a working prototype. Fundamentally the level of tolerance to the noise and inaccuracy of real world data, and practical considerations such as network bandwidth need to be considered. While the tolerance of this type of application to image synchronization, a network feature, has undergone initial study [62], this has not considered other potential sources of error, such as inaccuracies in camera characterization (intrinsic and extrinsic properties) and background segmentation artifacts. In addition, this approach requires the source images to be transmitted from the capture system to every rendering node for every update of the source data. Within a controlled and localized network this is entirely possible. However, collaborative communication systems rarely operate within a dedicated local network, and distributing this quantity of source data over longer range network infrastructures may be a significant challenge.

Evaluating this generation of approach using representative real-world data showed that in terms of raw performance the approach appears favourable to others. In volume based reconstruction, with tessellation of the form prior to rendering, Wu quotes 12fps [102], Chueng [13] and Tao [90] both quote better than 15fps. These publications are contemporary to the graphics hardware used in the evaluation tests conducted here, however, in most cases the quoted performance is only for the actual reconstruction process, not a combined reconstruction and rendering cycle. In this approach it is impossible to separate the two, as reconstruction takes place within the rendering pipeline. However, the result of our first two tests (Figures: 4.17 & 4.18) show the current approach is close to matching this even at high sampling density with a full set of 8 cameras.

Scalability is more difficult to define. From the first two tests it can be seen that reducing the processing requirements for the technique, by either reducing the sampling density (number of slice planes) or reducing the number of input images, elicits gains in performance (Figures: 4.17 & 4.18). These factors have relatively little impact on the CPU within the system, but dramatically reduce the amount of processing required by the GPU. This confirms that the vast majority of processing required to enact our technique is within the graphics hardware.

In the third test the same level of input data to the reconstruction technique was maintained, but reduce the amount of processing required by reducing the number of fragments that need to be evaluated. This confirms that either reducing the processing load, which would deliver a reduction in quality, or increasing the throughput of fragment processing will deliver faster reconstruction.

The hardware used to perform the tests comes from the current oCtAVE display system. Within this, the graphics hardware is a single nVidia Quadro FX 5600 graphics card with 128 GPU cores running at a clock speed of 600MHz. This has been compared with the performance of this system with that of a more modest nVidia 8800M GTX based system, and a more advanced nVidia 580M GTX system. In theory the FX5600 should be some 1.65x faster in processing our reconstruction fragments, than the 8800M GX and the 580M GTX should be 1.86x faster. This is reflected in the comparative trials undertaken (Figure: 4.19) confirming that the approach is highly scalable with graphics hardware processing power while having little dependence on the significantly different CPU provision in each system. It is very difficult to compare performance between different graphics hardware as manufacturers rarely publish definitive performance figures, and 3rd party reviews commonly use highly subjective tests based on specific application types. However, current generation high-end graphics hardware (nVidia Quadro 6000) has pixel and texture fill rates of approximately 27GP/s and 32GT/s respectively. Without considering any other improvements to the main CPU system, or the graphics card architecture, this would suggest a theoretical increase in throughput of fragment processing of 3.3 times. While there is not a linear relationship between number of fragments and performance of the technique an expected increase of some 2-3 times performance with current hardware is anticipated. Given the current performance of the technique (@30fps, using 8 camera sources and 50 slice planes) this would suggest frame rates of 60-90 fps. This is more than adequate for mono-scopic display of a 3D reconstructed form, matching current high-end tele-conferencing systems, and is close to the rate required to support stereo-scopic rendering required for fully immersive environments.

An alternative to pursing raw performance could be to utilise this increased processing power to add functionality, which would improve the quality of the reconnection. Current evaluations suggest that 50-100 slice planes are adequate to present a similar quality of reconstruction to that produced by a space carving technique; however, increasing the number of slice planes increases the sampling density resulting in a refinement of the visual quality of the generated reconstruction. Based on the tests and analysis of performance, newer graphics cards could enable similar performance with 70-80 slice planes. However, a better use of extra processing power could be to refine the evaluation of pixel colour for each fragment.

The current approach uses a simple weighted merging of the source textures, with the weighting based on the proximity of the viewpoint to the source cameras. This delivers a smooth, coherent texture across the entire form, without the visual rips and tears common to many geometry based approaches. However, the approach does not consider occlusion of a source camera, such as when a body fragment is blocked from direct line of sight to a camera by, for example, an arm. In this case a visual bleeding of the arm texture to the body is noticeable (Figure: 4.21). Testing for occlusions can be enacted relatively simply by performing a ray-cast operation. In this the lines of sight between each included fragment and source camera can be walked and sampled at discrete intervals. Each sample point is evaluated for inclusion in the 3D form, and if identified to be within the form, the texture contribution for the particular camera being tested is discarded

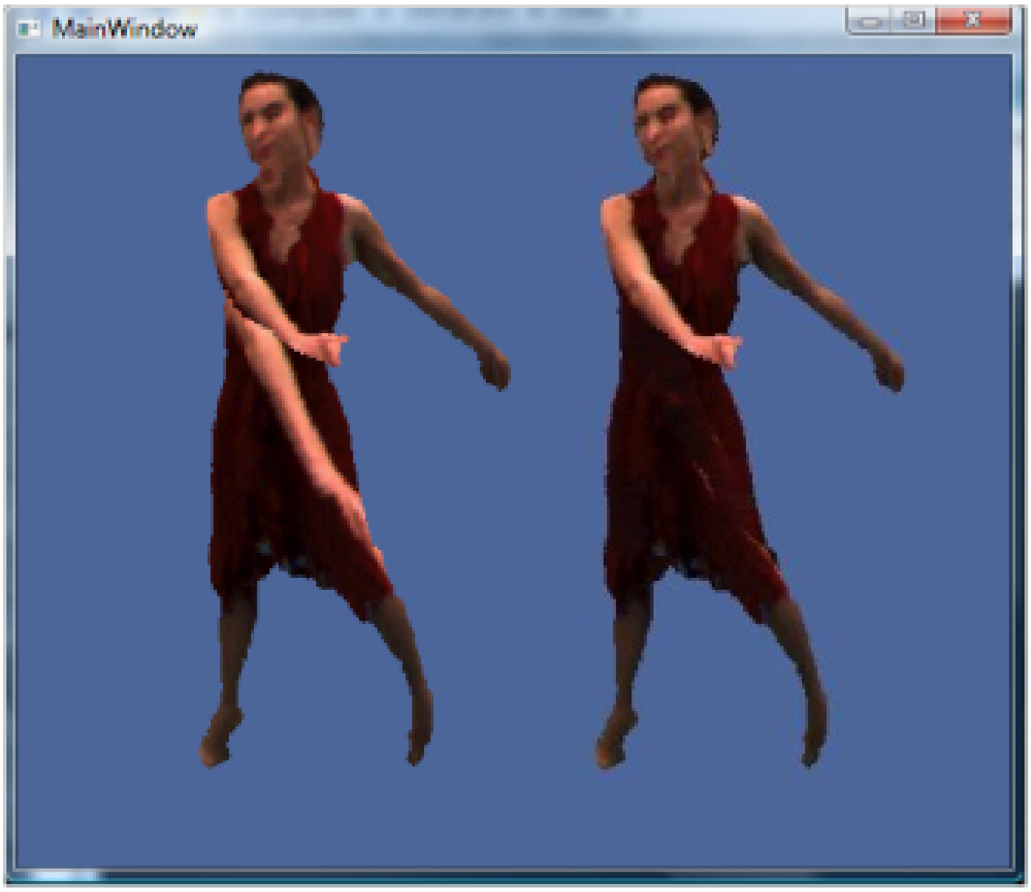


Figure 4.21: 3D reconstruction without occlusion testing (left) and with occlusion texturing, enacted by ray-casting (right)

(Figure: 4.21) (Appendix B.1.3). This is effective in removing the visual aberrations caused by the nave approach, but has a significant performance penalty. This is essentially similar to the approach taken by Orman [66] who uses the Bresenham Line algorithm to walk through the voxels within a volumetric set, however, in this approach the sampling process is enacted for each discrete point instead. Currently it is uncertain whether the impact of this additional computational overhead can be reduced with more recent, and more powerful, graphics hardware. However, regardless of whether raw processing power can help here, a more efficient approach needs to be sought and this should be the focus of future research, along with other refinements in image quality.

One significant issue also exists outside of the GPU based part of the algorithm. In order to define the sampling space, the volume for the potential 3D reconstruction needs to be estimated. To ensure efficiency, and reduce the need to process additional fragments that do not comprise the 3D reconstruction, this needs to be the minimal bounding volume for the reconstructed form. The current datasets define one static form (Simulated), one in which the person's position is static, only moving the head and upper torso (Salford), and finally one in which the reconstruction target is a dancer moving within a restricted space (Inria). In these cases a single specific volume has been manually estimated and pre-set for each test run. Obviously, this is effective for the trials we have performed, but will not be satisfactory for more general purpose 3D reconstruction where the reconstruction target may move freely within the capture space. However, this is not a significant problem. Volume estimation already needs to be performed by traditional volumetric reconstruction approaches [56]. Potentially this can be enacted as a CPU based process, as while the approach is heavily GPU intensive, the CPU is doing very little during rendering and reconstruction (eg, on the Dell M1730 laptop used for comparative trials, reconstruction and rendering at @30fps, utilized less than 30% of the available performance of either CPU core). This could be either as a pre-calculation to rendering a whole dataset, or, if the approach is sufficiently computationally inexpensive, prior to the reconstruction and rendering of each frame of the reconstruction.

4.8 Conclusion

This research applies a novel approach to generating 3D reconstructions of humans by enacting the population of a viewpoint independent volume, with a viewpoint dependent sampling mechanism, entirely within the highly optimized GPU processing pipeline. This removes the need for pre-processing operations, normally enacted on the CPU, or a cluster of CPUs, to resolve a volumetric model and formulate a 3D geometric form through computationally expensive processes of surface forming, tessellation and texturing. Moving from the previous CPU based, massively brutal, space carving approach to a more refined GPU based approach, utilizing hardware supported projective multi-texturing features, has enabled a refined texturing of the reconstructed 3D form, delivering significant gains in both performance and visual quality. Visually this is a significant increase on previous iterations. The inclusion of surface texture on the reconstructed form, while exhibiting a few minor artefacts, has enabled features such as eye direction and lip formation to be easily seen. This means that the inclusion of this approach within a tele-present communication system should enable spatial alignment of local and remote spaces with relative ease.

In terms of performance, while close to the current aim of matching high end video conferencing in terms of both resolution and update rate more work is required. The evaluation of the performance characteristics suggests several approaches that can potentially deliver these gains, specifically subsampling the source images and matching of screen occupancy to a 1:1 scale rendition of the form. Surprisingly, reducing the number of source images for this approach has little effect, although this may become important once the other enhancements have been made and the curious nature of the performance figures have been more thoroughly investigated.

Performance of the technique has been evaluated with respect to the two significant factors that affect the approach; magnitude of image data to be uploaded to the graphics hardware per reconstruction, and the number of samples points used (controlled by the number of slice planes used), evaluated per reconstruction. In both cases, these evaluations show that the approach matches predictions of the performance characteristics, showing near linear relationships (within the parameters of 'normal' application; 4-8 cameras, 50-100 slice planes) in both cases. In addition, these evaluations also show that for a reasonable quality reconstruction, visually similar to that reported by others, the performance of the approach, on the graphics hardware used, is close to that reported for conventional techniques.

The third evaluation sought to determine if the goal of scalability had been achieved. This was assessed by comparing the relative performance differences in 3 graphics card's pixel and texture fill rates, the critical processes in the approach, to the relative differences in performance of the approach, using the same dataset on different graphics hardware. The relationship between the graphics hardware performance and the evaluated approach performance is remarkably consistent.

These results show that this is an efficiently implemented technique, that scales in a predictable manner as the computational load is increased, and which is comparable to the current state of the art. The approach achieves @28Hz for combined reconstruction and rendering, with significant potential for much higher frame rates with newer hardware.

4.9 Chapter Summary

This chapter describes a GPU based, projective texturing approach to reconstructing the human form, from multiple images, at a quality and frame rate close to high end video conferencing. This is a major conceptual jump in the development of an integrated reconstruction and rendering technique that approaches the aims of the research and demonstrates a potential candidate solution that will, with evaluation and analysis, address the requirements defined. Further refinement of the approaches for measuring quality and performance of the reconstruction technique, evolved through the work undertaken in this chapter, contribute to the performance characterisation of the approach. In evolving, and making a major conceptual leap in approach, for the integrated reconstruction and rendering technique, this has also demonstrated the potential of the simulation frame work in both understanding the conceptual problem and identifying issues within the reconstruction process. In effect the research reported by this chapter has combined the initial formative investigations from a naive standpoint (Chapter: 3) with state of the art context to deliver a platform for addressing the research questions which will be explored further in the following chapter.

Chapter 5

Validating the Approach

This chapter defines methods for objectively measuring the view synthesis quality of 3D video algorithms and rendering that isolates evaluation from the impact of error in the capture pipeline. Unlike previous methods that did not isolate the reconstruction process, this accounts for both texture and form. These evaluation methods utilise the simulation framework defined for this research to leverage an almost infinite number of viewpoints from which to recreate, texture and evaluate the reconstruction. To achieve this a CGI source model is captured through virtual cameras, independent of the reconstruction pipeline, to establish a 'truth' data set for comparative evaluation. For a case study this evaluation approach is used to compare the quality of form and appearance achieved by two sampling approaches within a common volumetric reconstruction method. While these approaches produce seemingly similar visual results, the novel analysis highlighted clear differences in quality not apparent to a human observer. This analysis helped to characterise the relative view synthesis quality of each sampling and 3D reconstruction

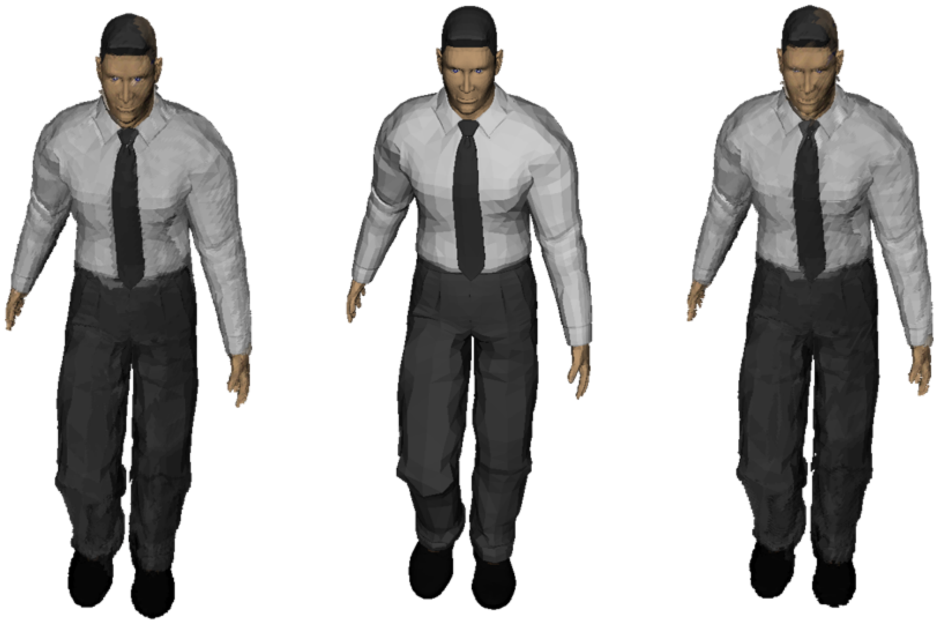


Figure 5.1: Spot the Difference: Image-based reconstruction of 3D geometric model (middle) using volumetric approach with sampling by slice planes (left) and ray-casting (right)

approach. This allows repeatable, quantifiable comparative critical evaluation of each, complementing subjective human interpretation which steers towards the ultimate outcome of faithful reconstruction. Ultimately this supported evidence based selection of approaches for application in real-time 3D graphical collaborative systems, but can additionally be used to objectively compare two different approaches within a common repeatable experimental situation.

5.1 Introduction

Video Based Reconstruction is becoming an important tool in TV production and the support of tele-presence. There are numerous approaches to VBR yet methods for comparing their visual quality have largely relied upon subjective impression of images (Figure: 5.1). Objective methods are beginning to emerge but these have not isolated the significant impact of arrangement and characteristics of cameras, and noise within the capture pipeline. As where the cameras are placed and how well they are calibrated typically makes more difference to quality than the algorithms themselves, this is a major shortcoming. Positioning and calibrating cameras is time consuming, hard to exactly replicate, and most achievable in lab conditions. Repeating the movement of sample objects or people is very hard. It is expensive and impractical to completely surround highly dynamic objects with cameras in a real world setting and to transmit huge numbers of images (video) across a network in real time. Thus, it is important to know how reducing the subset of possible cameras impacts on quality for a given reconstruction/display method. It is therefore useful to be able to measure the performance of the algorithm and rendering across various camera arrangements without measuring camera based error sources, such as calibration, and pipeline characteristics, such as quality of segmentation. The approach taken is to use a 3D geometric source model captured through virtual cameras. This is reconstructed from the dataset obtained by the virtual cameras and then evaluated by comparing the source and reconstructed object from multiple matched viewpoints that are independent of source cameras. Reconstruction from a virtual object means this is not constrained by practicalities of camera size, placement and environment. Additionally this approach is not effected by inherent camera and pipeline characteristics, such as noise and segmentation quality. In this simulated environment both set up and moving form are easily repeatable. Direct comparison of the pixels within each image pair, is used to assess localised image space differences in form and surface texture (appearance) of the graphical form. Resultant maps provide visualisation of quality, allowing assessment of variations around the entire reconstructed model, including view-point regions between source camera positions. Numerical analysis of the comparison results provide quantitative measures for evaluating evolving techniques.

This approach has several uses. In developing an approach test data from two iterations of an approaches development can be evaluated to determine whether the later iteration offers improvements in fidelity over it's predecessor. This complements traditional subjective testing, by user observation, by offering a finer granularity of testing that can review minor changes with slight improvement, and parameter optimization that might be used in the set-up process for subjective experimentation. Additionally, because the reconstruction algorithm is implemented within a plug-in component for the simulator system, two different approaches could be compared within the same experimental configuration. This would allow direct comparison of two different techniques.

In this chapter the objective analysis approach is used to evaluate two variations within a technique. Chapter 4 presented two variations of the same GPU aligned approach to reconstruction; slice plane and ray cast sampling (Algorithm 4.5.2 (Slice plane) & 4.5.4 (ray casting)). The variations are comparable in performance, with slice plane sampling offering a slight advantage, and, subjectively, visually similar. This objective analysis is used as a case study in which the better variation is selected for further development.

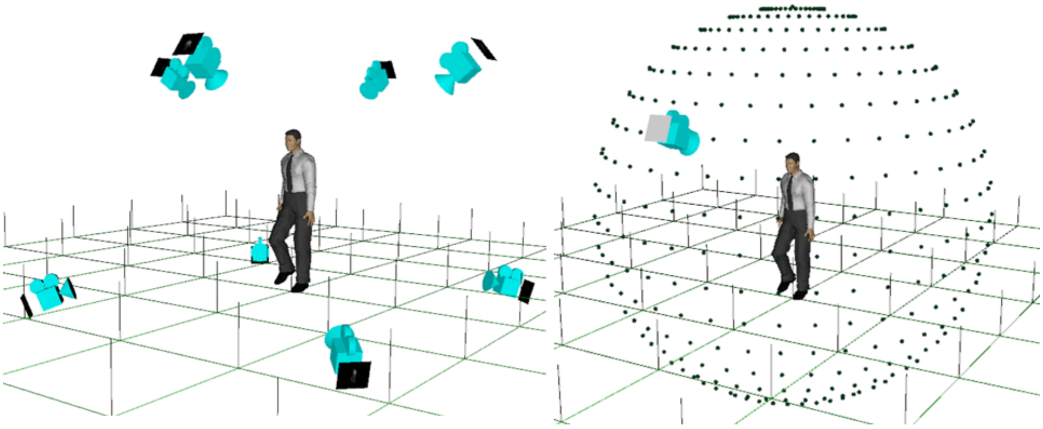


Figure 5.2: Simulated capture tool configurations. Two modes of capture shown. Left, capture of images from set camera positions used by reconstruction algorithms. Right, virtual camera capturing images of the geometric form from positions on a spherical surface surrounding the form, to be used in quantitative comparative analysis.

5.2 Approach

In developing approaches for real-time 3D reconstruction, this research started, not from an array of video feeds drawn from a 'live' capture system, but with a software simulator that allowed modelling of the proposed capture environment. This was created because at the time of starting the research the physical capture environment (the oCtAVE) was under construction. The use of a simulated environment allowed prototyping of possible configurations for that environment and enabled rapid prototyping of 3D reconstruction approaches. The simulator is an interactive 3D graphical application that allows positioning of virtual capture cameras within a simulated 3D space, where both the camera properties and the units of space match the real world environment. Within this space a textured and, optionally, animated 3D geometric model of a person can be loaded and captured as a synchronised sequences of images from any set of virtual cameras (Figure: 5.2, left). The position, orientation and imaging characteristics of the camera are set by the user, and stored as a fixed, mathematical configuration, while the images are generated as part of a graphical rendering process against a controlled background. This, therefore delivers both clean, segmented image and configuration data that accurately locates the capture source and the camera characteristics. These relate to the intrinsic and extrinsic matrices that would need to be computed for each camera, every time the system is re-configured. This approach also means that a truth data set is maintained; the original 3D model that camera images were captured against. This can be used as a baseline for comparison with the 3D reconstructed representation. This allows assessment of both the performance of the reconstruction algorithms and the quality of the reproduction as computed metrics.

These sequences, along with the camera properties can then be used as inputs to a real-time 3D reconstruction system to allow stable and repeatable trials to be performed. Extending this simulated capture system has allowed a feature impossible in the real-world; the capture of images of the source object, from a free-viewpoint camera, which are perfectly segmented and accurately located in the capture space (Figure: 5.2, right). These can be paired with images, taken from matching viewpoints around the reconstructed form, thereby allowing comparative evaluation of the entire reconstruction against a 'truth' data set, by means of evaluation of the matching image pairs (Figure: 5.3).

This analysis approach is to reconstruct an object from one, minimal, set of cameras and then compare the reconstruction to the source through a sec-

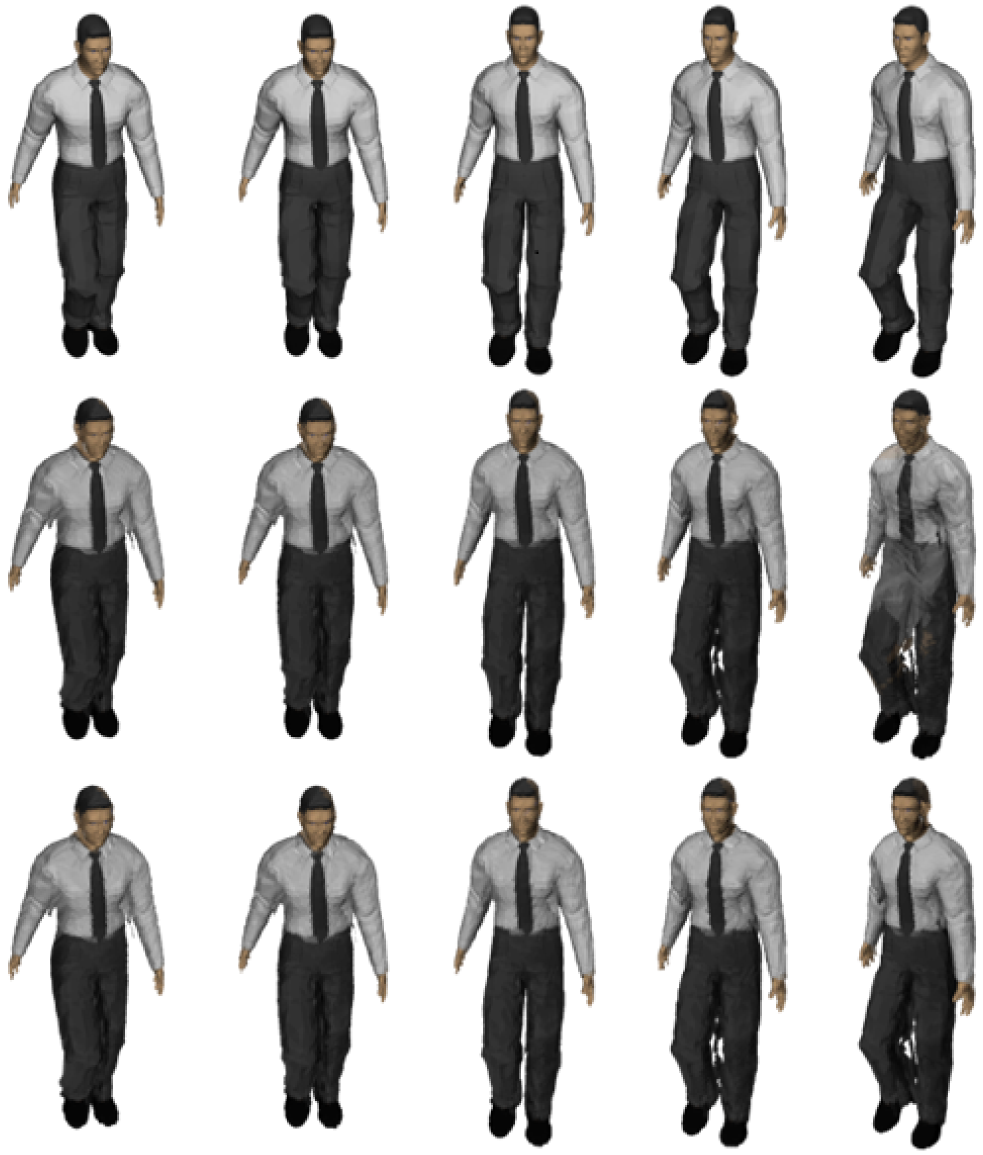


Figure 5.3: Evaluation images taken from virtual surround camera. Each column of images taken from the same viewpoint with identical camera characteristics. Top: Images of the source (geometric model). Middle: images generated using the slice plane sampling approach. Bottom: Images generated using a ray cast sampling approach.

ond more extensive set. This is done because it is expensive and impractical to completely surround a person with cameras in a real world setting, synchronise the capture processes and transmit huge numbers of images across a network in real time. It is therefore important to know how reducing the minimal subset of possible cameras impacts on quality for a given reconstruction method. To maximise granularity of the test reconstruction is made from a virtual object because these virtual cameras are not constrained by practicalities of camera size and placement. Direct comparison of the pixels within the resultant image pairs, is used to assess localised differences in form and surface texture.

Resultant maps, plotting the analysis result of each image pair against a Mercator projection of the surrounding evaluation camera's positions, provide a visualisation of holistic quality, allowing assessment of local variations around the entire reconstructed model, including view-point regions between reconstruction source camera positions. These maps are further reduced, by computation of median values, to give a single metric measure for the quality of reconstruction of form or appearance.

5.2.1 Analysis

Visual quality of the 3D reconstructed form has been analysed as a purely mathematical comparison, comparing images acquired from a free-view point virtual camera rotating round the reconstructed form against images gained from the original source model, obtained at the same camera positions and orientations. Two fundamental principles have been tested. Firstly, how well the reconstructed form matches that of the original, by comparing the segmented state of pixels included in the reconstructed form against pixels included in the source model image. Secondly, comparison of the colour vector (rgb) value of pixels included in the reconstructed form and the source form images to assess how well the visual appearance of the reconstruction replicates that of the original model. In both cases these have been assessed for a series of images taken from an inward looking virtual camera tracking across a spherical surface, surrounding the form, at n degree steps (Figure: 5.2, right). In this trial a single capture camera configuration is used. This matches the anticipated arrangement for a CAVE like capture and display environment, where 4 capture cameras are located at each of the CAVE corners, for the bottom edge of the CAVE screens, and at the top centre of the screens for the remaining 4 (Figure: 5.2, left).

To gain the image sets required for qualitative analysis the simulation tool is used to first define a capture and reconstruction environment within the simulation tool. In this a 3D textured, geometric model of the subject is imported form which reconstruction data captured. This is used in to created the reconstruction, in the reconConstructor application (Figure: 5.4). Virtual capture cameras are positioned round the model and their capture parameters set, these include the lens characteristics (intrinsic matrix) and the image resolution, size and aspect ratio of each camera. These cameras are then used to capture the reconstruction data set, a set of images, one for each camera, depicting the camera's background segmented image of the subject, and the calibration data (intrinsic and extrinsic matrix) for each camera. Before reconstruction a second set of camera images is also captured. In this case from a single virtual camera that rotates round the subject model capturing a set of images for fixed steps in longitude and latitude across a spherical surface surrounding the subject model (Figure: 5.2, right). These images are stored, and the camera position for each image is recorded. This later set of images is used for quality evaluation.

Evaluative image acquisition and preparation

To enact 3D reconstruction, the source images and camera data is input into the reconstruction system and used to generate a 3D reconstruction of the subject model using which ever technique is being tested. The virtual camera positions, and characteristics, used to capture the second surround data set are then loaded into the reconstruction tool and a matching set of images are recorded at exactly matching camera positions, delivering an image pair with perfect pixel alignment.

The result of this process is a set of image pairs taken from a virtual camera stepping across a spherical surface surrounding the reconstruction subject (Figure: 5.3). In each pair, one image records the source ('truth') geometric model used to generate the simulated reconstruction data, the second records the reconstructed form generated from the simulated reconstruction data. Each image in one set has a matching image in the second, generated from exactly the same position and orientation relative to the reconstruction subject. Therefore, comparative analysis of each image pair can be used to identify differences between the source model, and the reconstruction form generated by the process used. These images are captured as 32bpp RGBA images against a transparent background, therefore, background seg-

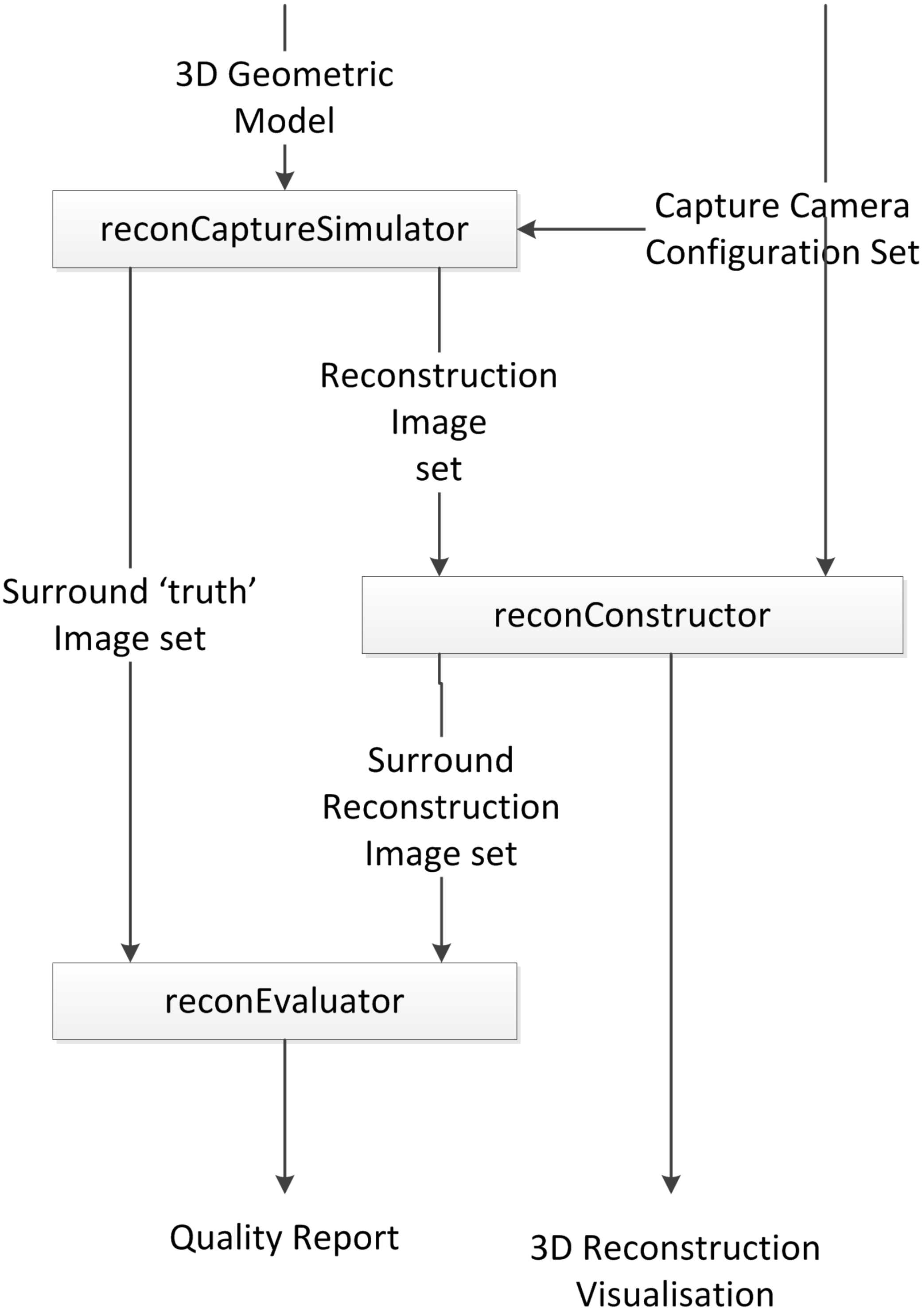

3D Geometric Model
reconCaptureSimulator
Capture Camera Configuration Set
Reconstruction Image set
Surround 'truth' Image set
reconConstructor
Surround Reconstruction Image set
reconEvaluator
Quality Report
3D Reconstruction Visualisation

Figure 5.4: Outline of applications and data flows comprising the simulated capture, reconstruction, and evaluation pipeline

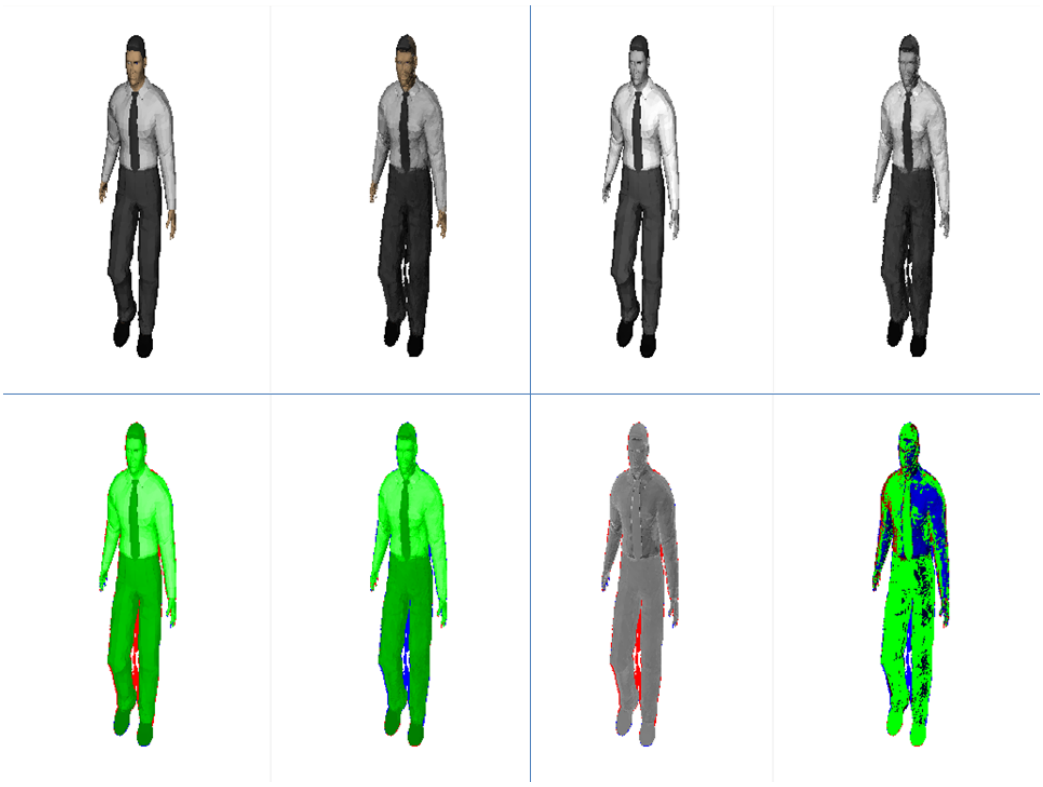


Figure 5.5: Image pair undergoing evaluation of quality. In each pair the left image is captured from the source geometric form and the right image from the reconstructed form. Top left: unprocessed images. Top right: image pair after grey scaling and colour normalization. Bottom left: Form analysis, green denotes matching pixels, red indicates that a segmented pixel is present in the matched image, but cannot be matched in the current image, blue denotes segmented pixel present in this image, but not in matching one. Bottom right: Colour (greyscale) analysis, greyscale in left image denotes level of difference in colours between matching pixels, colouring in right image denotes bands of error in colour matching, green¡5% error, blue 5%-10% error, red ¿10% error.

mentation is inherent in the process and does not need to be enacted as a pre-processing, image analysis step, which may introduce artefacts. In all images the 24bpp RGB colour is reduced to a single grey scale 8bpp value, with whole image colour normalisation of these pixel values enacted to allow comparative evaluation between the image pairs (Figure: 5.5).

Form Analysis

The form analysis seeks to quantify the relative error between the segmented pixels within each comparative image pair. To determine this, each pixel within the segmented form of the reconstruction image was compared against its matching pixel in the source image. Any segmented pixel within the reconstruction image that does not have a matching pixel within the source image is, therefore, erroneously generated by the reconstruction technique and recorded. This count includes both reconstruction artefacts, segmented pixels that occur within the reconstruction but not the source, and reconstruction omissions, segmented pixels that occur within the source image but not the reconstruction. For each image pair, a form reconstruction error value is calculated as the proportion of erroneously generated pixels, relative to the total number of segmented subject pixels within the subject image (Figure: 5.5, bottom left).

Visual Appearance (colour/greyscale) analysis

In assessing the visual appearance of the generated form we are seeking to determine how well the texture applied to the form generated by the reconstruction process matches that of that of the original textured source model.

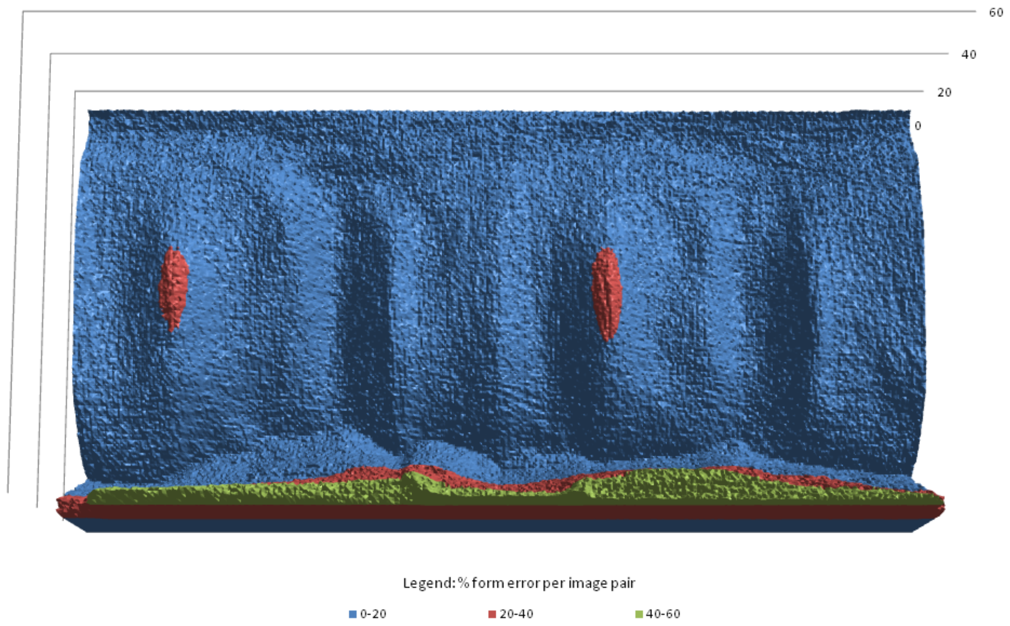


Figure 5.6: Mercator projected plot of analysis of form errors for each image pair taken at 3 degree steps across the surrounding spherical surface (slice plane method). X-Axis: longitude of camera position on sphere surface (meridian at front/centre of the geometric model). Y- Axis: latitude of camera position. Error is calculated as the sum of all erroneous segmented pixels within each image pair as a percentage of the number of matching segmented pixels.

In this case we are not measuring the accuracy of the form reconstruction, but how well the texture application, onto the reconstructed subject, reflects the localised colour variations within the original source model.

To assess this we once again evaluate each segmented pixel within the captured image pair. For each pixel that occurs within the segmented form in both the source and reconstruction image we record the absolute grey scale colour value difference between the two matching pixels. The sum of this absolute error is recorded and divided by the number of segmented pixels occurring within both images to deliver an average pixel grey scale colour value error for the entire reconstruction from the image pair's observation view point (Figure: 5.5, bottom right).

Reporting

The evaluation of each image pair delivers an expression of relative error, in terms of form and texture matching, for a single viewpoint. This is useful for evaluating reduction in reconstruction quality when the observation viewpoint alignment is significantly displaced from one of the capture camera positions, thereby denoting how effective the reconstruction process is, rather than when the image pair viewpoint is closely aligned to a capture camera position, which is simply demonstrating how effective the projection process is. The quantitative data delivered by this process is most useful when compared to results for all the other image pairs within the captured set as this characterises the quality reduction exhibited around the entire reconstructed form. Image pairs are captured for an equal number of points at each latitude on the spherical surface surrounding the reconstruction subject. These can

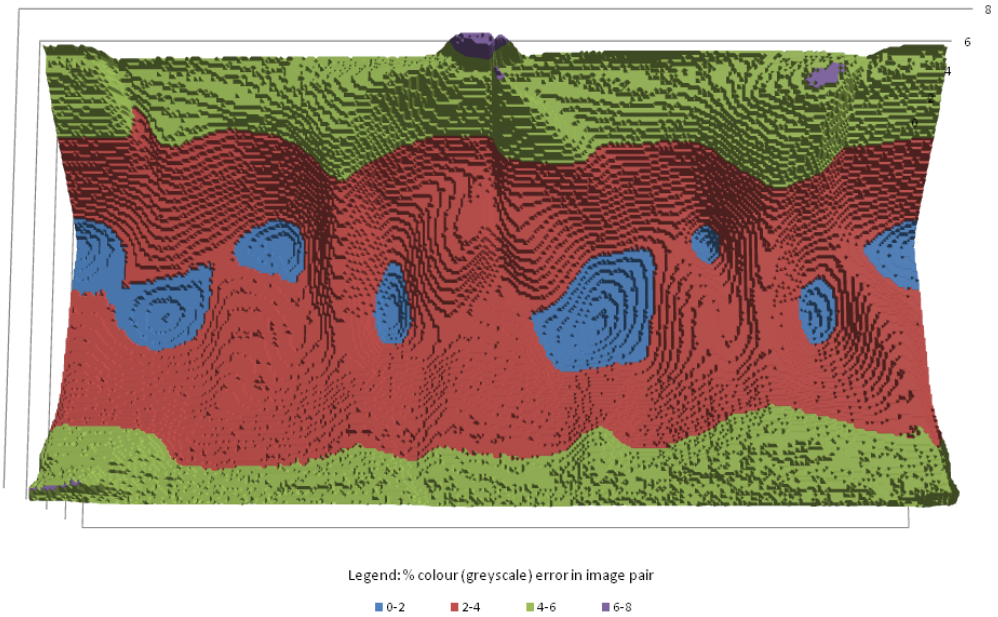


Figure 5.7: Mercator projected plot of analysis of colour (greyscale) discrepancy for each image pair taken at 3 degree steps across the surrounding spherical surface (slice plane method). X-Axis: longitudinal position of camera (meridian at front/centre of the geometric model). Y- Axis: latitudinal of position of camera. Colour discrepancy is calculated as the average discrepancy between each segmented pixel pair's greyscale value (as a percentage of the colour range) for each image pair.

then be plotted out to a rectangular table which, in effect gives a Mercator projected map of the relative errors at each point surrounding the subject (Figures: 5.6 & 5.7).

Plotting this data as a graph enables a visual comparison of the relative quality of a reconstruction for the entire free-viewpoint surrounding the subject. Predictably, lower levels of error are seen for any position on this graph where the image pair view point is closely aligned with a capture camera position.

An overall level of error, for both form and appearance can also be generated by calculating the median error for all of the image pairs in the evaluation image sets. However, in the case of 3D reconstruction of people to support tele-present communications it is highly unlikely that the observer of a reconstruction will choose to locate themselves directly above or below the reconstructed person, instead they are more likely to align themselves at eye level to the reconstructed person to form a more natural conversation configuration. These reconstruction systems are more likely to attempt to generate the highest quality reconstruction for these configurations and position capture cameras accordingly. Therefore, an error median of the central band of the spherical analysis, i.e. between +45 degrees and -45 degrees of the equator, is more likely to reflect the reconstruction characteristics required for the intended use.

5.3 Case Study

In this study two approaches for sampling the reconstruction space in the volumetric reconstruction technique are compared (Figure: 5.8). In this technique, a 3D location within the reconstruction space (volume) is tested for inclusion within the 3D form (Form Evaluation) and, if this initial condition is met, further evaluation is undertaken to determine the actual fragment colour (Colour Evaluation). Form evaluation tests each sample point against the segmented, forward projected set of reconstruction images to determine if that sample point occurs within the silhouette of the form to be reconstructed for all input images. If this fragment is accepted, it is further evaluated to determine the colour of the fragment by performing a weighted blending (with the weighting proportional to the angular proximity of reconstruction image projection to view direction), of the source image's projected pixel colour value. If the sample point fails the initial form inclusion test, the fragment is rejected. This process operates entirely within the GPU processing of the graphics card, and is implemented using GLSL routines which utilise inbuilt functionality for multi-texture projective texturing.

Testing the sample points within the sample volume is computationally expensive. Many points will need to be evaluated, at least one per screen pixel multiplied by the number of depth samples required to traverse the sample volume at a similar spatial resolution to the screen image. However, the reconstruction form occupies a small region within this space, so many sample points will need to be rejected. In effect we are sampling a 3D volumetric space; this is very similar to the sampling mechanisms required for

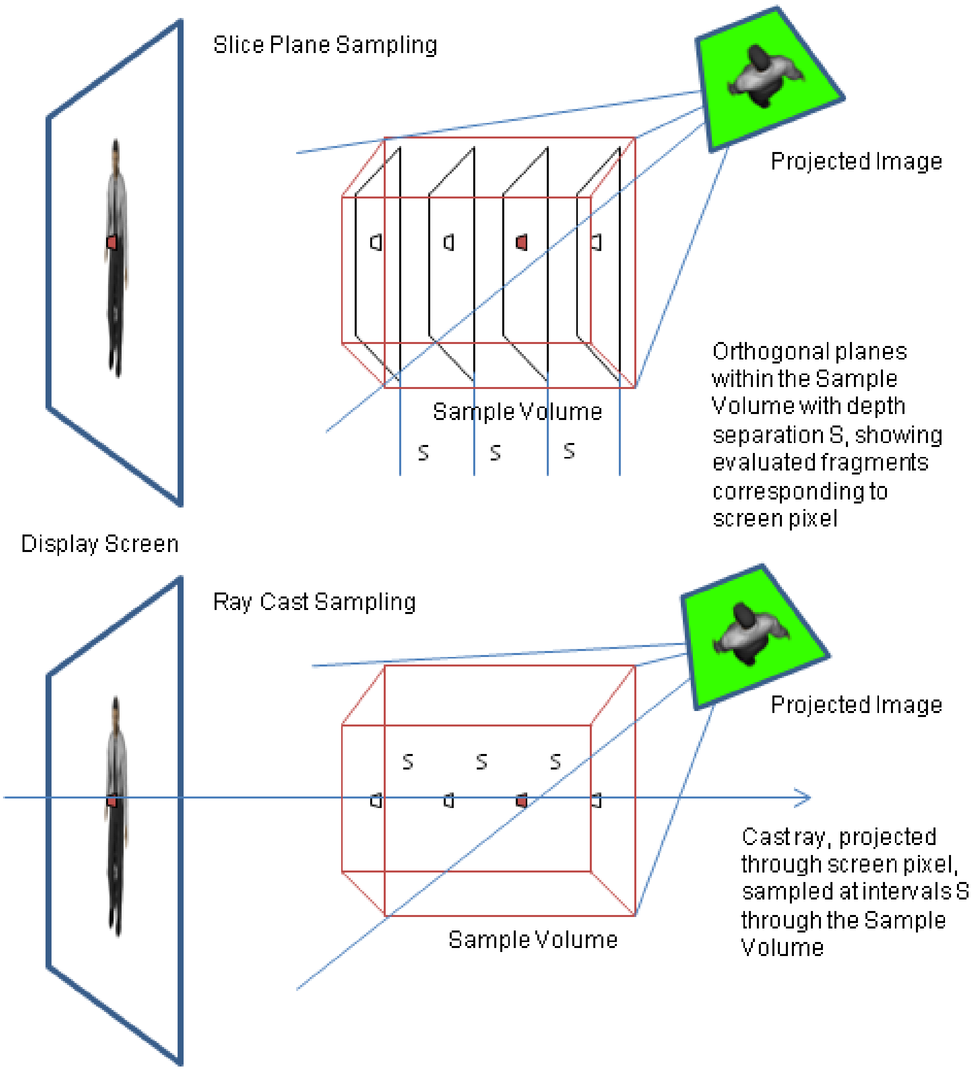

Slice Plane Sampling
Projected Image
Sample Volume
S
S
S
Orthogonal planes within the Sample Volume with depth separation S, showing evaluated fragments corresponding to screen pixel
Display Screen
Ray Cast Sampling
Projected Image
S
S
S
Cast ray, projected through screen pixel, sampled at intervals S through the Sample Volume
Sample Volume

Figure 5.8: Two different variations of volume space sampling. Top: Slice plane sampling, fragments on orthogonally aligned planes within the sample volume are tested for inclusion in the segmented projected images (only one projection image shown). Bottom: alternate approach in which the sample volume is evaluated at discrete intervals along a ray cast through the screen pixel.

3D visualisation of medical data sets. In this case there is not a single 3D texture, but a set of multiple images that are projected in to the volume from the capture camera positions. In the medical imaging domain, two sampling techniques are commonly used for addressing this operation and both fulfil our requirement for sampling. Slice plane sampling [8] utilises a set of depth ordered polygons, orthogonally aligned to the viewer orientation, drawn using the CPU based rendering instructions. When processed within the hardware graphics pipeline, each fragment of each of these polygons is evaluated by a GLSL fragment programme, which can be written to perform the required tests (Figure: 5.8, top). The alternative is to use a GPU based ray-casting algorithm. In our case this is implemented by the rendering of a single, orthogonally aligned polygon, drawn in front of the volume space. Depth sampling is achieved by modifying the slice plane GLSL fragment programme to include a ray case operation that samples points along a ray formed by the viewpoint and the target fragment at depth steps equivalent to the slice plane separation in the previous method.

In this case the ray is 'walked' along until either the maximum sample depth has been reached, in which case the fragment is discarded, or until the first sample point within the reconstruction form is found, in which case the fragment colour is evaluated and returned as the fragment colour (Figure: 5.8, bottom). These sampling approaches have similar speed performance (approx 15fps) on our test system, however, the latter imposes a greater load on the GPU processing pipeline, while the former offsets this by utilising the CPU rendering functionality to pre-determine sample depths.

The requirement is, therefore, to determine if either of these approaches has an advantage in terms of reconstruction/rendering quality. Principally this is determining if there is any substantial difference between the ray-casting and, our preferred, slice plane solution, with the latter approach offering a better system wide load balancing that reduces the complexity of the GPU activity, in terms of depth of iterative loops processed, thereby freeing up GPU resources to add more functionality for additional reconstruction refinement.

The trial seeks to characterise the quality of output for the two approaches by controlled manipulation of the input sources and data to gain an understanding of the appropriate input parameters for each approach and their impact on the resultant generated reconstruction. To achieve this reconstructions are evaluated using the quantitative evaluation process, under 3 different profiles of input; variations in the numbers of source images, variations in the resolution of the input images, and variations in the sampling density used within the reconstruction space.

As the two reconstruction processes use the same input data characterisation using the spherical maps is likely to show similar trends in the variation of reconstruction quality across the entire form. Instead it is more interesting to consider the overall quality of reconstruction, i.e. how well each technique's generation of form and surface texture matches that of the original for any view-point. Therefore, numerical analysis of the overall quality of reconstruction is more important. To generate comparative measures for the two approaches median error was therefore calculated for the entire set of image pairs in each analysis. This gives an overall measure of how well the reconstructed form matches the source object in its entirety, while standard deviation shows how much variation in the reconstructed form is likely to be evident as the observer viewpoint changes. In comparing these two techniques the one showing lower levels of error and lower levels of deviation is likely to have the better overall reconstruction from any viewpoint. Central band analysis is also presented. This only considers the median values for each image pair existing within viewing positions likely to be occupied by an observer assuming a natural conversation position to the reconstructed form and is, therefore, a measure that more closely reflects the quality of reconstruction as it is likely to be perceived by the observer.

5.3.1 Impact of Number of image sources (cameras) on reconstruction quality

In this case the relationship between the number of cameras used to gather source imagery and the quality of both the reconstructed form (hull) and the visual appearance of it as a textured entity is explored. In this the expectation is that the quality of the generated form will increase as a direct relationship with the number of source images used. This is broadly the case, with both approaches showing a similar trend and comparable results. In the case of the visual quality, this was expected to degrade as the number of cameras increases due to the blending of greater numbers of source texels from an increasing set of source images. This is indeed the case with the ray-casting method, however in the case of the slice plane method an improvement in noticed. Whilst it is unclear as to what is causing this, the suspicion is that the graphics hardware's functionality for projecting a tex-

	Slice Plane Method				Ray Cast Method			
No. Cameras	Spherical Analysis (%diff)	Spherical Analysis (Std Dev)	Central Band Analysis (%diff)	Central Analysis (Std Dev)	Spherical Analysis (%diff)	Spherical Analysis (Std Dev)	Central Band Analysis (%diff)	Central Analysis (Std Dev)
4	11.83	5.89	12.37	3.83	14.6	4.96	11.42	2.79
5	11.67	5.49	11.34	3.64	14.66	5	11.42	2.72
6	11.81	5.11	10.73	3.5	14.39	4.83	11.52	2.4
7	11.77	4.83	10.24	3.24	15.93	6.03	12.52	3.7
8	10.98	4.19	8.55	2.4	16.94	6.42	13.33	4.11

Table 5.1: Analysis of colour (greyscale) for differing numbers of reconstruction source cameras for the two reconstruction approaches by averaging differences between sets of image pairs. Results quoted for analysis of the median entire spherical comparison set (Spherical) and central band ($\pm 45^0$ of the analysis sphere equator) (Central)

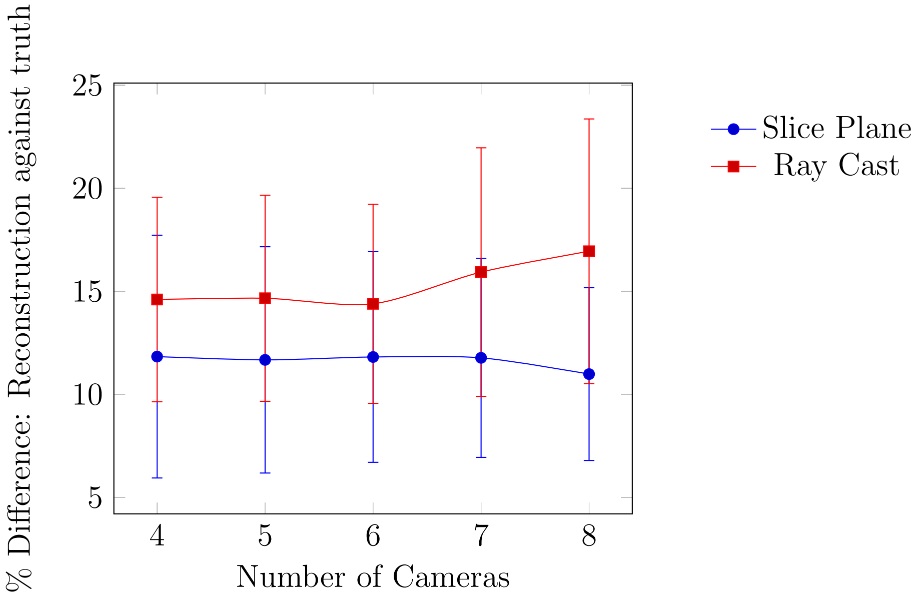


Figure 5.9: Analysis of colour (greyscale) for differing numbers of reconstruction source cameras for the two reconstruction approaches by averaging differences between sets of image pairs across the entire spherical data set (Table: 5.1).

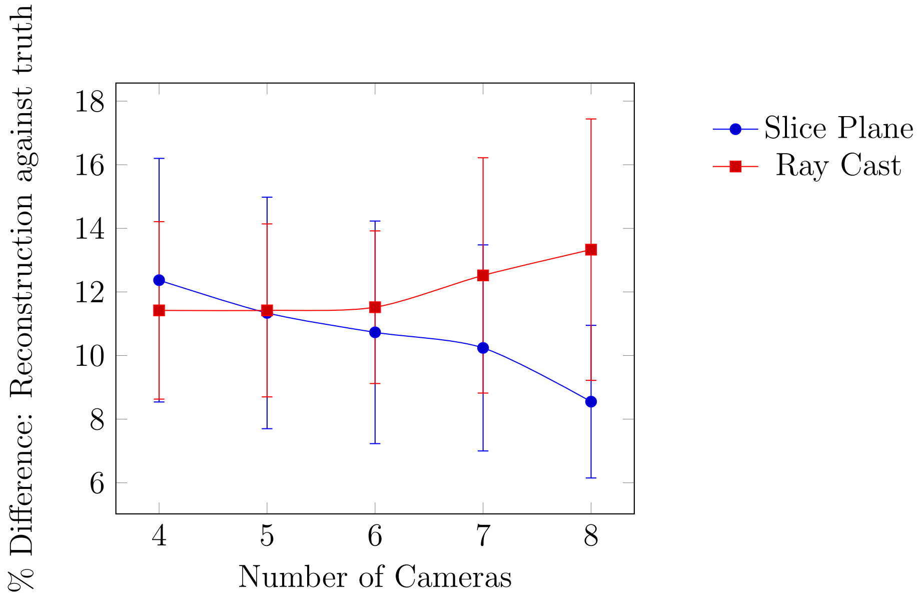


Figure 5.10: Analysis of colour (greyscale) for differing numbers of reconstruction source cameras for the two reconstruction approaches by averaging differences between sets of image pairs across the central band ($\pm 45^0$ of the analysis sphere equator) data set. (Table: 5.1).

	Slice Plane Method				Ray Cast Method			
No. Cameras	Spherical Analysis (%diff)	Spherical Analysis (Std Dev)	Central Band Analysis (%diff)	Central Analysis (Std Dev)	Spherical Analysis (%diff)	Spherical Analysis (Std Dev)	Central Band Analysis (%diff)	Central Analysis (Std Dev)
4	29.87	28.44	15.05	11.97	30.39	33.69	14.43	10.42
5	24.7	26.13	12.42	12.67	24.79	30.18	10.98	10.92
6	20.92	23.7	10.8	11.24	22.32	26.74	10.64	9.29
7	16.29	20.69	6.94	7.03	16.25	23.23	6.83	4.62
8	11.43	18.34	3.12	3.15	12.25	20.7	3.86	2.38

Table 5.2: Analysis of form for differing numbers of reconstruction source cameras for the two reconstruction approaches by averaging differences between sets of image pairs. Results quoted for analysis of the entire spherical comparison set (Spherical) and central band ($\pm 45^0$ of the analysis sphere equator) (Central)

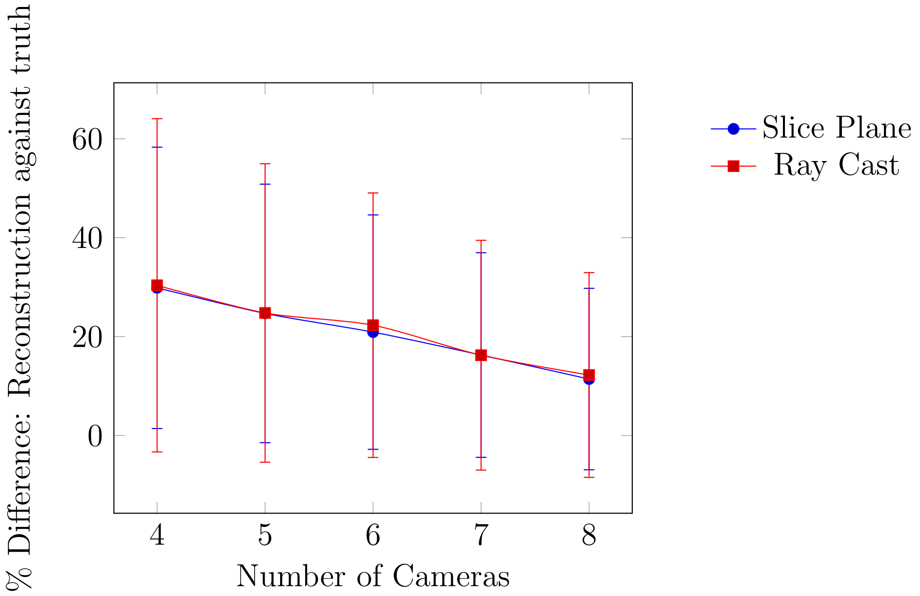


Figure 5.11: Analysis of form for differing numbers of reconstruction source cameras for the two reconstruction approaches by averaging differences between sets of image pairs across the entire spherical data set (Table: 5.2)

ture onto a planar primitive, rather than evaluating this on an individual point basis, as enacted by the ray casting approach, gives better accuracy within the hardware's texture minification/magnification filters (Tables: 5.1 & 5.2).

5.3.2 Impact of Texture resolution on reconstruction/rendering performance

In considering the effect of source image resolution on both form generation and texturing (visual quality) it was fully expected that both would benefit from higher resolution images (Figures: 5.15 & 5.16). This is certainly the case for form generation, however rather than the linear relationship expected

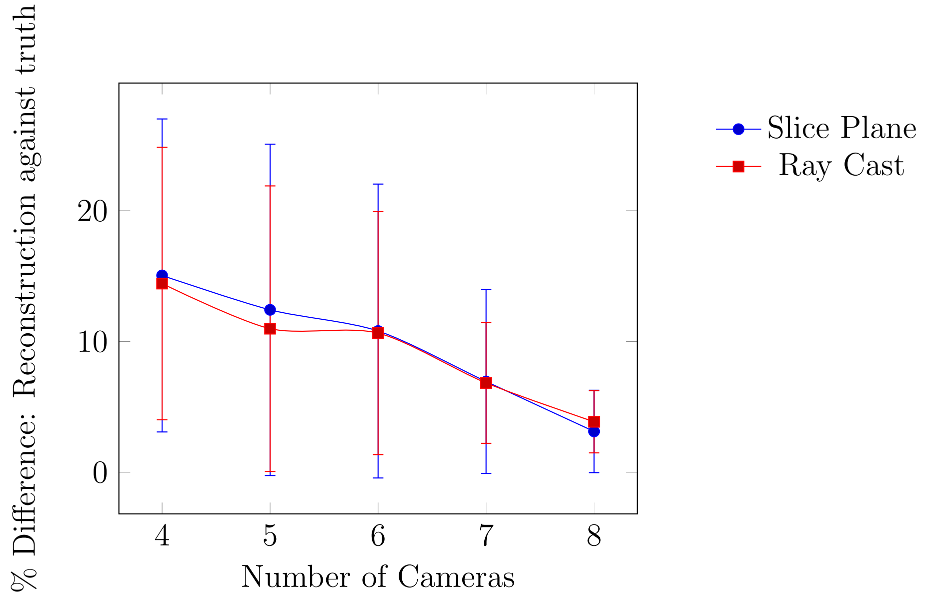


Figure 5.12: Analysis of form for differing numbers of reconstruction source cameras for the two reconstruction approaches by averaging differences between sets of image pairs across the central band (±45^0 of the analysis sphere equator) data set. (Table: 5.2).

			Slice Plane Method				Ray Cast Method			
Image Width (pixels)	Image Height (pixels)	Image Size (pixels)	Spherical Analysis (%diff)	Spherical Analysis (Std Dev)	Central Band Analysis (%diff)	Central Analysis (Std Dev)	Spherical Analysis (%diff)	Spherical Analysis (Std Dev)	Central Band Analysis (%diff)	Central Analysis (Std Dev)
2,048	2,048	4,194,304	10.74	4.08	8.48	2.49	16.45	6.12	12.82	3.71
1,024	1,024	1,048,576	10.98	4.19	8.55	2.4	16.94	6.42	13.33	4.11
512	512	262,144	11.17	4.2	8.74	2.36	15.53	6.22	12.2	3.93
256	256	65,536	10.23	3.8	8.27	2.31	11.2	4.33	9.15	2.74
128	128	16,384	8.11	3.25	6.63	1.86	4.26	1.52	3.54	0.82
64	64	4,096	1.21	0.78	0.85	0.4				

Table 5.3: Analysis of colour (greyscale) for differing source image resolutions for the two reconstruction approaches by averaging differences between sets of image pairs. Results quoted for analysis of the entire spherical comparison set (Spherical) and central band ($\pm 45^0$ of the analysis sphere equator) (Central)

			Slice Plane Method				Ray Cast Method			
Image Width (pixels)	Image Height (pixels)	Image Size (pixels)	Spherical Analysis (%diff)	Spherical Analysis (Std Dev)	Central Band Analysis (%diff)	Central Analysis (Std Dev)	Spherical Analysis (%diff)	Spherical Analysis (Std Dev)	Central Band Analysis (%diff)	Central Analysis (Std Dev)
2,048	2,048	4,194,304	12.94	18.06	4.77	3.66	16.03	24.49	4.89	3.69
1,024	1,024	1,048,576	11.43	18.34	3.12	3.15	12.25	20.7	3.86	2.38
512	512	262,144	11.95	16.74	5.33	2.74	17.04	9.79	15.82	4.12
256	256	65,536	20.47	11.87	18.67	4.64	39.35	11.13	44.76	3.08
128	128	16,384	44.52	11.15	47.96	2.7	83.07	2.4	84.28	0.98
64	64	4,096	95.51	3.19	97.59	0.76				

Table 5.4: Analysis of form for differing source image resolutions for the two reconstruction approaches by averaging differences between sets of image pairs. Results quoted for analysis of the entire spherical comparison set (Spherical) and central band ($\pm 45^0$ of the analysis sphere equator) (Central)

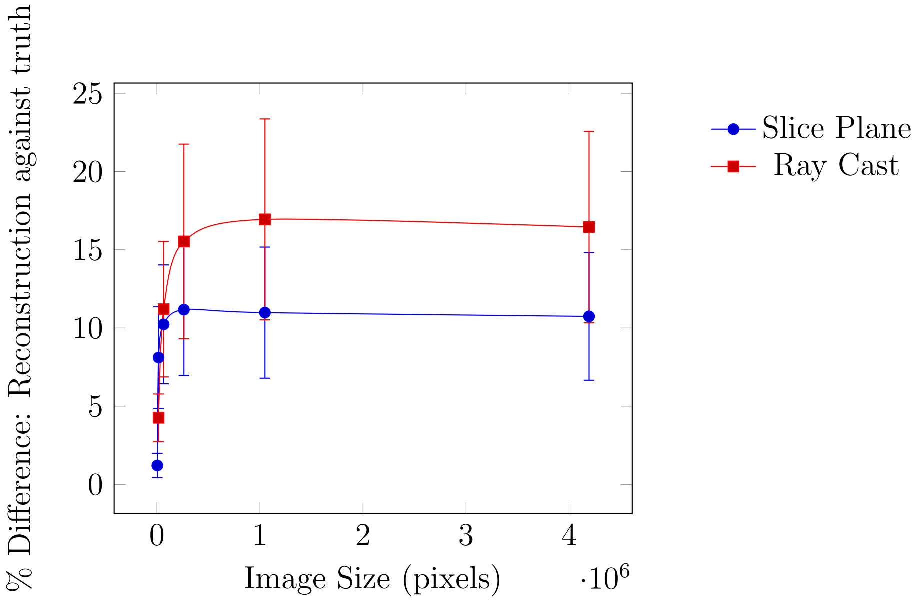


Figure 5.13: Analysis of colour (greyscale) for differing source image resolutions for the two reconstruction approaches by averaging differences between sets of image pairs across the entire spherical data set (Table: 5.3).

a more complex pattern is seen with a slight optimal occurring with the use of 1024^2 images. The degradation above this resolution may be to do with the internal management of textures within the graphics hardware; however the non-linear increase in form quality with image resolution below this is a surprise. Coupled with the relatively constant relationship between image size and visual quality above 512^2 pixels this may mean that a image resolution of 512^2 may be acceptable in most cases, offering performance benefits. However, this may remove some of the key fine detail features, such as eye pupils, that are required for effective communication.

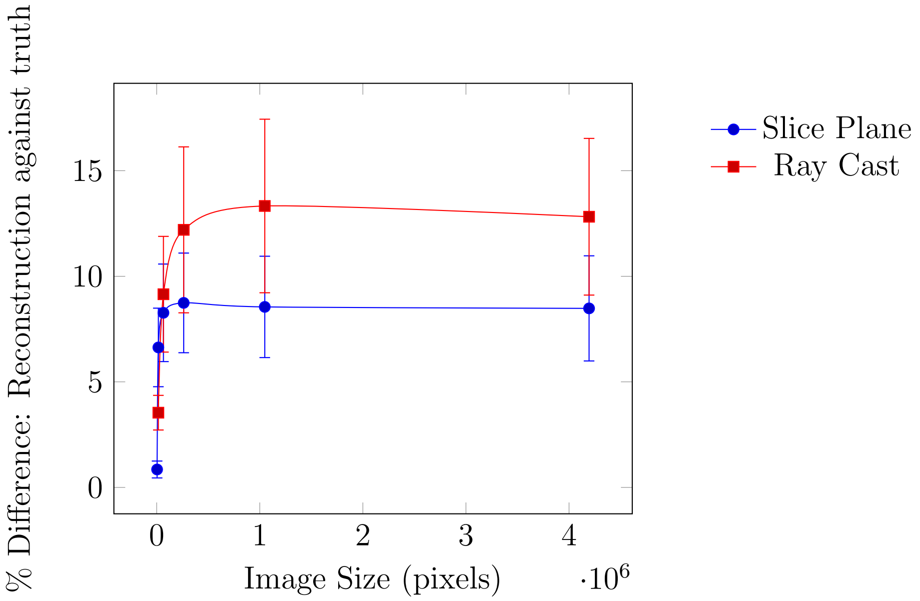


Figure 5.14: Analysis of colour (greyscale) for differing source image resolutions for the two reconstruction approaches by averaging differences between sets of image pairs across the central band ($\pm 45^0$ of the analysis sphere equator) data set. (Table: 5.3).

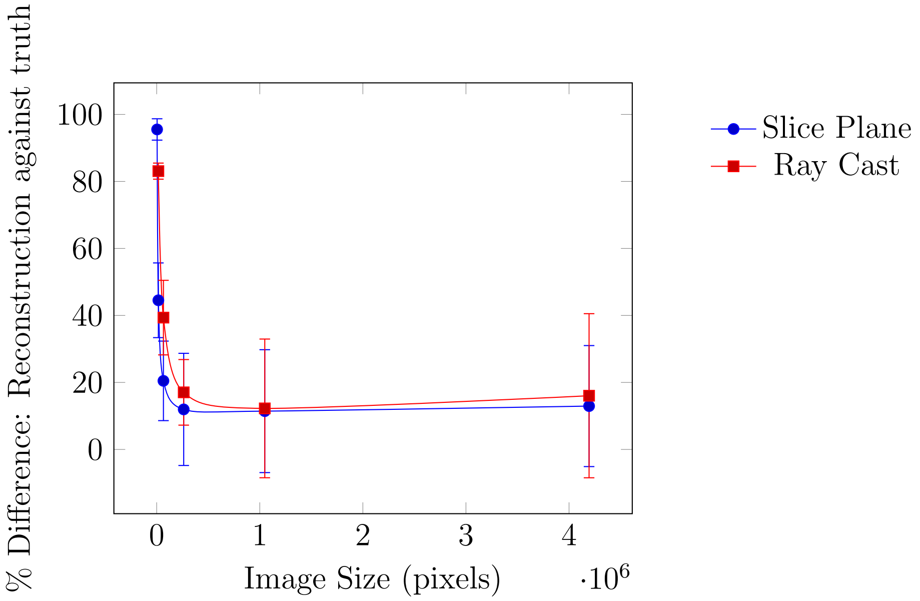


Figure 5.15: Analysis of form for differing source image resolutions for the two reconstruction approaches by averaging differences between sets of image pairs across the entire spherical data set (Table: 5.4).

5.3.3 Impact of sampling interval on reconstruction/rendering quality

The depth sampling process is the key difference between the two approaches being evaluated in this case study. Both should deliver the same number of sample points, at the same density within the volumetric sample space, however the methods for determining these will deliver localized variations in the actual screen space position of each sample point. In the case of the slice plane approach a sample point is defined by a screen fragment (effectively a pixel) occurring on one of the orthogonally aligned planes within the geometric structure used to intersect the multi-texture projected space. Evaluation of each projected source image texel is therefore performed as an interpolation

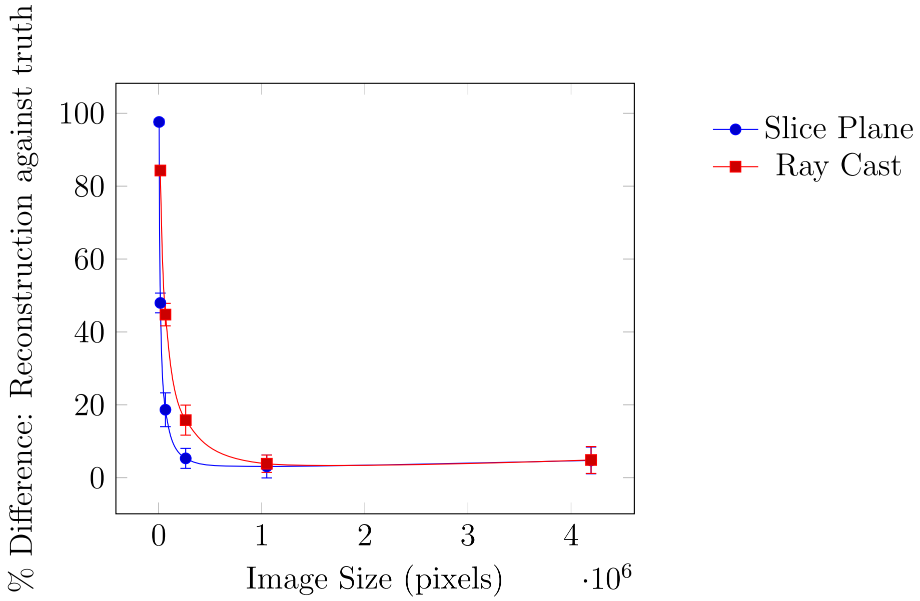


Figure 5.16: Analysis of form for differing source image resolutions for the two reconstruction approaches by averaging differences between sets of image pairs across the central band ($\pm 45^0$ of the analysis sphere equator) data set. (Table: 5.4).

	Slice Plane Method				Ray Cast Method			
Sample Depth Interval (m)	Spherical Analysis (%diff)	Spherical Analysis (Std Dev)	Central Band Analysis (%diff)	Central Analysis (Std Dev)	Spherical Analysis (%diff)	Spherical Analysis (Std Dev)	Central Band Analysis (%diff)	Central Analysis (Std Dev)
$1.6 \cdot 10^{-3}$	11	4.19	8.57	2.4	16.9	6.41	13.27	4.08
$1.8 \cdot 10^{-3}$	10.98	4.17	8.55	2.38	16.92	6.42	13.28	4.1
$2 \cdot 10^{-3}$	10.99	4.17	8.56	2.39	16.91	6.41	13.29	4.1
$2.3 \cdot 10^{-3}$	10.97	4.18	8.54	2.39	16.94	6.41	13.31	4.12
$2.7 \cdot 10^{-3}$	10.98	4.19	8.55	2.4	16.94	6.42	13.33	4.11
$3.2 \cdot 10^{-3}$	10.95	4.17	8.53	2.38	16.94	6.41	13.33	4.12
$4 \cdot 10^{-3}$	10.96	4.19	8.53	2.4	16.97	6.41	13.37	4.12
$5.3 \cdot 10^{-3}$	10.95	4.17	8.52	2.4	17	6.43	13.42	4.14
$8 \cdot 10^{-3}$	10.95	4.19	8.51	2.4	17.06	6.44	13.52	4.19
$1.6 \cdot 10^{-2}$	10.98	4.19	8.58	2.4	17.2	6.47	13.68	4.2

Table 5.5: Analysis of colour (greyscale) for differing sample intervals for the two reconstruction approaches by averaging differences between sets of image pairs. Results quoted for analysis of the entire spherical comparison set (Spherical) and central band ($\pm 45^0$ of the analysis sphere equator) (Central)

	Slice Plane Method				Ray Cast Method			
Sample Depth Interval (m)	Spherical Analysis (%diff)	Spherical Analysis (Std Dev)	Central Band Analysis (%diff)	Central Analysis (Std Dev)	Spherical Analysis (%diff)	Spherical Analysis (Std Dev)	Central Band Analysis (%diff)	Central Analysis (Std Dev)
$1.6 \cdot 10^{-3}$	11.52	18.34	3.2	3.18	12.24	20.79	3.81	2.38
$1.8 \cdot 10^{-3}$	11.5	18.34	3.18	3.17	12.24	20.77	3.81	2.38
$2 \cdot 10^{-3}$	11.48	18.33	3.16	3.16	12.24	20.76	3.82	2.38
$2.3 \cdot 10^{-3}$	11.46	18.34	3.14	3.15	12.24	20.73	3.84	2.38
$2.7 \cdot 10^{-3}$	11.43	18.34	3.12	3.15	12.25	20.7	3.86	2.38
$3.2 \cdot 10^{-3}$	11.4	18.35	3.09	3.13	12.24	20.64	3.89	2.38
$4 \cdot 10^{-3}$	11.36	18.35	3.05	3.11	12.24	20.54	3.94	2.38
$5.3 \cdot 10^{-3}$	11.31	18.38	3.01	3.06	12.24	20.39	4.02	2.39
$8 \cdot 10^{-3}$	11.24	18.36	2.98	2.96	12.23	20.09	4.19	2.43
$1.6 \cdot 10^{-2}$	11.2	18.17	3.21	2.61	12.24	19.16	4.78	2.62

Table 5.6: Analysis of form for differing sample intervals for the two reconstruction approaches by averaging differences between sets of image pairs. Results quoted for analysis of the entire spherical comparison set (Spherical) and central band ($\pm 45^0$ of the analysis sphere equator) (Central)

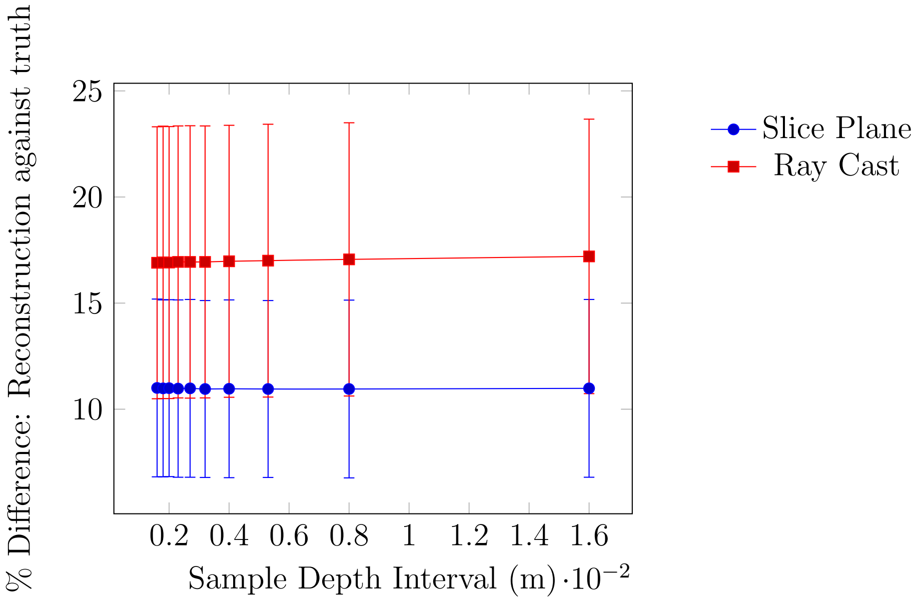


Figure 5.17: Analysis of colour (greyscale) for differing sample intervals for the two reconstruction approaches by averaging differences between sets of image pairs across the entire spherical data set (Table: 5.5).

of the polygon vertices texture coordinate for each source image projection. In the case of the ray casting approach the sample point is determined by stepping along a cast ray denoted by the viewpoint and the current fragment under evaluation on the surface of a single orthogonally aligned polygon. In this case the projected source image texel is determined by a back projection from the sample point to the projected image without any minification/magnification inherent in the interpolated texel determination. Given this, it would be expected the more mathematically accurate ray-casting approach to deliver a more precise matching of projected texels. This should deliver a more accurate form reconstruction, but also potentially a sharper blending of projected texel colour values, giving a harsher representation of visual appearance. However, given that both processes are inherently delivering the

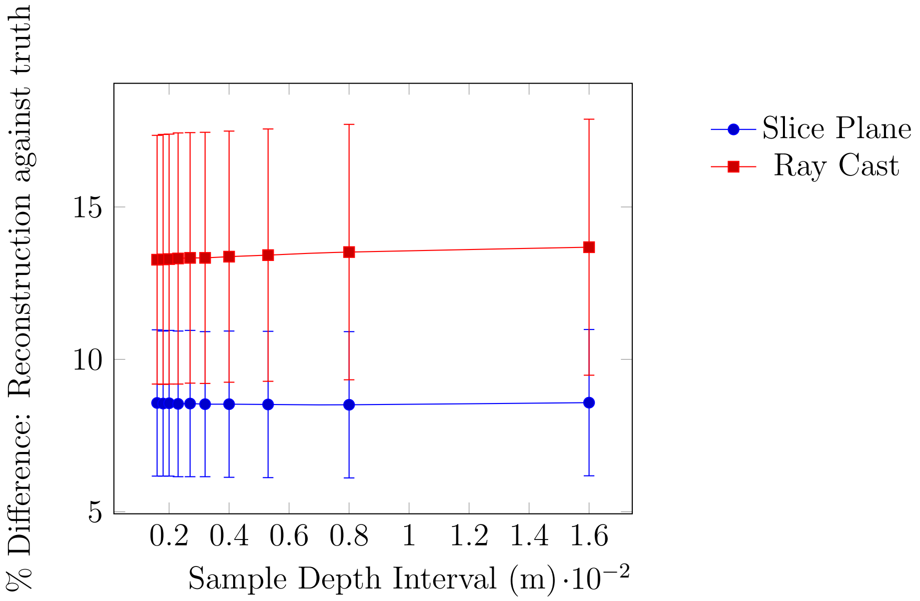


Figure 5.18: Analysis of colour (greyscale) for differing sample intervals for the two reconstruction approaches by averaging differences between sets of image pairs across the central band ($\pm 45^0$ of the analysis sphere equator) data set. (Table: 5.5).

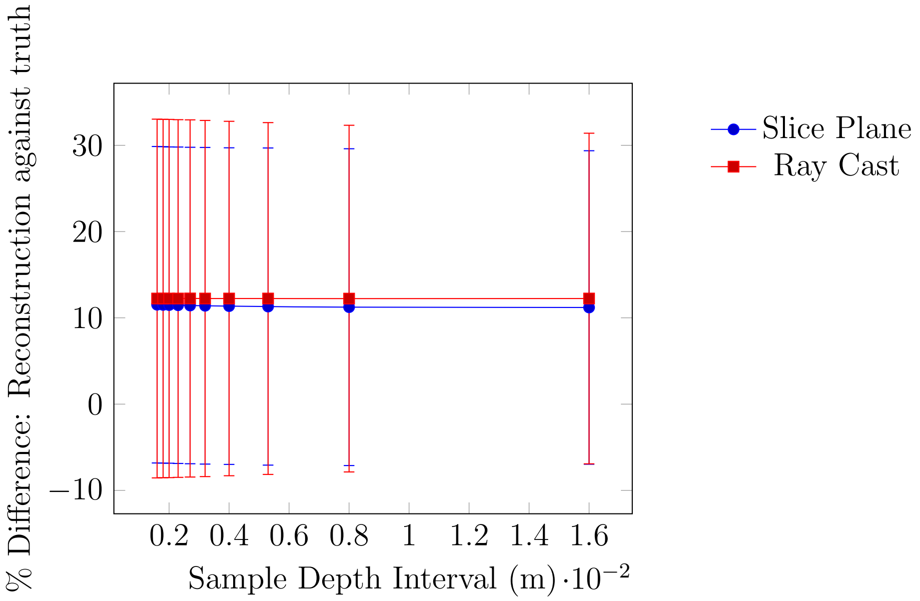


Figure 5.19: Analysis of colour (greyscale) for differing sample intervals for the two reconstruction approaches by averaging differences between sets of image pairs across the entire spherical data set (Table: 5.6).

same sample points, these variations could be expected to be insignificant or slightly favouring the ray-casting approach.

Analysis (Tables 5.5 & 5.6), however, suggests a different outcome. In the case of form evaluation the slice plane sampling approach performs slightly better, possibly due to the min/mag filter providing an element of smoothing at the projected image silhouette edges that reduces the possibility of sample points erroneously being determined as within the reconstruction subject. In terms of visual appearance the slice plane method shows significant benefits over the ray-casting approach. This is most likely due to a combination of factors. As this method delivers a more accurate depiction of the reconstruction subject's form, sample points for which the fragment colour is evaluated as a blending of the projected source image pixels are more representative

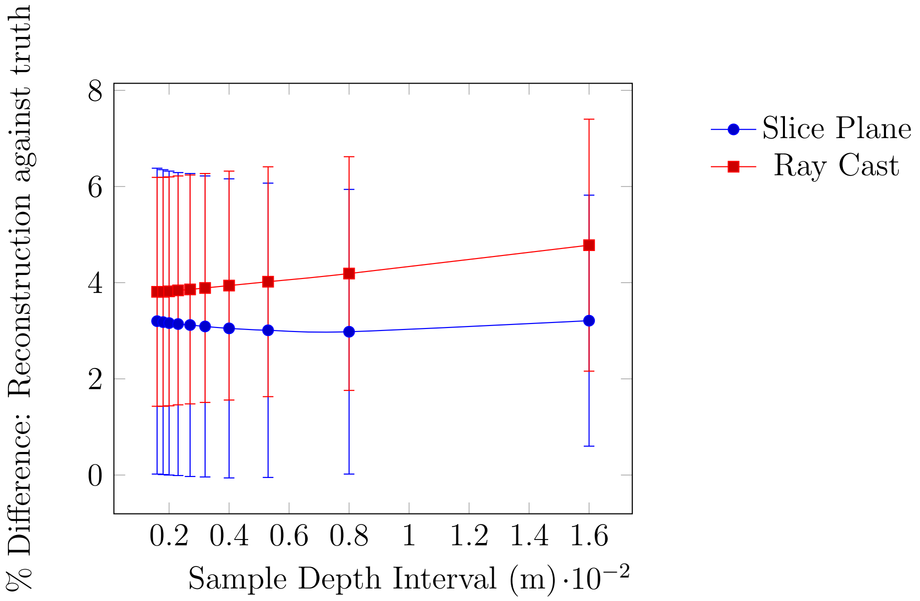


Figure 5.20: Analysis of form for differing sample intervals for the two reconstruction approaches by averaging differences between sets of image pairs across the central band ($\pm 45^0$ of the analysis sphere equator) data set. (Table: 5.6).

of the actual surface of the subject. In addition the interpolation of these projected texels is delivering an inherent smoothing of colour values that is likely to deliver a better blended fragment colour value.

5.4 Discussion

This process for evaluating the quality of the reconstructed form and appearance is made possible through the use of the simulated capture system. This was originally developed to allow rapid experimentation with camera positioning and configuration prior to our physical system being fully operational. However, by extending this to enable a second independent capture of images, from many viewpoints surrounding the reconstruction subject model, is introduced. This gives the ability to enact assessment of view synthesises in a manner that is consistent with the approach taken by Kilner [39], which is identified as suitable for measuring the quality of 3D reconstruction in image space [9] and therefore appropriate for our reconstruction techniques.

The use of a simulated capture and reconstruction system also enables us to remove pipeline error, such as suboptimal camera calibration and noise within the video streams. This removes the need for assessment of PSNR and VQM, which Kilner argues as inappropriate for assessing shape and appearance. The inherently perfect segmentation that the simulated process delivers also removes the need for Kilner's 'Completeness' measure which assesses foreground matting error. Therefore, this approach targets objective assessment of the reconstruction and display processes' generation of shape and appearance by isolating these from the potential sources of error within a live capture pipeline. This is in agreement with Kilner's classification of shape and appearance error and the approach delivers assessment values consistent with those previously presented for Visual Hull based methods.

The elimination of potential error inherent in the capture pipeline is advantageous in this case. By concentrating on detailed objective measures of the reconstruction/display process a methodology is defined that provides fine grained evaluation of the view synthesis. During iterative refinement of a technique, or group of techniques, for reconstruction, small changes to technique will deliver slight improvements in shape and appearance. This approach then enables differentiation between these iterations as part of a targeted continually refined process, that may be steered by traditional subjective analysis operating at more infrequent intervals.

The ability of the approach to deliver multi-viewpoint view synthesis assessment, to potentially infinite granularity, extends the approach taken by Kilner. In this was delivered both a global mapping of the viewing field surrounding the reconstructed subject (Figures 6 & 7) and an objective measure of the overall multi-viewpoint quality of shape and appearance likely to be experienced by an observer within a real-time free navigation environment (Tables 1-6). This bridges the potential application domain, 3D TV through to Tele-Presence, of such techniques, unlike previous approaches which target a single viewpoint which is relevant to 3DTV applications only.

5.4.1 Case Study

The case study presented here is used to illustrate how the evaluation process can be used to inform the development of reconstruction. An evaluation of two variants of an approach is presented. These are essentially the same in principle, but have a minor variation in implementation. Both variants deliver similar reconstruction time performance (in the order of 25fps on the test systems) and, from the perception of an observer, deliver a similar visual quality (Figure: 7.1). However, one, the slice plan sampling approach, has the potential to release processing resources in the GPU system which could be used to further refine the visual quality of the reconstruction. To evaluate both we have sought to characterise each technique's performance against variations in how the input source images are configured (number of capture cameras used and resolution of the source images) and against their susceptibility to differing sampling densities within the reconstruction volume. This both allows tuning of the speed performance of the techniques, and also understanding of the relative merits and limitations for each approach.

In terms of form quality, the slice plane sampling approach appears to have an advantage. In all three of the tests this approach shows less overall error in the form generated and also less variation in the levels of error reported across the multi-viewpoint data set. While this advantage is slight in the cases where the input data configuration (number of cameras and image resolution) is varied, it is more marked in the case of the sampling density.

When the evaluation of visual quality (ie, texturing) is considered there is a more significant bias in favour of the slice plane sampling approach. This is particularly noticeable in the tests that consider numbers of capture cameras used and variations in sampling density. This may be symptomatic of the use of linear minification/magnification filters, used within the graphics hardware texel interpolation process. These deliver a more refined projected source texel value from the input images, than the ray-casting approach which is limited to a 'nearest' texel evaluation. Another factor may well be related to the quality of the reconstruction form. The new process evaluates a view-point weighted blending of the projected texel values for each evaluated fragment on the surface of the determined reconstruction form. If this surface is inherently inaccurate due to a miscalculation of which fragments constitute the surface, typically placing the reconstruction surface outside the original subject surface, then the evaluation of surface colour will be inaccurately selecting projected texel values for blending into a surface fragment colour, resulting in distortions to the final textured surface.

Given these considerations, adopting the slice plane sampling approach both delivers a better visual quality, and also frees GPU processing resources for inclusion of more refined evaluation of both form and surface texture.

The process described here delivers a repeatable and dependable evaluation tool for measuring the relative quality performance of multiple techniques for 3D person reconstruction. This in itself is useful and enables objective metrics to be used in the evolution of techniques. Future work is needed to connect the ability to analyse the simulated capture and reconstruction pipeline to real-world live capture data which would enable a link to be established between the quantitative analysis of reconstruction quality and the human user's perception of the reconstructed forms. Some of this could be achieved by using a similar analysis approach to compare the original captured images, used for reconstruction, against images taken, from the same viewpoints, of the reconstructed form. In this case the smaller subset of image pairs could be used to establish a linkage between key viewpoints within the analysis that would explain how the simulated capture and reconstruction pipeline, which has perfectly calibrated capture cameras generating perfectly segmented images, compares to a real-world situation with less precise characteristics for both.

5.4.2 Wider impact of the objective analysis approach

The use of this approach for objective evaluation of the quality of reconstructed form against a truth data model, enacted by analytical comparison of imagery from matched view points offers a useful tools for detailed investigation of technique. In the case study, presented, this is demonstrated as a method for evaluating design/implementation choices within an iterative development cycle. The use of a simulated capture and reconstruction pipeline to achieve this also offers another important mode of evaluation; direct comparison of two different techniques.

Currently subjective evaluation of a 3D reconstruction is the common approach to evaluation of reconstruction quality. Typically this is enacted within an experimental framework that allows the evaluation to 'objectify' the trial through assessment of conditions such as eye-gaze [72]. However, the nature of the reconstruction challenge is such that it is typically lined to a physical configuration and, therefore, location, with the subjects of the experimentation being specific to that region or institution; because of this comparative evaluation of techniques is rare.

The use of the simulated capture and reconstruction environment developed to support his research potentially addresses this. As well as enabling the creation of standard data sets, some of which do exist (e.g Inria's [35]), the simulator provides a consistent and repeatable tools for evaluation that is not constrained by location or time. Expansion and distribution of the simulator is possible. In this researchers would be able to implement their own approaches to the 3D reconstruction problem, as plug-in components using the same common interface, and evaluate these within the same experimentation framework and datasets as others. This would allow direct comparison of techniques and build up a resource for researchers.

5.5 Conclusion

This chapter presents a method of analysis which enables characterisation of quality of form and appearance in video based 3D reconstruction by evaluation of the quality of multiple view synthesises to potentially infinite levels of granularity. The development of this approach within a simulated capture and reconstruction pipeline has also enabled isolated evaluation of the reconstruction process with sources of error inherent in a live video capture and data preparation process removed. These features of the objective assessment approach therefore deliver a reliable and repeatable framework for evaluation of evolving reconstruction techniques, and comparative evaluation of alternate techniques, with a level of precision sufficient to differentiate improvements in reconstruction outputs within the iterative refinement processes of technique development. This enables targeted refinement of technique towards an overall quality objective. In addition, the extension to multi-viewpoint evaluation of view synthesis, with an unlimited number of viewpoints, has also extended the scope of previous assessment approaches to cover the full range of video based reconstruction applications, ranging from 3DTV through to Tele-presence.

The case study presented uses the assessment approach to differentiate between two, visually similar, variations of the same 3D volumetric rendering technique. The overall results of the view synthesis analysis align reasonably well with those of previous assessments of visual hull-based approaches [39], of which the techniques developed here are a variation. However, the results also demonstrate a clear distinction between the two variations, something that is not visually apparent, which provides quantifiable evidence for selecting one variation of the technique (slice plane sampling) over the other. To be able to objectively differentiate between such subtle refinements is clear validation of the assessment approach.

5.6 Chapter Summary

This chapter concludes the contributions of the research. It presents, and illustrates though the use of a case study, how the tools developed to support the research undertaken can be extended to support objective analysis for evaluating reconstruction technique(s). This offers repeatable, dependable experimentation and quantifiable metrics for comparison of techniques that potentially offer, to the related research community, standardised approaches to comparison and selection of techniques.

Chapter 6

Discussion and Conclusion

This chapter presents a summary of the research undertaken and reviews the aim, objectives and research questions with respect to the work done. The methodology is reviewed and its impact on the research discussed. The key contribution chapters are considered and reviewed against the hypotheses presented. The chapter, and thesis, are then completed with a statement of the conclusions and identification of potential future research directions

6.1 Summary of Work Done

The approach taken was to develop research informed tools and techniques for understanding the capture environment characteristics and enable repeatable, quantifiable, objective analysis of the reconstruction/rendering techniques evolution towards the goals of performance and fidelity established by the current state of the art.

The research undertaken is presented in 3 key chapters:

- **Chapter 3: An Initial Study in Voxel Based Evaluation and Rendering of 3D Reconstructed, Human-like Forms.** In this previous understanding of 3D medical visualisation techniques was used to rapidly produce a proof of concept approach to 3D reconstruction and rendering in which brute force processing was applied to populate a volumetric description of the reconstruction space and an adapted slice plane algorithm used to render this without surface forming. This enabled basic understanding of the challenge to be developed and seeded the formulation and implementation of a simulation platform that would enable rapid experimentation with capture space configuration to determine how this impacted on reconstruction form. Parameters and requirements for the simulation platform were informed by literature review.

- **Chapter 4: An Approach for GPU Aligned Integrated Rendering and Reconstruction.** In this, the findings of the initial research were used to specify, design and develop an evolved version of the simulation platform that enabled repeatable experimentation with capture space configuration and reconstruction subject to deliver standardised test data sets and consistent performance (speed) measurement. Evolution of the reconstruction technique was informed by literature review to enact a transition from a volumetric solution to one utilising GPU features for projective multi-texturing which offered potential for meeting the aim of the research. The performance of this approach was characterised with respect to configurations of input data and demonstrated the potential of the approach as a scalable solution that would gain performance in a defined relationship with improvements in graphics hardware.

- **Chapter 5: Validating the Approach.** In this a case study presenting objective analysis of performance (speed) and fidelity(quality), comparing two variations in sampling techniques within the reconstruction/rendering process, is used to demonstrate how the simulation framework developed, supports decision making for iterative refinement of technique. This demonstrates how the understanding gained through the research has been applied to define repeatable, reliable evaluation. In achieving this it introduces the concept of holistic evaluation of the 3D reconstructed form as a way to fully evaluate free-viewpoint quality and that end-to-end performance timing is essential for evaluating latency in a combined reconstruction/rendering technique.

6.2 Point of Departure

This work builds on previous work investigating 3D reconstruction of people from multiple, synchronised images that enabled an actor within a multi-camera space to be recorded and reconstructed within a virtual image space [1, 25, 26, 93]. In that research recorded capture data from real-world environments was used to enact reconstructions and enable subjective evaluation of the resultant form. Various approaches to reconstruction were reported with free-view point surfaced, volume based reconstruction ('space carving') [56, 24, 44, 25] and Depth Based Image Rendering [16, 18]. This research sought to combine the attractive features of each of these approaches to achieve fast, efficient, free-view point 3D reconstruction without the computational over head of traditional surface based approach and view constraint limitations of DBIR. Whilst this work demonstrated a reconstruction approach that combined the volumetric starting form of polyhedral approaches with image projection techniques, inspired by DBIR, [3] to deliver both static and dynamic reconstructions, a key weakness was that this was not achieved in a real-time capture/display environment.

6.3 Review of Aim and Objectives

This research aimed to improve and study the combined visual, spatial and temporal qualities of video based reconstruction. In particular it was seeking a lightweight solution to the video based 3D reconstruction and rendering problem that would allow balancing of visual, spatial and temporal qualities to achieve targeted solutions. As such, a significant part of the research sought approaches to aiding the development of such techniques through rapid prototyping supported by repeatable, objective assessment of the holistic form created by the reconstruction/rendering technique.

Specifically the objectives for this research are:

- **O1:** Develop a novel approach to holistically measure the balance of visual, spatial and temporal qualities of video based reconstruction while varying key impacting factors.
- **O2:** Develop a novel reconstruction/rendering technique that improves

upon the balance of visual, spatial and temporal qualities of video based reconstruction.

- **O3:** Assess the novel technique to improving the balance of qualities by applying the novel approach and objectively measuring its holistic performance.

These were formulated into a single research question: Can 3D video based reconstruction and rendering be combined and reduced to remove computationally expensive processes and deliver a GPU aligned approach that matches current state of the art performance and quality metrics within a componentisable form suitable for future consumer type systems.

The parallel evolution of the simulation tools and the reconstruction/rendering technique have blurred the boundaries for these objectives. Initial work undertaken to conceptualise the problem (Chapter 3) delivered a crude approach to reconstruction and rendering, but did demonstrate that a simulation framework was possible and could support evaluation of technique and aid understanding of the impacts of the capture environment on reconstruction outcome. Initially, refinement of this technique was sought, however, research suggested a significant change in approach and a new technique for reconstruction was identified that better matched the GPU architecture, offering improved performance and scalability (Chapter 4). Supporting this change of approach, was also a significant evolution in the form and structure of the evaluation tools and simulation framework that supported the research. This enabled dependable characterisation of performance (Chapter 4) with useful objective measures for fidelity and performance, required significant development of understanding and experiential research before this could be demonstrated (Chapter 5). The use of this simulation framework, both in the work presented here, and in support of the work of others [19, 61] shows that such an approach is a positive aid to development of technique, and potentially a useful tool for the wider research community).

The initial approach to reconstruction and rendering (Chapter 3) did little to address the question of addressing an algorithm that utilised the potential computational power of the graphics hardware (**Q2**). However, the change in approach (Chapter 4), informed by literature review while seeking ways of texturing the crude volumetric form, presented an algorithmic technique with features that potentially addressed these. Further refinement of this technique, which demonstrated scalability (Chapter 4) and fidelity (Chapter 5) demonstrate how the technique, developed, utilises GPU processing power by embedding the majority of the process into the reprogrammable graphics pipeline.

Determination of quantifiable, objective measurements that could establish confidence in matching technique to current state of the art performance and quality proved a significant challenge. Little established literature existed in this area. Reported performance commonly only concerned the reconstruction process [56, 58, 102, 13, 90] and did not consider latency introduced through the rendering process of the model generated. In adopting an approach in which reconstruction and rendering were implemented within the graphics pipeline meant that reconstruction timings could not be isolated. However this work argues (Chapter 4) that in evaluating the speed performance of a real-time process latency introduced within the rendering pipeline is equally likely to affect end user perception of the reconstruction and that end-to-end cycle time is a more appropriate measure. Subjective analysis of visual quality remains the norm within the field where user perception of the presented reconstruction form is evaluated through user trials. However in depth image based rendering techniques measures such as Peak-Signal-to-Noise-Ratio (PSNR) and Video Quality Metric (VQM) are commonly used to assess the resultant representation, from a single viewpoint, of 3D video delivery [22] and form the basis for identifying quality improving techniques [18]. Kilner [39] argues against subjective analysis and conventional pixel-wise evaluation, suggesting that these measures do not reflect the quality of view synthesis as perceived by the user. In this model an evaluation of image pairs, based on a leave-one-out test, is used to measure reconstruction quality rather than fidelity, presenting measures in quality of reproduction of shape and appearance rather than PSNR and VQM. It suggests that this framework is more suited to the measurement of quality of 3D reconstruction in image space, pointing out that this is only applicable to geometry based 3D reconstruction techniques [9]. The approach to holistic objective analysis presented in this research (Chapter 5) potentially addresses this by offering a overall assessment of the reconstruction quality independent of view point. This makes the approach potentially applicable to both model forming processes and image based rendering approaches. The evaluation metrics presented by this research offer a direct comparison to 'truth' models which establish quantifiable data for comparison against any technique and establish the performance of the technique as matching current state of the art techniques for performance and fidelity.

6.4 Review of Methodology

The methodology applied sought to address the aim by pursuing the activities of doing, developing and measuring 3D video based reconstructions as a holistic framework for bettering understanding and new technique development. In this the methodology applied was used to manage a transition from known techniques and naive application of these, through established knowledge and understanding to the discovery of new approaches and understanding.

Aim, Objectives, and Research Question were identified at the start of the research. These formed the basis of research, which was conducted through iterative development of approach to reconstruction supported by evolution of simulation frameworks and evaluation tool kits to continually assess the performance and fidelity of the reconstruction achieved.

In this a basic prototype system for both reconstruction and simulation of the capture/reconstruction pipeline was built using initial understanding as a platform for development and refinement. This prototype proved highly valuable in that whilst the approach created was deeply flawed, it identified the need for a platform for experimentation that could then drive a further review of literature and refinement of approach. As the research developed the tools that this simulation platform provided became increasingly useful, both as their sophistication improved and as the techniques for reconstruction became more refined. In this the ability to measure within a dependable and repeatable experimental platform enabled identification of specific research/developmental tasks, such as the way finding approaches to texturing the initial volume model lead to a step change in the approach to remove the volume construction. In effect, the model for research became integrated with each of the three objective areas being progressed together as the research question was undertaken. This meant that rather than having multiple iterations through the research with each refining the understanding of the problem, three distinct investigations (naive investigation (Chapter: 3), informed prototyping (Chapter: 4) and assessment of quality (Chapter: 5)) were made.

6.5 Contributions of Research to Literature

The work presented in the initial study (Chapter 3) was presented at CVMP 2010 [2]. This sought to demonstrate that an approach which directly rendered the volumetric 'space carved' volume was possible without the need to generate tessellated geometric topology common in other approaches [100, 30, 58, 6, 27]. The rational for this approach to direct rendering of the reconstruction volume description was that this would offer greater potential for improved performance and scalability than evolutions of these common approaches towards the utilisation of GPU processing for the resolving of Visual Hulls [43].

The research undertaken for the GPU aligned approach (Chapter 4) was presented at CSCW 2011 [3]. This builds on principles of projective texturing [73] and multi-texturing [23] within a modified version of a plane based volume intersection algorithm [47, 49]. This sought to demonstrate that the approach taken could replicate the fidelity and speed performance of alternate approaches to directly capture and reconstruct an animated human form [80], within a free-viewpoint video system [28]. The key feature of this solution was a mapping of the entire pipeline process into the graphics rendering pipeline, thereby addressing issues of scalability and increased system cost common to other approaches which utilised computational clusters and/or GPU processing to accelerate sections of the reconstruction process within a more traditional sequential pipeline [56, 93, 102]. This final reconstruction approach also built on the principles of 'Floating Textures' [20] for weighting texture blending contributions, based on observer position, by applying them within the 3D reconstruction process itself, rather than using this approach to texture the 3D reconstruction surface.

Chapter 5 (Validating the Approach), presents work which utilises the advantages of a simulated capture/reconstruction environment in positioning, and matching the position of, many virtual cameras to develop an enhanced platform for objective analysis of the fidelity of 3D reconstruction. This builds on more constrained objective analysis, based on real world capture, [39] that sought to address concerns that single viewpoint 3D video analysis techniques, such as Peek-Signal-To-Noise and Video Quality Metric [22] were not ideally suited to measurement of quality of 3D reconstruction in image space [9]. This work is currently under review (2014) for acceptance in a special issue on Interactive Media Processing for Immersive Communication in the IEEE Journal on Selected Topics in Signal Processing. The work also contributes to a second complementary article, in the same special issue, reviewing approaches to 3D video based reconstruction.

Additional contributions to published works include articles in IEEE

Transactions in Visualisation and Computer Graphics [72] which described experimentation to assess how users perceived eye-gaze in 3D reconstruction. In this the experimental platform was based on a derivation of the simulation platform used in this research adapted to use both live capture data and pre-computed 3D reconstruction from other techniques. This was an extended study of work originally presented at IEEE VR 2009 [71]. The simulation platform for this research also provided an experimental and developmental framework for exploring the capture and synchronisation of image data suitable for 3D reconstruction presented at DS-RT 2010 [62].

6.6 Conclusion

The aim of this research was to improve upon the temporal and visual quality of 3D video based reconstruction through new approaches to developing, doing and measuring it. This started with an assumption: that it was unnecessary to enact the time consuming and resource heavy stages of surfacing and texturing to deliver high speed, high fidelity 3D video based reconstruction.

The work itself has generated three substantial contributions. The most obvious of these is the reconstruction approach itself; a hybrid approach combining the advantageous features of polyhedral and image based reconstruction to deliver a fast, effective and scalable approach that matches similar state of the art approaches. However, in achieving this two additional contributions were made; a robust simulation platform for exploring a 3D video based reconstruction that offers reliable, dependable experimentation, and an objective approach to iterative, and potentially comparative, evaluation of 3D reconstruction techniques.

In addition to these substantive contributions, there is also a methodological contribution in approach to the creation and evolution of 3D reconstruction technique. This utilises the facilities of the simulation framework and the modes of analysis it offers to enable iterative refinement of an evolving reconstruction approach in which evidence based selection can be used to guide technique development based on specific criteria.

By creating a framework for simulation of the capture and reconstruction pipeline repeatable, dependable experimentation could be performed. This created a flexible tool that extend the objective quality assessment approaches drawn from real world data capture [39], with dynamic virtual space configuration and the introduction of additional virtual camera systems. The features of this simulation tool enabled rapid experimentation with camera position, configuration and imaging parameters enabling many trials to be performed without the need for physical reconfiguration of devices and systems.

In the early stages of the simulation framework (Chapter: 3) the tools allowed rapid positioning of cameras based on navigation of a 3D scene, with the camera position recorded for multiple view points. The reconstruction method used at this time was a crude process that delivered a low resolution (blocky) form. However this was sufficient to review the impact of camera positioning on the reconstructed form, and to a certain extent highlighted the impact of poor camera positioning (Figure 3.4). From this it could be seen that an initial concept for camera configuration in which cameras were located at the vertices of a box surrounding the construction subject, analogous to the sort of configuration that might be used in a CAVE system, generated elongation of features (colloquially described as 'pointy head syndrome'). Positioning cameras around the subject in a uniform ring improved the reconstruction but still introduced horizontal linear artefacts that extended protuberances round the reconstruction. With the use of a simulated capture reconstruction pipeline it was possible to demonstrate a 'ideal' camera configuration in which a uniform ring was used, complemented with a single camera located under the chin of the subject to remove horizontal features on the face. This demonstrated that for the space carving approaches that this research used, practical real world capture configurations would always have some compromise due to the logistical challenges of positioning cameras against both a cluttered physical environment and dynamic subject.

With the re-development of the simulation framework in the mid stages of the research (Chapter: 4) a more precise approach to camera configuration was introduced in which the simulation environment could be controlled and manipulated in a manner more analogous to 3D modelling packages. This enabled precise positioning of cameras and individual definition of characteristics (image resolution, lens type etc) which were recordable and reusable. This allowed more detailed review of the form generation and also introduced the ability to import real-world camera configuration data drawn from a physical capture system. This both extended understanding of the impact of camera configuration on reconstruction and also enabled prototyping of the physical environment before implementing time consuming changes to the devices within the capture space.

The later stages of the research introduced objective measurement of the reconstruction form against 'truth' data drawn from an independent holistic capture process enacted by introducing a second set of virtual capture cameras into the simulated space. This process is only possible within a simulated capture environment (Chapter 5: Validating the Approach). Detailed analysis of data sets of image pairs (a source image and a reconstruction image) could then be used to generate maps detailing the profile of matching between source form and reconstruction from any viewpoint (Figures 5.6 & 5.7). These enabled objective review of the holistic nature of reconstruction against a constance source subject, thereby allowing direct comparison of multiple camera configurations.

The initial reconstruction approach implemented enacted reconstruction within the CPU processing units to deliver a volumetric texture which was evaluated and rendered within the graphics pipeline (Chapter: 3). This was based on an adaptation of previous work [4] which utilised medical volumetric rendering approaches [8] to display a pre calculated volume image.

Following attempts to apply surface texture to this form [16] and a realisation that multiple projected images could be evaluated against a 3D sample space (Chapter: 4) a more refined technique for reconstruction started to emerge. In this all of the core processing for the reconstruction and rendering was migrated to the graphics pipeline. CPU processes were then reduced to managing the sample space definition and delivery of image sets and configurations.

A further refinement of this process introduced view-point dependent weighting of images within the texturing process [20] which gave greater emphasis to the source image closest to the observation viewpoint, thereby providing a smooth blending of texture as the view-point moved round the reconstruction form. This imposed a small load on the CPU section of the approach in which the rendering viewpoint needed to be evaluated against the capture camera positions to generate a normalised weighting vector for each rendered frame.

Overall this reconstruction technique defines a highly componentised solution which can sit within a single system resources without requiring additional infrastructure in the form of processing clusters or computational load. As such the approach is relatively light-weight with little impact on the main CPU thereby lending itself to potential end user systems [3].

The use of a simulated capture and reconstruction pipeline has developed through the research undertaken. As the work developed these tools became more sophisticated as the understanding of the problem domain developed. With the final version of the simulation tools this enabled objective analyse of both end-to-end performance (speed) of reconstruction and rendering and fidelity (visual quality) of the reconstructed form against stable data sets drawn from a truth model. By using these complementary analysis techniques objective decision making over features to adopt during iterative refinement of technique could be made. This is demonstrated with a case study (Chapter: 5) in which two variations of space sampling approach are assessed. In this the review of the fidelity metrics show that there is little difference in the visual quality of the approaches, while one offers a measurable performance benefit. Adopting this approach therefore reduced the computational overhead of the approach, thereby freeing more processing cycles for additional features, without negative impact on the resultant reconstruction form. It has been argued that adequate approaches for robust subjective assessment of such free viewpoint video techniques is lacking and more needs to be done to define the quality influencing factors that might formulate a framework for evaluation [38].

Visual presentation of images of the reconstruction forms (Figures 4.2, 4.3, 4.16, & 5.3) show strong correspondence with those reported in published literature [87, 56, 102], which is a common mode of comparison in this field. Objective analysis of the reconstruction forms broadly agrees with the one other objective analysis technique identified [39].

The performance characteristics of the reconstruction approach developed were evaluated against differing input data configurations against a range of different graphics hardware systems (Chapter: 4). This sought to profile the performance of the technique with the aim of testing to see if the goal of scalability had been achieved. This showed that the relationship between graphics hardware performance and reconstruction/rendering cycle time was highly consistent, enabling a conclusion to be drawn that the technique will scale in performance. This is an important finding. Utilising the graphics hardware effectively by matching a technique for reconstruction to the parallelism of the architecture means that each screen pixel can be considered independently without cross referencing to other data segments. Therefore as graphics hardware continues to gain performance through increased parallelism the technique will gain speed and potentially consume less of the available resource thereby freeing up computational load for additional refinement. This is in contrast to approaches that seek performance gain without matching of process to architecture, typically those that attempt parallelism of sub processes within a sequential framework, such as traditional space carving/surfacing and epipolar approaches.

Kilner [39] presents a framework for objective quality assessment in free-viewpoint video production. In this dissatisfaction with subjective analysis and conventional pixel-wise evaluation is expressed, arguing that these measures do not reflect the quality of view synthesis as perceived by the user. In this model an evaluation of image pairs, based on a leave-one-out test, is used to measure reconstruction quality rather than fidelity, presenting measures in quality of reproduction of shape and appearance rather than PSNR and VQM. Berger [9], identifies this framework as being more suited to the measurement of quality of 3D reconstruction in image space, pointing out that this is only applicable to geometry based 3D reconstruction techniques of the type presented by the research.

6.6.1 Short Comings of the Research

This research has undertaken a highly technically focused approach to the problem of 3D video based reconstruction. It has taken an objective standpoint to the process of doing, developing and measuring techniques for 3D reconstruction that both develops understanding of the domain and enables targeted development of reconstruction technique. Against this it could be argued that the standard domain approach for user based subjective testing is under represented [18, 22] and that the direction the development of the reconstruction technique took differed from the common approaches at the time of starting the research [24, 44, 56, 60].

Rapid prototyping of the capture and reconstruction pipeline allowed investigation of camera positioning and its impact on reconstruction [2]. This allows exploration of the 3D reconstruction environment and enables identification of artefacts inherent in the reconstruction due to poor camera placement. This can inform camera positioning in a capture space. However, other approaches seek to define probabilistic models that analysis the camera configuration as suitable for achieving the reconstruction goal [34]. While the tools developed within this research enables exploration and development of understanding, they do not directly support explicit improvement to capture quality and thereby improved 3D reconstruction quality.

The reconstruction approach chosen [3] related to other approaches [13, 16, 44] but sought to gain efficiency through direct visualisation of the space carved volume without surfacing and tessellation. While the analysis of this approach shows good correlation with reported fidelity of alternate approaches [90, 57, 89], direct comparison has not been undertaken within the research presented here. Image texturing of the resultant form and the resolution of obscured regions has been addressed in the approach taken, with a texturing implementation that partially implements the 'floating textures' approach [20], but which only offers a basic demonstration of how the occlusion problem could be resolved (Chapter: 5).

The introduction of a second set of virtual cameras into the synthetic capture, and reconstruction, spaces has enabled direct objective comparison of the resultant reconstruction within a free viewpoint context. This is a unique approach that extends a previous approach in which a single viewpoint could be evaluated using real world data [39]. However, in recent years, several other approaches to object quality assessment have been presented which consider view assessment [11, 46] and the quality fo the reconstructed surface mesh[45, 95]. There are in addition to approaches which assess the quality of representation in a sequence of reconstruction images from a 3D delivery perspective [22]. Each of these approaches assess different features of the reconstruction and rendering with some addressing user perception and others assessing quality of aspects of the reconstruction and there is some debate over the relevance of the differing analysis techniques against reconstruction types [9].

During the course of the research the introduction of cheap, commodity depth cameras, in the form of Microsoft's Kinect device (Nov 2010) had a significant impact on approaches to 3D reconstruction of dynamic human forms [88, 97, 98]. The line of investigation was already well established at this point in time and the approaches to reconstruction, simulation and evaluation were reaching a stable state to start delivering results. However, this is a significant disruptive technology in domain and while an major influence on future developments, this has not been considered by the research enacted as the impact of these devices was only becoming apparent during the development of the research.

6.6.2 Future Research Directions

The processes for objective analysis presented here support the technical development of understanding and method. These have been used within a process steered by subject user testing, enacted by others in complementary research [72]. However, it should be possible to more tightly integrate the objective and subjective aspects of this. Objective analysis techniques, demonstrated in this research, give precise measures of the matching between reconstruction and 'truth' source object. This is useful for testing for absolute difference, but does not consider the user perception difference [11] or metrics derived from image quality. This would suggest that further research could be enacted to investigate the point at which measured difference becomes critical from a user perspective [95].

The platform for simulation of the capture/reconstruction pipeline that has supported this work, and enabled objective analysis of reconstruction quality, is a robust and reliable experimental platform. While the approaches for evaluation are similar to others [39], many alternate approaches to reconstruction quality evaluation exist that are both objective [11, 22, 9], and subjective [72, 38, 46]. What is evident from the reporting of research into similar reconstruction techniques is a lack of consistency in evaluation approaches. This makes critical, comparative evaluation of technique difficult and suggests a significant research opportunity in aligning and resolving the disparate evaluation techniques that would update existing understanding [74].

The reconstruction process developed by this research does show comparable performance to alternate techniques and has several potential advantages, specifically in terms of alignment with current hardware evolution and future scalability. However, several issues still remain, specifically with occlusion artefacts and texture blending. The analysis of the technique shows that future generations of graphics device will exceed the computational requirements for the current form of the approach, thereby releasing spare computational capacity to address these. Future research could then build on this work to further refine the approach.

References

[1] J. Allard, J. Franco, C. Menier, E. Boyer, and B. Raffin. The grimage platform: A mixed reality environment for interactions. In *Computer Vision Systems, ICVS '06. IEEE International Conference on*, pages 46–46, 2006. doi: 10.1109/ICVS.2006.59.

[2] R. Aspin and D. Roberts. An exploration of non-tessellated 3d space carving for real-time 3d reconstruction of a person through a simulated process. In *Visual Media Production (CVMP), 2010 Conference on*, pages 151–160, 2010. doi: 10.1109/CVMP.2010.26.

[3] R. Aspin and D. Roberts. A gpu based, projective multi-texturing approach to reconstructing the 3d human form for application in tele-presence. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, CSCW '11, pages 105–112, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0556-3. doi: 10.1145/1958824.1958840. URL `http://doi.acm.org/10.1145/1958824.1958840`.

[4] R. Aspin, M. Smith, M. A. Nazar, C. Hutchinson, and L. Funk. Medivol: A practical initial evaluation of refined, 3d, interactive volumetric

representations of soft tissue pathologies in virtual environments. In *Distributed Simulation and Real Time Applications, 2009. DS-RT '09. 13th IEEE/ACM International Symposium on*, pages 73–81, 2009. doi: 10.1109/DS-RT.2009.40.

[5] Rob Aspin, Matt Smith, Charles Hutchinson, and Len Funk. Medivol: An initial study into real-time, interactive 3d visualisation of soft tissue pathologies. In *12th 2008 IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications*, pages 103–110, 2008. doi: 10.1109/DS-RT.2008.9.

[6] C. Basso and A. Verri. Fitting 3d morphable models using implicit representations. *Journal of Virtual Reality and Broadcasting*, 4(18), 2007.

[7] B. Baumgart. Apolyhedron represenatation for computer vision. In *Proceedings of the national computer conference and exposition (AF-PIS75)*, May 1975. doi: 10.1145/1499949.1500071.

[8] A. Benassarou, E. Bittar, N. W. John, and L. Lucas. Mc slicing for volume rendering applications. *Lecture Notes in Computer Science,*, pages 314 – 321, 2005. doi: 10.1007/11428848_39.

[9] K. Berger, C. Lipski, C. Linz, A. Sellent, and M. Magnor. A ghosting artifact detector for interpolated image quality assessment. In *Consumer Electronics (ISCE), 2010 IEEE 14th International Symposium on*, pages 1 –6, June 2010. doi: 10.1109/ISCE.2010.5523697.

[10] OpenMP Architecture Review Board. Openmp application program interface. Technical report, OpenMP Architecture Review Board, 2013. URL `http://www.openmp.org/mp-documents/OpenMP4.0.0.pdf`.

[11] E. Bosc, R. Pepion, P. Le Callet, M. Koppel, P. Ndjiki-Nya, M. Pressigout, , and L. Morin. Towards a new quality metric for 3-d synthesized view assessment. *Selected Topics in Signal Processing, IEEE Journal of*, 5(7):1332–1343, 2011.

[12] T. K. Capin, P. I. Sunday, H. Noser, N. M. Thalmann, and D. Thalmann. Virtual human representation and communication in vlnet. *IEEE Computer Graphics and Applications*, 17:42–53, 1997. doi: 10.1109/38.574680.

[13] G. K. M. Cheung, T. Kanade, J. Y. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *Computer Vision and Pattern Recognition, Proceedings. IEEE Conference on*, volume 2, pages 714–720 vol.2, 2000.

[14] C Cruz-Neira, D Sandin, T DeFanti, R Kenyon, and J Hart. The cave: Audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6):64–72, 1992. doi: 10.1145/129888.129892.

[15] C Cruz-Neira, D Sandin, and T DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *SIGGRAPH'93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, pages 135–142, 1993.

[16] P. E. Debevec, G. Borshukov, and Y. Yizhou. Efficient view-dependent image-based rendering with projective texture-mapping. In *9th Eurographics Rendering Workshop*, pages 105–116. Eurographics, Eurographics Association, June 1998. doi: 10.1007/978-3-7091-6453-2-10.

[17] Digita. Qt - cross platform application and ui framework, 2013. URL `http://qt.digia.com/`.

[18] L. Do, S. Zinger, Y. Morvan, and P. H. N. de With. Quality improving techniques in dibr for free-viewpoint video. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2009*, pages 1 –4, may 2009. doi: 10.1109/3DTV.2009.5069627.

[19] T. Duckworth. *Improving the performance of video based reconstruction and validating it within a Telepresence context*. PhD thesis, University of Salford, May 2013. URL `http://usir.salford.ac.uk/29219/`. University of Salford PHd Thesis.

[20] M. Eisemann, B. De Decker, M. Magnor, P.Bekaert, E.de Aguiar, C. Theobalt, and A.Sellent. Floating textures. *Computer Graphics Forum*, 27(2):409–418, 2008. doi: 10.1111/j.1467-8659.2008.01138.x.

[21] P. Eisert, C. Jacquemin, and A. Hilsmann. Virtual jewel rendering for augmented reality environments. In *Image Processing (ICIP), 17th IEEE International Conference on*, pages 1813–1816, 2010. doi: 10.1109/ICIP.2010.5652299.

[22] E. Ekmekcioglu, S. Worrall, D. De Silva, A. Fernando, and A. M. Kondoz. Depth based perceptual quality assessment for synthesized camera

viewpoints. In *Second International Conference on User Centric Media*, 2010. doi: 10.1007/978-3-642-35145-7_10.

[23] C. Everitt. Projective texture mapping. Technical report, nVidia, 2001. URL `http://developer.nvidia.com/object/Projective_Texture_Mapping.html`.

[24] J. S. Franco and E. Boyer. Efficient polyhedral modeling from silhouettes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(3):414–427, 2009. doi: 10.1109/TPAMI.2008.104.

[25] O. Grau, T. Pullen, and G.A. Thomas. A combined studio production system for 3d capturing of live action and immersive actor feedback. *IEEE Transactions on Circuits and Systems for Video Technology*, 14: 370–380, 2004. doi: 10.1109/TCSVT.2004.823397.

[26] O. Grau, R. Koch, F. Lavagetto, A. Sarti, S. Tubaro, and J. Woetzel. The origami project: Advanced tools for creating and mixing real and virtual content in film and tv production. *Vision, Image and Signal Processing, IEE Proceedings* -, 152(4):454–469, 2005. doi: 10.1049/ip-vis:20045134.

[27] O. Grau, A. Hilton, J. Kilner, G. Miller, T. Sargeant, and J. Starck. A free-viewpoint video system for visualisation of sport scenes. Technical report, BBC, 2006. URL `http://www.bbc.co.uk/rd/publications/whitepaper142`.

[28] O. Grau, G. A. Thomas, A. Hilton, J. Kilner, and J. Starck. A robust

free-viewpoint video system for sport scenes. In *3DTV Conference, 2007*, 2007. doi: 10.1109/3DTV.2007.4379384.

[29] M. Gross, S. Warmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. Van Gool, S. Lang, K. Strehlke, A. V. M. Oliver, E. Zarich, and O. Staadt. blue-c: A spatially immersive display and 3d video portal for telepresence. In *Proceedings of ACM SIGGRAPH 2003*, volume 22 of *ACM Transactions on Graphics*, pages 819–827, 2003. doi: 10.1145/882262.882350.

[30] J. Y. Guillemaut, A. Hilton, J. Starck, J. Kilner, and O. Grau. A bayesian framework for simultaneous matting and 3d reconstruction. In *3-D Digital Imaging and Modeling, 3DIM '07. Sixth International Conference on*, pages 167–176, 2007. doi: 10.1109/3DIM.2007.3.

[31] Ilona Heldal, Anthony Steed, Maria Spante, Ralph Schroeder, Sophia Bengtsson, and Marja Partanen. Successes and failures in co-present situations. *Presence: Teleoper. Virtual Environ.*, 14(5):563–579, October 2005. ISSN 1054-7460. doi: 10.1162/105474605774918679. URL http://dx.doi.org/10.1162/105474605774918679.

[32] A. Hilton, D. Beresford, T. Gentils, R. Smith, and Sun Wei. Virtual people: capturing human models to populate virtual worlds. In *Computer Animation, 1999. Proceedings*, pages 174–185, 1999. doi: 10.1109/CA.1999.781210.

[33] J. Hindmarsh, M. Fraser, C. Heath, S Benford, and C. Greenhalgh. Fragmented interaction: establishing mutual orientation in virtual en-

vironments. In *CSCW '98 Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 217–226. ACM, 1998. doi: 10.1145/289444.289496.

[34] E. Holec and N. Papanikolopoulos. A probabilistic quality metric for camera placement in 3d reconstructions. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5899 – 5904, 2011. doi: 10.1109/ICRA.2011.5980520.

[35] Inria. 4d repository. Electronic Repository, 2007. URL `http://4drepository.inrialpes.fr/public/datasets`.

[36] S Izadi, D Kim, O Hilliges, D Molyneaux, R Newcombe, P Kohli, J Shotton, S Hodges, D Freeman, and A. Davison. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM Press, 2011.

[37] F. Kelly and A. Kokaram. The gpu in the broadcast media production chain: A disruptive technology? In *Visual Media Production, 2005. CVMP 2005. The 2nd IEE European Conference on*, pages 275–284, 2005. doi: 0-86341-583-0.

[38] Sara Kepplinger. Subjective quality assessment of free viewpoint video objects. In *Adjunct Proceedings of EuroITV 2011, June 29th - July 1st 2011, Lisbon, Portugal, Doctoral Consortium, p. 47*, 2011.

[39] J. Kilner, J. Starck, JY Guillemaut, and A. Hilton. Objective quality assessment in free-viewpoint video production. In *3DTV*

Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2008, pages 225 – 228. Elsevier, 2008. doi: 10.1109/3DTV.2008.4547849.

[40] A. Kratz, M. Hadwiger, R. Splechtna, A. Fuhrmann, and K. Buhler. Gpu-based high-quality volume rendering for virtual environments. In *Proceedings of AMI-ARCS.*, 2006.

[41] M. W. Krueger, T. Gionfriddo, and K. Hinrichsen. Videoplace: an artificial reality. In *CHI '85 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 35–40, 1985. doi: 10.1145/317456.317463.

[42] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38:307–314, 1998. doi: 10.1109/ICCV.1999.791235.

[43] A. Ladikos, S. Benhimane, and N. Navab. Efficient visual hull computation for real-time 3d reconstruction using cuda. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–8, 2008. doi: 10.1109/CVPRW.2008.4563098.

[44] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(2):150–162, 1994. doi: 10.1109/34.273735.

[45] G. Lavou. A multiscale metric for 3d mesh visual quality assessment. *Computer Graphics Forum*, 30(5):1427–1437, 2011.

[46] J. Lee, L. Goldmann, and T. Ebrahimi. Paired comparison-based subjective quality assessment of stereoscopic images. *Multimedia Tools and Applications*, 2012.

[47] S. Lee, I. Kim, S. C. Ahn, H. Ko, M. Lim, and H. Kim. Real time 3d avatar for interactive mixed reality. In *VRCAI '04 Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*. ACM, 2004. doi: 10.1145/1044588.1044602. 1044602 75-80.

[48] S. Lee, I. Kim, S. C. Ahn, H. Ko, M. Lim, and H. Kim. Toward immersive telecommunication: 3d video avatar with physical interaction. In *ICAT '05 Proceedings of the 2005 international conference on Augmented tele-existence*. ACM, 2005. doi: 10.1145/1152399.1152411. 1152411 56-61.

[49] Z. Lee, D. B. Sodee, M. Resnick, and G. T. Maclennan. Multimodal and three-dimensional imaging of prostate cancer. *Comput Med Imaging Graph*, 29(6):477–86, 2005. doi: 10.1016/j.compmedimag.2005.01.004.

[50] M. Li, M. Magnor, and H. Seidel. A hybrid hardware-accelerated algorithm for high quality rendering of visual hulls. In *GI '04 Proceedings of the 2004 Graphics Interface Conference*, pages 41–48, 2004. doi: 1-56881-227-2.

[51] C. Liao, Z. Liu, L. Huang, and B. Chapman. Evaluating openmp on chip multithreading platforms. In *IWOMP'05/IWOMP'06 Proceedings*

of the 2005 and 2006 international conference on OpenMP shared memory parallel programming, 2005. doi: 3-540-68554-5 978-3-540-68554-8.

[52] C. Lu. K3 moore's law in the era of gpu computing. In *VLSI Design Automation and Test (VLSI-DAT), 2010 International Symposium on*, pages 5–5, 2010. doi: 10.1109/VDAT.2010.5496638.

[53] S Mann and R. Picard. Being undigital with digital cameras: Extending dynamic range by combining differently exposed pictures. In *IS&T 48th annual conference*, pages 422–428, 1995.

[54] S Mann, R Lo, J Huang, V Rampersad, and R. Janzen. Hdrchitecture: real-ttime setereoscopic hdr imaging for extreme dynamic range. In *ACM SIGGRAPH 2012, Emerging Technologies*, page 11. ACM, 2012.

[55] T. Matsuyama. Exploitation of 3d video technologies. In *Informatics Research for Development of Knowledge Society Infrastructure, 2004. ICKS 2004. International Conference on*, pages 7–14, 2004. doi: 10.1109/ICKS.2004.1313403.

[56] T. Matsuyama, Wu Xiaojun, T. Takai, and T. Wada. Real-time dynamic 3-d object shape reconstruction and high-fidelity texture mapping for 3-d video. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(3):357–369, 2004. doi: 10.1109/TCSVT.2004.823396.

[57] T. Matsuyama, S. Nobuhara, T. Takai, and T. Tung. 3d surface texture generation. In *3D Video and Its Applications*, pages 151–194. Springer London, 2012. ISBN 978-1-4471-4119-8.

[58] W. Matusik, C. Buehler, R. Raskar, S.J. Gortler, and L McMillan. Image-based visual hulls. In *SIGGRAPH'00: Computer Graphics and Interactive Techniques*, pages 369–374. ACM/Addison-wesley, 2000.

[59] Wojciech Matusik. *Image-Based Visual Hulls*. PhD thesis, 1997.

[60] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J. Gortler, and Leonard McMillan. Image-based visual hulls, 2000.

[61] C. Moore. *Data acquisition and processing for 3D video based reconstruction of humans*. PhD thesis, University of Salford, 2013. University of Salford PHd Thesis.

[62] Carl Moore, Toby Duckworth, Rob Aspin, and Dave Roberts. Synchronization of images from multiple cameras to recnostruct a moving human, October 17-20, 2010 2010.

[63] F. Multon, R. Kulpa, and B. Bideau. Mkm: A global framework for animating humans in virtual reality applications. *Presence*, 17(1):17–28, 2008. doi: 10.1162/pres.17.1.17.

[64] J. O'Hare. Facilities - octave system. Technical Report July, University of Salford, 2009. URL `http://www.cve.salford.ac.uk/page/octave`.

[65] Robert Oldfield. *The Analysis and Imporvment of Focused Source Reporduction with Wave Field Synthesis*. PhD thesis, Univesity of Salford, School of Computing, Science and Engineering, UK, 2013.

[66] N. Orman, H. Kim, R. Sakamoto, T. Toriyama, K. Kogure, and R. Lindeman. Gpu-based optimization of a free-viewpoint video system. In *I3D '08 Proceedings of the 2008 symposium on Interactive 3D graphics and games*, volume 7, pages 120–133, 2008. doi: 10.1145/1342250.1357027.

[67] R. Osfield. Osg: Openscenegraph, 2013. URL `http://www.openscenegraph.org/`.

[68] Oyewole Oyekoya, William Steptoe, and Anthony Steed. Sphereavatar: A situated display to represent a remote collaborator. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2551–2560, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208642. URL `http://doi.acm.org/10.1145/2207676.2208642`.

[69] L. Quante, B. Muhlbach. Eye-contact in multipoint videoconferencing. In *Proceedings of the 17th International Symposium on Human Factors in Telecommunications*, pages 123–131, 1999.

[70] V. Ramesh, R. L. Glass, and I. Vessey. Research in computer science: an empirical study. *Journal of Systems and Software*, 70:165 – 176, 2004. ISSN 0164-1212. doi: 10.1016/S0164-1212(03)00015-3.

[71] D. Roberts, R. Wolff, J. Rae, A. Steed, R. Aspin, M. McIntyre, A. Pena, O. Oyekoya, and W. Steptoe. Communicating eye-gaze across a distance: Comparing an eye-gaze enabled immersive collaborative virtual environment, aligned video conferencing, and being together. In *Virtual*

Reality Conference, 2009. VR 2009. IEEE. IEEE Computer Society, 2009. doi: 10.1109/VR.2009.4811013.

[72] D.J. Roberts, J. Rae, T.W. Duckworth, C.M. Moore, and R. Aspin. Estimating the gaze of a virtuality human. *Visualization and Computer Graphics, IEEE Transactions on*, 19(4):681–690, 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.30.

[73] R. Rodrigues and A. R. Fernandes. Accelerated epipolar geometry computation for 3d reconstruction using projective texturing. In *SCCG '04 Proceedings of the 20th spring conference on Computer graphics*, pages 200–206. ACM, 2004. doi: 10.1145/1037210.1037241.

[74] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 519 – 528, June 2006. doi: 10.1109/CVPR.2006.19.

[75] J. Seyama and R. S. Nagayama. The uncanny valley: Effect of realism on the impression of artificial human faces. *Presence*, 16(4):337–351, 2007. doi: 10.1162/pres.16.4.337.

[76] K. Shah, R. K. Harrell, K. W. Erion, and R. Kumar. System and method for single action initiation of a video conference. patent: Us 7532232 b2, 2009.

[77] R. Singh and A. K. Singh. Transcending from virtual reality into tele-

immersive technologies and applications: A perspective. *Ubiquity*, 2008 (June):3–3, 2008. doi: 10.1145/1399612.1399615. 1399615.

[78] Roy Soliman, Robin Braun, and Simeon Simoff. The essential ingredients of collaboration. In *Proceedings of the 2005 International Conference on Collaborative Technologies and Systems*, CTS'05, pages 366–373, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2387-0, 978-0-7695-2387-3. URL `http://dl.acm.org/citation.cfm?id=1895420.1895483`.

[79] C. Spagno and A. Kunz. Construction of a three-sided immersive telecollaboration system. *Presence: Teleoperators and Virtual Environments - Special issue: IEEE VR*, pages 22–26, 2003 2003. doi: 10.1162/1054746041382410.

[80] J. Starck and A. Hilton. Model-based human shape reconstruction from multiple views. *Computer Vision and Image Understanding*, 111 (2):179–194, 2008. doi: 10.1016/j.cviu.2007.10.001.

[81] A. Steed, D. Roberts, and R. Schroeder. Interaction between users of immersion projection technology systems. In *11th International Conference on Human Computer Interaction*, 2005.

[82] A. Steed, W. Steptoe, W. Oyekoya, F. Pece, T. Weyrich, J. Kautz, D. Friedman, A. Peer, M. Solazzi, F. Tecchia, M. Bergamasco, and M. Slater. Beaming: An asymmetric telepresence system. *Computer Graphics and Applications, IEEE*, 32(6):10–17, Nov 2012. ISSN 0272-1716. doi: 10.1109/MCG.2012.110.

[83] W. Stephan, L. Edouard, and G. Markus. 3d video fragments: Dynamic point samples for real-time free-viewpoint video. *Computers and Graphics, Special Issue on Coding, Compression and Streaming Techniques for 3D and Multimedia Data*, 28(1):3–14, 2004. doi: 10.1016/j.cag.2003.10.015.

[84] W. Steptoe, O. Oyekoya, A. Murgia, R. Wolff, J. Rae, E. Guimaraes, D. Roberts, and A. Steed. Eye tracking for avatar eye gaze control during object-focused multiparty interaction in immersive collaborative virtual environments. In *Virtual Reality Conference, 2009. VR 2009. IEEE*, pages 83–90, 2009. doi: 10.1109/VR.2009.4811003.

[85] W. Steptoe, A. Steed, and M. Slater. Human tails: Ownership and control of extended humanoid avatars. *Visualization and Computer Graphics, IEEE Transactions on*, 19(4):583–590, April 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.32.

[86] William Steptoe, Jean-Marie Normand, Oyewole Oyekoya, Fabrizio Pece, Elias Giannopoulos, Franco Tecchia, Anthony Steed, Tim Weyrich, Jan Kautz, and Mel Slater. Acting rehearsal in collaborative multimodal mixed reality environments. *Presence: Teleoperators and Virtual Environments*, 21(4):406–422, November 2012. ISSN 1054-7460. doi: 10.1162.00109.

[87] P. Stroia-Williams, A. Hilton, and O Grau. Example-based reflectance estimation for capturing relightable models of people. In *Visual Media*

Production (CVMP), 5th European Conference on, pages 1–10, 2008. doi: 978-0-86341-973-7.

[88] Q. Sun, Y. Tang, P. Hu, and J. Peng. Kinect-based automatic 3d high-resolution face modeling. In *Image Analysis and Signal Processing (IASP), 2012 International Conference on*, pages 1–4, 2012. doi: 10.1109/IASP.2012.6425065.

[89] W. Sun, L. Xu, O.C. Au, S.H. Chui, and C.W. Kwok. An overview of free viewpoint depth-image-based rendering (dibr). In *APSIPA ASC 2010 - Asia-Pacific Signal and Information Processing Association Annual Summit and Conference.*, pages p. 1023–1030, 2010.

[90] Y. Tao, Z. Yanning, L. Meng, S. Dapei, and Z. Xingong. A multi-camera network system for markerless 3d human body voxel reconstruction. In *Image and Graphics, ICIG '09. Fifth International Conference on*, pages 706–711, 2009. doi: 10.1109/ICIG.2009.89.

[91] C. Tay and R. Green. Human motion capture and representation 3-d avatar interaction. In *Image and Vision Computing New Zealand, 2009. IVCNZ '09. 24th International Conference*, pages 209–214, 2009. doi: 10.1109/IVCNZ.2009.5378411.

[92] M Tocci, C Kiser, N Tocci, and P. Sen. A versatile hdr video production system. *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2011)*, 30(4), 2011.

[93] M. Ueda, D. Arita, and R. Taniguchi. Real-time free-viewpoint video generation using mutilpe cameras and a pc-cluster. *Advances in Multi-*

media Information Processing - PCM 2004, 3331:418–425, 2004 2004. doi: 10.1007/978-3-540-30541-5_52.

[94] V. Vinayagamoorthy, A. Steed, and M. Slater. The impact of a character posture model on the communication of affect in an immersive virtual environment. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):965–982, 2008. doi: 10.1109/TVCG.2008.62. 1446250.

[95] L. Vsa and J. Rus. Dihedral angle mesh error: a fast perception correlated distortion measure for fixed connectivity triangle meshes. *Computer Graphics Forum*, 31(5), 2012.

[96] J. Wainer, C. G. Novoa Barsottini, D. Lacerda, L. Rodrigues, and M. de Marco. Empirical evaluation in computer science research published by {ACM}. *Information and Software Technology*, 51(6):1081 – 1085, 2009. ISSN 0950-5849. doi: 10.1016/j.infsof.2009.01.002.

[97] Y. Wan, J. Wang, J. Hu, T. Song, Y. Bai, and Z. Ji. A study in 3d-reconstruction using kinect sensor. In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on*, pages 1–7, 2012. doi: 10.1109/WiCOM.2012.6478374.

[98] J. Weidi, Y. Won-Jae, J. Saniie, and E. Oruklu. 3d image reconstruction and human body tracking using stereo vision and kinect technology. In *Electro/Information Technology (EIT), 2012 IEEE International Conference on*, pages 1–4, 2012. doi: 10.1109/EIT.2012.6220732.

[99] I. M. Williams. A summary of recent gpu developments and key enabling technologies for digital media applications. In *Visual Media*

Production, 2005. CVMP 2005. The 2nd IEE European Conference on, pages 251–254, 2005. doi: 0-86341-583-0.

[100] P. D. Williams and A. Hilton. 3d reconstruction using spherical images. In *3rd European Conference on Visual Media Production*, pages 179–179, 29-30 Nov 2006 2006.

[101] W. Wu, A. Arefin, R. Rivas, K. Nahrstedt, R. Sheppard, and Z. Yang. Quality of experience in distributed interactive multimedia environments: toward a theoretical framework. In *Proceedings of the 17th ACM international conference on Multimedia*, MM '09, pages 481–490, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-608-3. doi: 10.1145/1631272.1631338.

[102] X. Wu, O. Takizawa, and T. Matsuyama. Parallel pipeline volume intersection for real-time 3d shape reconstruction on a pc cluster. In *Computer Vision Systems, 2006 ICVS '06. IEEE International Conference on*. IEEE Computer Society, 2006. doi: 10.1109/ICVS.2006.49.

[103] Z. Yue, L. Zhao, and R. Chellappa. View synthesis of articulating humans using visual hull. *Proceedings of the 2003 International Conference on Multimedia and Expo*, 2:489 – 492, 2003. doi: 10.1109/ICME.2003.1220961.

Appendix A

First Generation: Code samples

A.1 Volume Rendering: GLSL Vertex Shader Code

```
varying vec4 diffuse0, diffuse1, diffuse2, diffuse3, ambient;
varying vec3 lightDir0;
varying vec3 lightDir1;
varying vec3 lightDir2;
varying vec3 lightDir3;
varying vec3 vStep;

void main()
{
    vec4 v;
    vStep[0]=5.0/512.0;
```

```
vStep[1]=5.0/512.0;
vStep[2]=5.0/223.0;

v = vec4(−10.0, −5.0f, −20.0, 1.0);
lightDir0=normalize(vec3(v[0], v[1], v[2]));

v = gl_ModelViewMatrix*gl_LightSource[1].position;
lightDir1=normalize(vec3(v[0], v[1], v[2]));

v = gl_ModelViewMatrix*gl_LightSource[2].position;
lightDir2=normalize(vec3(v[0], v[1], v[2]));

v = gl_ModelViewMatrix*gl_LightSource[3].position;
lightDir3=normalize(vec3(v[0], v[1], v[2]));

diffuse0 = gl_FrontMaterial.diffuse *
    gl_LightSource[0].diffuse;
diffuse1 = gl_FrontMaterial.diffuse *
    gl_LightSource[1].diffuse;
diffuse2 = gl_FrontMaterial.diffuse *
    gl_LightSource[2].diffuse;
diffuse3 = gl_FrontMaterial.diffuse *
    gl_LightSource[3].diffuse;
    ambient = gl_LightModel.ambient *
        gl_FrontMaterial.ambient;
    ambient += gl_FrontMaterial.ambient *
        gl_LightSource[0].ambient;
    ambient += gl_FrontMaterial.ambient *
        gl_LightSource[1].ambient;
    ambient += gl_FrontMaterial.ambient *
        gl_LightSource[2].ambient;
    ambient += gl_FrontMaterial.ambient *
        gl_LightSource[3].ambient;

    gl_TexCoord[0] = gl_MultiTexCoord0;
    gl_Position = ftransform();
}
```

A.2 Volume Rendering: GLSL Fragment Shader Code

```
varying vec4 diffuse0, diffuse1, diffuse2, diffuse3, ambient;
varying vec3 lightDir0;
varying vec3 lightDir1;
varying vec3 lightDir2;
varying vec3 lightDir3;
uniform sampler3D tex;
```

```
varying vec3 vNormal;
varying vec3 vStep;

bool isEdge2(int iChannel, float fMin)
{
    bool bEdge=false;

    vNormal=vec3(0.0, 0.0, 0.0);
    vec3 vPos;
    vec4 vCol;

    vPos=vec3(-vStep[0], -vStep[1], -vStep[2]);
    vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
    if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

    vPos=vec3(-vStep[0], 0.0, -vStep[2]);
    vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
    if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

    vPos=vec3(-vStep[0], vStep[1], -vStep[2]);
    vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
    if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

    vPos=vec3(-vStep[0], -vStep[1], 0.0);
```

```
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(-vStep[0], 0.0, 0.0);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(-vStep[0], vStep[1], 0.0);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(-vStep[0], -vStep[1], vStep[2]);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(-vStep[0], 0.0, vStep[2]);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(-vStep[0], vStep[1], vStep[2]);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(vStep[0], -vStep[1], -vStep[2]);
```

```
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(vStep[0], 0.0, -vStep[2]);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(vStep[0], vStep[1], -vStep[2]);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(vStep[0], -vStep[1], 0.0);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(vStep[0], 0.0, 0.0);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(vStep[0], vStep[1], 0.0);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(vStep[0], -vStep[1], vStep[2]);
```

```
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(vStep[0], 0.0, vStep[2]);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(vStep[0], vStep[1], vStep[2]);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(0.0, -vStep[1], -vStep[2]);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(0.0, 0.0, -vStep[2]);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(0.0, vStep[1], -vStep[2]);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(0.0, -vStep[1], 0.0);
```

```
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(vStep[0], 0.0, 0.0);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(0.0, vStep[1], 0.0);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(0.0, -vStep[1], vStep[2]);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(0.0, 0.0, vStep[2]);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

vPos=vec3(0.0, vStep[1], vStep[2]);
vCol=texture3D(tex,vec3(gl_TexCoord[0])+vPos);
if(vCol[iChannel]>fMin) vNormal+=vPos; else bEdge=true;

if(bEdge) vNormal=normalize(vNormal);
```

```
    return bEdge;
}

void main()
{
    vec4 vColour=texture3D(tex, vec3(gl_TexCoord[0]));

    if(vColour[3]>0.001)
    {
        vColour = ambient;

        if(isEdge2(3, 0.1))
        {
            float NdotL = max(dot(vNormal,lightDir0),0.0);

            if(NdotL > 0.0)
                vColour += diffuse0*dot(lightDir0, vNormal);

            NdotL = max(dot(vNormal,lightDir1),0.0);

            if(NdotL > 0.0)
                vColour += diffuse1*dot(lightDir1, vNormal);
            NdotL = max(dot(vNormal,lightDir2),0.0);

            if(NdotL > 0.0)
                vColour += diffuse2*dot(lightDir2, vNormal);
        }

        gl_FragColor = vColour;
    }
    else
        discard;
}
```

Appendix B

Third Generation: Code samples

B.1 Slice Plane Sampling GLSL Code

B.1.1 Slice Plane Sampling Vertex Shader

```
uniform mat4 osg_ViewMatrixInverse;
varying vec4 vPos;
void main()
{
    vPos=osg_ViewMatrixInverse*gl_ModelViewMatrix*gl_Vertex;

    gl_TexCoord[0]=gl_TextureMatrix[0]*vPos;
    gl_TexCoord[1]=gl_TextureMatrix[1]*vPos;
    gl_TexCoord[2]=gl_TextureMatrix[2]*vPos;
    gl_TexCoord[3]=gl_TextureMatrix[3]*vPos;
    gl_TexCoord[4]=gl_TextureMatrix[4]*vPos;
    gl_TexCoord[5]=gl_TextureMatrix[5]*vPos;
    gl_TexCoord[6]=gl_TextureMatrix[6]*vPos;
    gl_TexCoord[7]=gl_TextureMatrix[7]*vPos;

    gl_Position=ftransform();
}
```

B.1.2 Slice Plane Sampling Fragment Shader

```
uniform sampler2D CamTex0;
uniform sampler2D CamTex1;
uniform sampler2D CamTex2;
uniform sampler2D CamTex3;
uniform sampler2D CamTex4;
uniform sampler2D CamTex5;
uniform sampler2D CamTex6;
uniform sampler2D CamTex7;

uniform vec4 vCamPos0;
uniform vec4 vCamPos1;
uniform vec4 vCamPos2;
uniform vec4 vCamPos3;
uniform vec4 vCamPos4;
uniform vec4 vCamPos5;
uniform vec4 vCamPos6;
uniform vec4 vCamPos7;

uniform mat4 camWeights;
uniform int NumCams;
varying vec4 vPos;

vec4 vCols[8];
float fWeights[8];
vec4 avCamPos[8];
int iMaxWeight;
int iSecWeight;

int aiActiveWeights[8];
int iNumActiveWeights=0;

vec4 texLookup(int i, vec4 vCoord)
{
   switch(i)
   {
      case 0: return textureProj(CamTex0, vCoord);
      case 1: return textureProj(CamTex1, vCoord);
      case 2: return textureProj(CamTex2, vCoord);
      case 3: return textureProj(CamTex3, vCoord);
```

```
        case 4: return textureProj(CamTex4, vCoord);
        case 5: return textureProj(CamTex5, vCoord);
        case 6: return textureProj(CamTex6, vCoord);
        case 7: return textureProj(CamTex7, vCoord);
    }
    return vec4(0.0, 0.0, 0.0, 0.0);
}

float weightLookup(int i)
{
    switch(i)
    {
        case 0: return camWeights[0][0];
        case 1: return camWeights[0][1];
        case 2: return camWeights[0][2];
        case 3: return camWeights[0][3];
        case 4: return camWeights[1][0];
        case 5: return camWeights[1][1];
        case 6: return camWeights[1][2];
        case 7: return camWeights[1][3];
    }
    return 0.0;
}
```

```
vec4 camLookup(int i)
{
    switch(i)
    {
        case 0: return vCamPos0;
        case 1: return vCamPos1;
        case 2: return vCamPos2;
        case 3: return vCamPos3;
        case 4: return vCamPos4;
        case 5: return vCamPos5;
        case 6: return vCamPos6;
        case 7: return vCamPos7;
    }
    return vec4(0.0, 0.0, 0.0, 0.0);
}

float checkOccupancy()
{
    float fAlpha=1.0;
    for(int i=0;i<NumCams;i++)
    {
        vCols[i]=texLookup(i, gl_TexCoord[i]);
        fAlpha*=vCols[i][3];
    }
    return fAlpha;
}

vec4 calcColWeighted()
{
    vec4 v=vec4(0.0, 0.0, 0.0, 0.0);
    for(int i=0;i<iNumActiveWeights;i++)
    {
        v+=vCols[aiActiveWeights[i]]*
                weightLookup(aiActiveWeights[i]);
    }
    v[3]=1.0;
    return v;
}

void checkWeights()
{
    iNumActiveWeights=0;
    for(int i=0;i<NumCams;i++)
        if(fWeights[i]>0.0)
            aiActiveWeights[iNumActiveWeights++]=i;
}

void normWeights()
```

```
{
    float fLen=0.0;

    for(int i=0;i<iNumActiveWeights;i++)
        fLen+=fWeights[aiActiveWeights[i]]*
                fWeights[aiActiveWeights[i]];

    fLen=sqrt(fLen);

    for(int i=0;i<iNumActiveWeights;i++)
        fWeights[aiActiveWeights[i]]/=fLen;
}

void main()
{
    gl_FragColor=vec4(0.0,0.0,0.0,1.0);

    if(occupancyTest(gl_Position)<0.5)
    {
        discard;
        return;
    }

    checkWeights();
    normWeights();
    gl_FragColor=calcColWeighted();
}
```

B.1.3 Slice Plane Sampling Fragment Shader with Occlusion Testing

```
uniform sampler2D CamTex0;
uniform sampler2D CamTex1;
uniform sampler2D CamTex2;
uniform sampler2D CamTex3;
uniform sampler2D CamTex4;
uniform sampler2D CamTex5;
uniform sampler2D CamTex6;
uniform sampler2D CamTex7;

uniform vec4 vCamPos0;
uniform vec4 vCamPos1;
uniform vec4 vCamPos2;
uniform vec4 vCamPos3;
uniform vec4 vCamPos4;
uniform vec4 vCamPos5;
uniform vec4 vCamPos6;
uniform vec4 vCamPos7;
uniform mat4 camWeights;
uniform int NumCams;
varying vec4 vPos;
vec4 vCols[8];
float fWeights[8];
vec4 avCamPos[8];

int aiActiveWeights[8];
int iNumActiveWeights=0;

vec4 texLookup(int i, vec4 vCoord)
{
   switch(i)
   {
      case 0: return textureProj(CamTex0, vCoord);
      case 1: return textureProj(CamTex1, vCoord);
      case 2: return textureProj(CamTex2, vCoord);
      case 3: return textureProj(CamTex3, vCoord);
      case 4: return textureProj(CamTex4, vCoord);
      case 5: return textureProj(CamTex5, vCoord);
      case 6: return textureProj(CamTex6, vCoord);
      case 7: return textureProj(CamTex7, vCoord);
   }
   return vec4(0.0, 0.0, 0.0, 0.0);
```

```
}

float weightLookup(int i)
{
   switch(i)
   {
      case 0: return camWeights[0][0];
      case 1: return camWeights[0][1];
      case 2: return camWeights[0][2];
      case 3: return camWeights[0][3];
      case 4: return camWeights[1][0];
      case 5: return camWeights[1][1];
      case 6: return camWeights[1][2];
      case 7: return camWeights[1][3];
   }
   return 0.0;
}

vec4 camLookup(int i)
{
   switch(i)
   {
      case 0: return vCamPos0;
      case 1: return vCamPos1;
```

```
        case 2: return vCamPos2;
        case 3: return vCamPos3;
        case 4: return vCamPos4;
        case 5: return vCamPos5;
        case 6: return vCamPos6;
        case 7: return vCamPos7;
    }
    return vec4(0.0, 0.0, 0.0, 0.0);
}

float checkOccupancy()
{
    float fAlpha=1.0;
    for(int i=0;i<NumCams;i++)
    {
        vCols[i]=texLookup(i, gl_TexCoord[i]);
        fAlpha*=vCols[i][3];
        vCols[i][3]=1.0;
    }
    return fAlpha;
}

void checkWeights()
{
    iNumActiveWeights=0;
    for(int i=0;i<NumCams;i++)
        if(fWeights[i]>0.0)
          aiActiveWeights[iNumActiveWeights++]=i;
}

bool occTest(vec4 vStart, vec4 vEnd,
             float fStepSize, float fLen, float fStart)
{
    float fPos=fStart;
    vec4 vDir=normalize(vEnd-vStart);
    vec4 vPos;
    float fAlpha=1.0;

    while(fPos<fLen)
    {
        vPos=vStart+(vDir*fPos);

        for(int i=0;i<NumCams;i++)
          fAlpha*=texLookup(i, gl_TextureMatrix[i]*vPos)[3];

        if(fAlpha>0.9) return true;

        fPos+=fStepSize;
    }
    return false;
}

void occTestWeights(vec4 vFragPos, float fStep, float fRange,
                     float fStart)
{
    int iAW[8];
    int iNW=0;

    for(int i=0;i<iNumActiveWeights;i++)
        if(!occTest(vFragPos, avCamPos[aiActiveWeights[i]],
                    fStep, fRange, fStart))
            iAW[iNW++]=aiActiveWeights[i];

    for(int i=0;i<iNW;i++) aiActiveWeights[i]=iAW[i];
    iNumActiveWeights=iNW;
}

void normWeights()
{
    float fLen=0.0;
    for(int i=0;i<iNumActiveWeights;i++)
        fLen+=fWeights[aiActiveWeights[i]]*
```

```
                fWeights[aiActiveWeights[i]];

    fLen=sqrt(fLen);

    for(int i=0;i<iNumActiveWeights;i++)
        fWeights[aiActiveWeights[i]]/=fLen;
}

vec4 calcColWeighted()
{
    vec4 v=vec4(0.0, 0.0, 0.0, 0.0);
    for(int i=0;i<iNumActiveWeights;i++)
    {
        v+=vCols[aiActiveWeights[i]]*
                weightLookup(aiActiveWeights[i]);
    }
    v[3]=1.0;
    return v;
}

void main()
{
    gl_FragColor=vec4(0.0,0.0,0.0,1.0);
    if(checkOccupancy()<0.5)
    {
        discard;
        return;
    }

    checkWeights();
    occTestWeights(vPos, 0.01, 1.0, 0.1);
    normWeights();

    gl_FragColor=calcColWeighted();
    return;
}
```

B.2 Ray Cast Sampling GLSL Code

B.2.1 Ray Cast Sampling Vertex Shader

```
uniform mat4 osg_ViewMatrixInverse;
uniform mat4 osg_ModelViewMatrix;
varying vec4 vVertPos;
varying vec4 vEyePos;

void main()
{
    vVertPos=osg_ViewMatrixInverse*gl_ModelViewMatrix*gl_Vertex;
    vEyePos=osg_ViewMatrixInverse*vec4(0.0, 0.0, 0.0, 1.0);
    gl_Position=ftransform();
}
```

B.2.2 Ray Cast Sampling Fragment Shader

```
uniform sampler2D CamTex0;
uniform sampler2D CamTex1;
uniform sampler2D CamTex2;
uniform sampler2D CamTex3;
uniform sampler2D CamTex4;
uniform sampler2D CamTex5;
uniform sampler2D CamTex6;
uniform sampler2D CamTex7;
uniform vec4 vCamPos0;
uniform vec4 vCamPos1;
uniform vec4 vCamPos2;
uniform vec4 vCamPos3;
uniform vec4 vCamPos4;
uniform vec4 vCamPos5;
uniform vec4 vCamPos6;
uniform vec4 vCamPos7;
uniform mat4 osg_ViewMatrixInverse;
uniform mat4 osg_ViewMatrix;
uniform int iHigh;
uniform float fHigh;
uniform mat4 camWeights;
uniform int i1RelCam;
uniform int i2RelCam;
uniform int i3RelCam;
varying vec4 vVertPos;
varying vec4 vEyePos;
uniform mat4 camBright;
uniform int NumCams;

vec4 vCols[8];
float fWeights[8];
vec4 avCamPos[8];
int iMaxWeight;
int iSecWeight;

int aiActiveWeights[8];
int iNumActiveWeights=0;

vec4 texLookup(int i, vec4 vCoord)
{
    switch(i)
    {
        case 0: return textureProj(CamTex0, vCoord);
        case 1: return textureProj(CamTex1, vCoord);
        case 2: return textureProj(CamTex2, vCoord);
        case 3: return textureProj(CamTex3, vCoord);
        case 4: return textureProj(CamTex4, vCoord);
        case 5: return textureProj(CamTex5, vCoord);
        case 6: return textureProj(CamTex6, vCoord);
        case 7: return textureProj(CamTex7, vCoord);
    }
    return vec4(0.0, 0.0, 0.0, 0.0);
}

float weightLookup(int i)
{
    switch(i)
    {
        case 0: return camWeights[0][0];
        case 1: return camWeights[0][1];
        case 2: return camWeights[0][2];
        case 3: return camWeights[0][3];
        case 4: return camWeights[1][0];
        case 5: return camWeights[1][1];
        case 6: return camWeights[1][2];
        case 7: return camWeights[1][3];
    }
    return 0.0;
}

float brightLookup(int i)
{
    switch(i)
    {
        case 0: return camBright[0][0];
        case 1: return camBright[0][1];
        case 2: return camBright[0][2];
        case 3: return camBright[0][3];
        case 4: return camBright[1][0];
        case 5: return camBright[1][1];
        case 6: return camBright[1][2];
        case 7: return camBright[1][3];
    }
    return 0.0;
}

vec4 camLookup(int i)
{
    switch(i)
    {
        case 0: return vCamPos0;
        case 1: return vCamPos1;
        case 2: return vCamPos2;
        case 3: return vCamPos3;
        case 4: return vCamPos4;
        case 5: return vCamPos5;
        case 6: return vCamPos6;
        case 7: return vCamPos7;
    }
    return vec4(0.0, 0.0, 0.0, 0.0);
}

float checkOccupancy(vec4 vPos)
{
    float fAlpha=1.0;
    for(int i=0;i<NumCams;i++)
    {
        vCols[i]=texLookup(i, gl_TextureMatrix[i]*vPos);
        fAlpha*=vCols[i][3];
    }
    return fAlpha;
}

vec4 calcColWeighted()
{
    vec4 v=vec4(0.0, 0.0, 0.0, 0.0);
    for(int i=0;i<iNumActiveWeights;i++)
    {
        v+=vCols[aiActiveWeights[i]]*
                weightLookup(aiActiveWeights[i]);
    }
    v[3]=1.0;
    return v;
}

void checkWeights()
{
    iNumActiveWeights=0;
    for(int i=0;i<NumCams;i++)
        if(fWeights[i]>0.0)
            aiActiveWeights[iNumActiveWeights++]=i;
}

void main()
{
    vec4 dir=vVertPos-vEyePos;
    dir=normalize(dir);
    dir*=0.9;
    vec4 vStart=vVertPos-dir;
    vec4 vStep=dir/fStepLen;

    for(int i=0;i<fStepLen;i++)
    {
        if(checkOccupancy(vStart)>0.9)
        {
            populate2();
            checkWeights();
            normWeights();
            gl_FragColor=calcColWeighted();
            return;
        }
        else vStart+=vStep;
    }
    discard;
}
```