

Improving the performance of
Video Based Reconstruction
and validating it within a
Telepresence context

Tobias William Duckworth



UNIVERSITY OF SALFORD

SCHOOL OF COMPUTING, SCIENCE AND ENGINEERING

2013

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

Contents

0.1	Abstract	xii
1	Introduction	1
1.1	Communication and technology	2
1.2	Capturing 3D models	4
1.3	Motivation and setting	6
1.4	Scope of this thesis	9
1.4.1	Contributions	10
1.4.2	Collaboration and this research	11
1.4.3	Thesis overview	12
2	Methodology	14
2.1	Setting	15
2.2	Aim and Objectives	15

<i>CONTENTS</i>	ii
2.2.1 Aim	15
2.2.2 Objectives	15
2.2.3 Research questions	16
2.2.4 Hypotheses	16
2.3 Research process	17
2.3.1 Summary of research process	17
2.3.2 Common practise	18
2.3.3 Derivation of research process	19
2.3.4 Literature review	20
2.3.5 Prototyping and evaluation	21
2.3.6 Interpretation	25
2.3.7 Comparison to literature	26
2.3.8 Conclusions and future research	26
2.4 Experiments	27
2.5 Evolution of the research	29
2.5.1 Selecting an approach to 3D reconstruction	29
2.5.2 Working with real data	31
2.5.3 Focus on temporal quality	33
2.5.4 Parallelisation of EPVH	34

<i>CONTENTS</i>	iii
2.5.5 Evaluation for telepresence applications	35
3 Background and Related work	37
3.1 Introduction	38
3.2 Telepresence	39
3.2.1 Telepresence and eye gaze	39
3.2.2 Remote collaboration	41
3.2.3 Remote collaboration and human modelling	41
3.3 3D reconstruction approaches	42
3.3.1 Active 3D reconstruction methods	43
3.3.2 Passive 3D reconstruction methods (Image based modelling)	43
3.3.3 Single camera image based modelling	44
3.3.4 Two camera (stereo) image based modelling	45
3.3.5 Multiple camera image based modelling	46
3.3.6 Volumetric modelling from multiple camera images	47
3.3.7 Surface reconstruction from multiple camera images	49
3.3.8 Image based rendering	53
3.3.9 Camera calibration	53
3.4 Early research	54

<i>CONTENTS</i>	iv
3.4.1 Initial literature survey	54
3.4.2 Pilot experiments	56
3.4.3 EPVH implementation and development	59
3.5 Conclusion	60
4 Quality of 3D reconstruction	62
4.1 Research questions and hypotheses	63
4.1.1 Hypotheses	63
4.2 Qualities	63
4.2.1 Spatial quality	64
4.2.2 Visual quality	66
4.2.3 Temporal quality	67
4.2.4 Relationship between spatial, visual and temporal quality .	68
4.2.5 EPVH algorithm and quality	69
4.2.6 Quality requirements for telepresence	70
4.3 3D reconstruction system components impacting upon quality . .	71
4.3.1 Camera choice	71
4.3.2 Camera placement	73
4.3.3 Camera calibration	76

<i>CONTENTS</i>	v
4.3.4 Background segmentation	76
4.3.5 Camera exposure and colour balance	79
4.3.6 Reconstructed model transmission	80
4.3.7 Experiment E1 - Comparing the latency of an ICVE and video conference	80
4.4 Temporal consistency of inputs and quality	82
4.4.1 Experiment E2 - Synchronization of images from multi- ple cameras to reconstruct a moving human	83
4.4.2 Experiment E4 - Investigating the suitability of a soft- ware capture trigger in a 3D reconstruction system for telepresence	87
4.5 Experiment E5 - Whole frame period temporal inconsistency . . .	90
4.5.1 Method	91
4.5.2 Results	92
4.5.3 Discussion	103
4.6 Conclusion	106
4.6.1 Quality measures and their relationship	106
4.6.2 Impact of 3D reconstruction system components on quality	107
4.6.3 Temporal consistency of inputs	108
5 Improving performance of VBR	110

5.1	Objectives and research questions	111
5.2	Introduction	111
5.2.1	Context	112
5.3	Related Work	113
5.4	Motivation and Scope of work	115
5.5	EPVH Algorithm and primitives	116
5.5.1	Camera Images and 2D primitives	116
5.5.2	Silhouette cones and 3D primitives	117
5.6	Implementation	118
5.6.1	Methodology	119
5.6.2	Inputs, outputs and data preparation	119
5.6.3	Parallelisation	120
5.7	Evaluation	131
5.7.1	Methodology	131
5.7.2	OpenCL vs OpenMP for parallel CPU execution	133
5.8	Results	134
5.8.1	Comparison with sequential EPVH	134
5.8.2	Comparison with distributed EPVH	134
5.8.3	Evaluation of parallel implementation	136

<i>CONTENTS</i>	vii
5.8.4 Comparison of OpenCL vs OpenMP performance	142
5.8.5 R+-Tree optimisation results	143
5.8.6 Evaluation using humans	143
5.9 Discussion	148
5.9.1 Comparison with sequential EPVH	148
5.9.2 Comparison with distributed EPVH	148
5.9.3 Evaluation of parallelisation	148
5.9.4 OpenCL vs OpenMP for parallel CPU execution	150
5.9.5 R+-Tree intersection	151
5.9.6 Evaluation using humans	151
5.10 Conclusion	151
6 Evaluation	153
6.1 Objectives and research questions	154
6.2 3DRecon, a utility for investigating quality in 3D reconstruction .	154
6.2.1 Related work	155
6.2.2 Overview of 3DRecon's features	156
6.2.3 Camera interface	156
6.2.4 Viewer	157

<i>CONTENTS</i>	viii
6.2.5 Simulator	160
6.3 Case study 1: Camera Placement	161
6.3.1 Spatial quality: overall constraint of the visual hull	161
6.3.2 Reconstruction of the face	164
6.3.3 Discussion	172
6.4 Case study 2: Investigating Eye Gaze	173
6.4.1 Introduction	173
6.4.2 Method	174
6.4.3 Results and analysis	179
6.4.4 Discussion	186
6.5 Conclusions	186
7 Discussion and Conclusions	188
7.1 Discussion	189
7.1.1 Aim and point of departure	189
7.1.2 Review of objectives and research questions	189
7.1.3 Review of methodology	191
7.1.4 How literature guided the research	191
7.1.5 Contributions of the research to literature	192

<i>CONTENTS</i>	ix
7.1.6 Prototyping and evaluation	193
7.2 Review of hypotheses	195
7.2.1 Shortcomings of the research	196
7.3 Conclusion	198
7.3.1 Future research directions	200
A Equations	220
A.0.2 Pinhole camera model	220
A.0.3 Formation of a unit vector from an image point	221
A.0.4 Projection at infinity of a unit vector on a camera image plane	221
A.0.5 Projection of the minimum epipolar extent	222
A.0.6 Projection of a 2D epipolar line intersection onto a 3D line	222

*In loving memory of Heather Moffett,
Companion, soul-mate, lover, best friend and partner for a decade.*

*With your encouragement I embarked upon this journey,
Your daily loving support helped me to continue with it,
Through the memory of you I found the strength to complete it.*

Tragically gone, you are deeply missed, never forgotten, loved forever.

Acknowledgements

With special thanks to: David Roberts, for his supervision and friendship. John O'Hare, for his long term friendship, suggestion of doing the PhD in the first place, and indispensable technical support in the Octave laboratory. Carl Moore for being my PhD brother, and our collaboration throughout, culminating in creating the final end-to-end system.

Thanks to Neil Robertson and Ian Drumm, for examining this thesis, and providing useful comments and suggestions for improving the delivery of its content.

Many thanks to EPSRC and OMG Vicon for sponsoring this research.

Thankyou to all of my family and close friends. Each of you helped me through 2012, the worst year of my life so far. Without your understanding, support and belief in my ability to pick myself up again, I may never have completed this work.

0.1 Abstract

This thesis investigates visual, spatial and temporal qualities of video based reconstruction with respect to telepresence. State of the art was improved and validated through a new parallelisation of an established algorithm; a tool that allows visio-spatial impact of algorithm and camera arrangement to be visualised; and a set of experiments to derive requirements and investigate outcomes. The motivation is to support the exchange of appearance and attention between moving humans through video based reconstruction. A previous research project showed moving humans could faithfully convey attention in virtual environments and appearance through video-conferencing, suggesting that it may be possible to combine the two. Video based 3D reconstruction of humans appeared to be able to achieve both, but it was uncertain whether this could be achieved at sufficient quality. Research began by justifying the approach and setting the requirements. A literature survey and initial experiments indicated that the visual hull provided a form suitable for modeling humans. However, evidence of visual and temporal qualities necessary to support gaze was not found. A state of the art visual hull reconstruction algorithm was parallelised to run on a modern multi-core processor, enabling human reconstruction on a single computer, thus providing stable visual and temporal qualities. A parallelisation scheme theoretically better suited for execution on a single multi-core processor than distributed over a network is proposed. Importantly, the way in which the problem is parallelised has been optimised to reflect the various stages of the process rather than the need to minimise data communication across a network. A utility application has been developed providing a framework for rapidly testing algorithms, validating requirements, and as a platform for conducting experiments. This underpinned a collaborative experiment that showed for the first time that eye gaze could be conveyed to accuracies sufficient for human social interaction. To facilitate the analysis, the utility allowed the impact of camera placement on spatial and visual quality to be investigated.

Chapter 1

Introduction

The ability to capture the human form in three dimensions has many applications and remains the focus of much research. One such application is telepresence, the ability to convey the feeling of presence across a distance, so that participants may feel or appear as though they are present at another location. Video conferencing (VC) and immersive collaborative virtual environments (ICVE) are both examples of telepresence systems. VC faithfully conveys what participants look like, whilst ICVEs can faithfully convey what participants are looking at, but neither can achieve both. In VC, careful alignment of camera and screen can create the illusion of eye contact, but gaze direction and focus of attention are not evident. In ICVEs participants are represented by an avatar that may not resemble them and does not convey facial expressions or meaningful gestures, both of which are important parts of human communication.

The research described by this thesis aims to capture the human form in three dimensions, quickly enough, and with sufficient detail to be useful for future telepresence systems such as ICVEs. By doing so the future collaboration of remotely situated humans can surely be improved.

1.1 Communication and technology

The telephone, and two way radio have enabled voice communication over a distance for well over one hundred years, and they have become indispensable to modern society. Teleconferencing allows multiple parties from multiple locations to take part in a telephone conversation, but this technology gives rise to turn taking problems, such as interruption and extended silences, since the participants are not sure who should be talking next.

Video conferencing

Video conferencing has its conceptual roots as far back as the development of the television, but has only become technically feasible for the general population since the implementation of global high bandwidth digital infrastructure. In the early days of the internet the infrastructure was such that even short text based communications such as emails could only be achieved in pseudo real-time. It is well known that human communication extends beyond voice alone, and that visual cues are used to convey additional information not present in the words spoken or voice tone used. Video conferencing adds some of these visual cues to a conversation by enabling participants to see each other. This works very well for conferences involving two participants, but begins to break down where more people are involved. This is largely because it is not possible to determine where each participant is looking, and eye-gaze is an important cue for turn taking in a conversation.

Immersive collaborative virtual environments

ICVEs extend the concept of remote collaboration into three dimensions, enabling participants to interact in a shared virtual environment, and work on a shared data set. However, the representation of participants by an avatar does not convey what

someone looks like, or any of the visual cues available in a video conference; smiling, frowning, etc. Recently, researchers have extended ICVEs to address some of these shortcomings by capturing important information such as eye gaze direction, eye pupil dilation and whether the user is currently talking. This information can be used to animate the avatar's eyes and mouth in order to make its appearance more convincing, and importantly to convey the direction of participants' eye gaze.

Existing telepresence systems

Telepresence is the ability for a human to feel, or appear as if they are present at a place other than their true location. Modern telepresence systems are typically high-end 2D video conferencing systems. A number of carefully positioned cameras, screens and seats are employed to allow a group of people in one location to meet with another group in a different location and have the impression of being able to judge approximately where participants in the meeting are looking. This leads to a number of shortcomings, primarily that participants cannot look directly at each other due to the offset between cameras and screens, but also that they are unable to move around freely as the illusion of eye gaze is created by the alignment of cameras, seats and screens.

3D telepresence

Bringing together the advantages of both VC and ICVEs into a 3D telepresence system requires the ability to capture the true form and look of the participants in real-time. This replaces the avatar with a model that looks like the participant and conveys their facial expressions, bodily movements and eye gaze. ICVEs enable participants to move around one another and their shared virtual environment, viewing the virtual scene from the point of view of their avatar. Therefore, participants would need to be captured and modelled in their entirety, in three dimensions, so that their representation can be viewed from any arbitrary direction.

3D telepresence is the application forming the main motivation for this research, although there are other applications the research lends itself to such as free view-point 3D video for television or film. Technology has always been, and remains key to advances in telepresence, whether by simply increasing quality through input component improvements, or by adding further senses or dimensions to the telepresence experience.

1.2 Capturing 3D models

A number of approaches to 3D model capture exist, falling largely into two categories, active and passive. Active methods ([124], [115]) direct signals towards the object and analyse their reflection to form a 3D representation. Passive methods use natural images, such as those from a conventional camera, of an object from more than one viewpoint to reconstruct a 3D model. These are known as image based reconstruction (IBR) techniques [30]. Video based 3D reconstruction (VBR) is a technique that can be used to create dynamic three dimensional models of objects from video streams. The process may be regarded as an extension into four dimensions of IBR. In its simplest form, VBR can be implemented as a series of discreet IBR instances, each instance reconstructing a single frame from a video sequence. Such an implementation results in a static 3D model for each input video frame, the impression of movement in the model or scene being created by repeating the process at high frequency.

Over the past decade there has been significant research into IBR, and since there are diverse applications for 3D models, a multitude of approaches to creating them exist. Some attempt only the animal binocular (stereo) 3D modelling of an object [7], which yields at most half of the object. Others attempt full 3D reconstruction of the object: by taking simultaneous images from several viewpoints [9], or by progressively modifying the reconstructed 3D model as the object is moved around a single viewpoint [125]. Each technique has its advantages and disadvantages, and the appropriate approach for any particular application varies. Some

result in spatially faithful and detailed reconstructions of an object, which can be a computationally expensive, and therefore a time consuming operation; others form less faithful reconstructions, but are suitable for real-time applications.

Quality of 3D reconstruction

Specific 3D reconstruction applications determine the requirements, which can be defined in terms of three quality measures:

- Spatial quality defines the faithfulness of the form, how accurately the reconstruction models the original object.
- Visual quality provides a measure of how well the reconstruction resembles the original.
- Temporal quality measures how quickly the reconstruction is achieved.

In general, as spatial and visual quality increase temporal quality decreases, and therefore for real-time applications such as telepresence these qualities must be balanced to achieve the desired quality of representation and frame rate.

Spatial and visual quality are largely determined by the complexity of inputs to a reconstruction system, namely the number of cameras and their resolution. Adding cameras will increase spatial quality by providing more constraints to the reconstruction algorithm. Increasing camera resolution will improve both spatial and visual quality by capturing more detail that can be used for reconstruction and texturing respectively. Increased input complexity will result in a decrease in temporal quality as more data will take longer to process. However, a careless approach to increasing input complexity may significantly reduce temporal quality, whilst having a negligible effect on spatial or visual quality. For example, when adding cameras to a reconstruction system careless placement may result in an increased processing burden without an appreciable improvement in faithfulness of the reconstruction. Furthermore, adding cameras will increase the data transfer

requirements (usually constrained by network bandwidth) of a reconstruction system, resulting in a maximum possible number of cameras for a particular system infrastructure.

The focus of this thesis is to determine how 3D reconstruction can be used to capture in 3D the dynamic human form in real-time, as faithfully as is necessary for use in a telepresence system.

1.3 Motivation and setting

Human communication between individuals in different locations could be enormously improved if people were able to see each other, and move around each other, as if they were in the same room. By capturing images of humans with a number of cameras, and creating 3D models from these images, it should be possible to recreate human interactions in a virtual setting, across a distance.

Video conferencing (VC) is able to convey what someone looks like, including non-verbal cues such as facial expressions, that form an important part of human communication. In VC, however, cameras and screens need to be closely aligned to give the viewer the impression that the person being captured is looking towards them. Not only does this illusion begin to break down when several parties are involved, but it forces participants to remain in one fixed location. Previous research [102] has shown that immersive collaborative virtual environments (ICVEs), in combination with eye-tracking equipment to convey eye gaze, and body tracking to allow participants to freely move around each other, can be used to enhance communication at a distance.

Real-time 3D Reconstruction from video streams has the potential to capture both what someone looks like and what they are looking at, combining the benefits of VC and ICVEs whilst solving some of their shortcomings. If the avatars used in ICVEs could be replaced with real-time generated models of the participants at sufficient quality, people would be able to move around each other in the vir-

tual setting and use body language, facial expressions, gestures and eye gaze to communicate in a manner more closely resembling real-world human interaction.

Participants would be captured from a number of cameras whose position and orientation have no special alignment requirements, allowing free movement within the capture area. The resulting camera images can be used to reconstruct the 3D form of each participant for display at a remote location. Since each person is no longer tied to a particular camera, and their entire form is captured by cameras surrounding them, participants should be able to look directly at one another, or better judge the direction of each other's eye gaze. Furthermore, participants can now meet in a 3D virtual setting, enabling collaboration with dynamic virtual objects.

The literature provides a wealth of approaches suitable for 3D reconstruction of the entire human. Many of these, such as multi-view stereo [43] are capable of producing very high quality, spatially accurate and visually faithful models, but take minutes to achieve this result. For real-time applications such as telepresence, models must be calculated quickly enough to achieve interactive frame rates. Techniques based on the shape-from-silhouette principle [9] form an approximation to the 3D shape known as the visual hull [67] and can be achieved at a much lower processing cost. These techniques provide a good point of departure for balancing temporal, spatial and visual quality in the context of a real-time 3D reconstruction system. This is because the faithfulness of the visual hull to the object being modelled improves as more cameras are used to form it, whilst the time taken to achieve it increases.

Many researchers have investigated volumetric 3D reconstruction [77] as a means for reconstructing the visual hull, and a number of optimisations can be used to increase performance [22], some suitable for real-time applications [117]. More recently, parallel processing has been used to achieve better performance either through network distributed processing [123], or local processing on multi-core CPUs or GPUs [65]. Whilst volumetric reconstruction is a good fit for the temporal quality purposes of this research there are a number of shortcomings that

are discussed in more depth in Chapter 3. Primarily, volumetric reconstruction results in a voxel set which must be post processed to generate a polygonal surface model, which increases processing overhead and gives rise to surface artefacts. The preference for a polygonal model in the context of this research is twofold: A polygonal model occupies less memory than a voxel set and can be further compressed, reducing latency in transmission. Polygonal models can be rendered using general purpose primitives such as those provided by OpenGL, and hardware acceleration of this rendering is commonplace in graphics chips.

Therefore, the ability to form the visual hull directly from camera images, without an intermediate volumetric step could directly form a polygonal model without surface artefacts and remove the burden on memory of the voxel set. Recent research has introduced such direct surface recovery techniques [68] but fails to make topological guarantees. The Exact Polyhedral Visual Hulls (EPVH) algorithm forms a guaranteed watertight and manifold polyhedral surface model [39], making it the state-of-the-art approach. Whilst the original description of the algorithm is presented for sequential processing and does not achieve real-time 3D reconstruction, a network distributed parallelisation is presented in [41] enabling interactive frame rates with 8 cameras. This appears to be the only research into parallelising direct surface 3D reconstruction for real-time applications.

Recently CPUs and GPUs have become available that have increasing numbers of processing cores, potentially making it possible to achieve real-time 3D reconstruction on a single computer. Adapting the EPVH algorithm for processing in a shared memory context on local multi-core compute resources could therefore eliminate the requirement for network distributed processing. Such an achievement would be of benefit for a real-time telepresence system for two reasons: Latency incurred by network distributed processing would be eliminated, improving human communication through the medium. The system cost and complexity would be enormously reduced by replacing a network of computers with a single machine.

1.4 Scope of this thesis

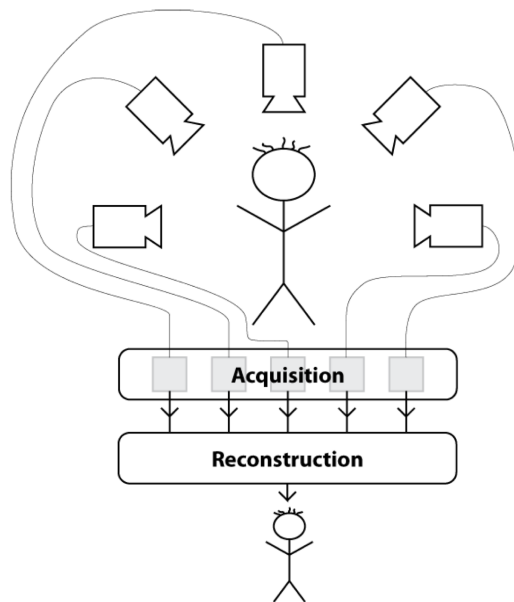


Figure 1.1: Overview of a 3D reconstruction system.

Figure 1.1 depicts a 3D reconstruction system, from cameras to reconstructed output model. The scope of this thesis is confined to the "Reconstruction" region of the system. The camera acquisition stage forms the basis of a concurrent research project.

There are a number of existing approaches to 3D reconstruction from camera images, ranging significantly in performance and quality output. The scope of this research is to investigate 3D reconstruction approaches and algorithms for the purposes of reconstructing humans at sufficient speed and quality to combine the strengths of VC and ICVEs.

Within this scope, and in the context of a 3D telepresence system, the thesis will:

- Define the requirements in terms of visual, spatial and temporal quality.
 - The relationships between these qualities, and impact of wider 3D reconstruction system components upon them is explored. The focus

becomes temporal quality.

- Support the requirements through 3D reconstruction algorithm implementation.
 - 3D reconstruction algorithms are implemented and their ability to meet the requirements is determined. A particular algorithm, and improving the performance of its implementation becomes the focus.
- Validate the requirements.
 - A research platform is developed that allows investigation of the impact of camera placement on spatial and visual qualities required. Two case studies validate it for this purpose.

1.4.1 Contributions

The primary contributions of this thesis are:

- Provides the first investigations into the effect removing camera image synchronisation has on the reconstruction of moving humans, validating the requirement for hardware synchronised cameras.
- Presents a parallelisation of a state of the art 3D reconstruction algorithm for execution on a single multi-core processor and tests the performance over a range of camera numbers and resolutions. The performance of the implementation on multi-core CPUs and GPUs is also compared.
- A research platform is developed that can be used to study 3D reconstruction of humans in the context of telepresence. The platform provides the basis for a collaborative experiment that showed for the first time that eye gaze could be conveyed to accuracies sufficient for human social interaction

1.4.2 Collaboration and this research

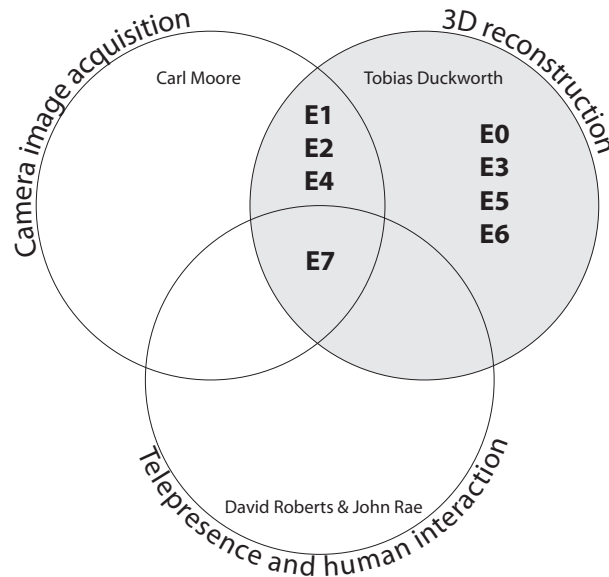


Figure 1.2: Placement of the project within the wider research group. The shaded region denotes the scope of this thesis, overlapping regions denote collaborative research undertaken with other members of the research group.

The scope of this thesis has been described and is confined to the 3D reconstruction algorithm. However, some outputs from this research have been collaborative where there is overlap with the agendas of other projects or interests within the wider research group. Figure 1.2 illustrates this overlap with the primary researchers with whom collaborative outputs were achieved. The collaborations shown in the overlapping circles of the diagram are presented in terms of experiments that were conducted, the details of each can be found in Chapter 2. Where experiments led to publications the relationship is shown in Table 1.1.

Camera image acquisition

The camera image acquisition part of the collaborative research diagram in Figure 1.2 depicts the work of a concurrent research project that investigated the acquisition and distribution of images captured by a number of cameras to the 3D re-

E1	Comparing the latency of VC and ICVE [103]
E2	Synchronization of images for reconstruction [82]
E3	Investigating the effects of "free-running" camera synchronisation [32]
E4	Investigating the suitability of "Pull" camera synchronisation [83]
E5	Accelerated polyhedral visual hulls using OpenCL [33]
E6	Parallel processing for real-time reconstruction from video streams [34]
E7	Investigating the gaze of a virtuality human [104]

Table 1.1: De-referencing specific experiments and their publication output

construction algorithm. This part of the system was researched and implemented by Carl Moore. The challenges faced by the camera acquisition stage included the synchronised capture of images from cameras, compression of resulting video streams to enable high frame-rate throughput to the reconstruction algorithm, and background segmentation of the video streams to provide silhouettes to the shape-from-silhouette based algorithm. The main overlap between the the research described within this thesis and that undertaken by the camera acquisition stage is where decisions made within the scope of camera acquisition affect output quality of the 3D reconstruction. Many of these dependencies are discussed in Chapter 4.

1.4.3 Thesis overview

The rest of this thesis is organised as follows:

- Chapter 2 - The aim, approach, objectives, research questions and hypotheses are presented, followed by a description of the research methods employed in this thesis. Experiments aimed at answering research questions or meeting objectives are presented.
- Chapter 3 - A review of the literature and background to the research, followed by some outcomes of the early research undertaken to narrow down choice of 3D reconstruction approaches.
- Chapter 4 - A study of quality in 3D reconstruction. Three quality measures

are defined. The relationship between these qualities is explored, and the necessity to balance them discussed. Factors affecting quality in the context of a 3D telepresence system are studied, leading to experiments designed to validate requirements and better understand their impact on quality.

- Chapter 5 - Leading on from the definition of three quality measures, we focus on temporal quality. A parallelisation of a state-of-the-art 3D reconstruction algorithm is presented, along with a number of optimisations that enable more cameras at a higher resolution in real-time, and hence provide the mechanism by which better spatial and visual qualities may be achieved.
- Chapter 6 - A research platform used to evaluate 3D reconstruction is presented. The various aspects of this utility are described in relation to validating the research as a whole. Two case studies are presented as a means to demonstrate the efficacy of the research platform in analysing the impact of camera placement on spatial and visual quality.
- Chapter 7 - Discussion of the thesis, reflecting on the methods and approaches taken, culminating in conclusions and identifying directions for future research.

Chapter 2

Methodology

This chapter presents the methodology used in conducting this research. The overall aim of the research is described, key objectives and research questions identified, and hypotheses formulated. Following this, the methods comprising the iterative research process are described, along with how each has contributed to the overall research, and how one has fed into another. The experiments and evolution of the research are then summarised, providing the framework for the remaining chapters in the thesis.

2.1 Setting

This research examines the use of 3D reconstruction from video streams to model humans at sufficient quality levels to enable a 3D telepresence system to convey the subtleties of human non-verbal communication.

2.2 Aim and Objectives

2.2.1 Aim

To study and improve the balance of temporal, spatial and visual quality of virtual humans, reconstructed from multiple video streams, to support most kinds of non-verbal communication in telepresence.

The approach used was to understand the requirements placed on reconstruction algorithms, improve temporal qualities of an algorithm or its implementation, and study what visual and spatial qualities could be achieved within the real time constraints of telepresence.

2.2.2 Objectives

- **O1:** Determine the requirements and approaches to 3D reconstruction suitable for a real-time telepresence system.
- **O2:** Determine whether algorithms can be improved upon to achieve higher temporal or spatial/visual quality.
- **O3:** Develop a platform through which the impact of camera placement on spatial and visual quality of 3D reconstruction can be studied.

2.2.3 Research questions

- **Q1:** Which approaches to 3D reconstruction can provide the best balance between performance and model quality for representing the dynamic human form in the context of a telepresence system?
- **Q2:** How can algorithms or their implementation be improved upon to achieve higher qualities in real-time?
- **Q3:** What are the requirements of a system with which the impact of camera placement on spatial and visual quality can be studied?

2.2.4 Hypotheses

Where an objective is to answer a research question, the questions are refined to hypotheses:

- **H1:** Approaches to 3D modelling from images can be used to model moving humans at sufficient qualities for telepresence applications.
- **H2:** Relaxing constraints on temporal consistency of inputs will result in a degradation of spatial and visual quality in the case where the subject is moving.
- **H3:** Temporal quality can be improved upon through parallel processing using multi-core GPUs or CPUs. Such an improvement will allow for higher spatial and visual quality in real-time.

2.3.2 Common practise

This thesis is the overlap of two key fields in the research community: 3D reconstruction, and virtual reality/telepresence. Common practice within these disciplines is drawn upon in the research methodology employed. 3D reconstruction researchers implement algorithms, often only testing them in a simulated setting, and report their performance through timing measurements and present visio-spatial quality through pictures in research papers. Telepresence researchers build prototype systems and examine human factors, such as conversations, impact of embodiment and timings. We implement a combination of these practices to enable algorithm prototyping and initial evaluation in a simulated setting, followed by the building of a prototype system which can be evaluated using real humans.

Quality evaluation

Evaluation of quality of 3D reconstruction can be a non-trivial process. Whilst temporal quality is simple to evaluate in quantitative terms by measuring latency, spatial and visual qualities are much more difficult to define quantitatively. These qualities will be defined in more depth in Chapter 4, but here we describe the rationale for the methods used to evaluate them.

Spatial quality encompasses the accuracy and completeness of the representation compared with the object being modelled. One method for measuring this quantitatively would be to obtain ground truth data of the object being reconstructed, for example by using a laser scanner. The reconstructed form could then be compared to the ground truth model in terms of the differences between them, which could be quantified as two distinct measures: volumes present in the reconstructed model but missing in the ground truth data, i.e. errors in the reconstruction giving rise to superfluous volume. Or volumes present in the ground truth data but missing in the reconstructed model, i.e. errors in the reconstruction in the form of missing volume. Such measures could be obtained through constructive solid ge-

ometry (CSG) techniques, which are able to evaluate boolean operations on three dimensional data. Given that the visual hull is only an approximation to the form in the first place, and is unable to model surface concavities, the value of quantifying differences between model and ground truth data becomes questionable. The importance of spatial quality in the context of telepresence applications is that human behaviour and non-verbal communication can be conveyed, and therefore it is more important that the waving of a hand or shaking of a fist can be differentiated than that the width of the arm used to convey that action is accurate.

Visual quality can be regarded as a subjective quality measure. Whilst quantitative differences between pixel colours in camera images compared to the reconstruction could be compared, the meaningful value of these differences is dubious. The cameras themselves may have colour or exposure settings that affect appearance of the whole image, and these would first need to be quantified in terms of differences compared to the camera used to obtain the ground truth reference image. Secondly, the importance in terms of a telepresence system is that human communication can be conveyed effectively, and in this sense it is more important that facial expressions can be accurately perceived than whether a person's reconstructed face is slightly more red in the reconstruction than in real life.

Given the complications and possibly dubious value of quantitative measures for spatial and visual qualities in the context of this research, this thesis presents quantitative results for temporal quality and adopts the common practise used in the 3D reconstruction community of presenting spatial and visual quality pictorially.

2.3.3 Derivation of research process

The individual research methods employed are adopted from those widely used in computer science. [99] gives an analysis of the popularity and frequency of different methods used in computer science research papers and provided guidance for methods suitable for this thesis. The specifics of how each method was applied, and the relationship between them are, however, uniquely conceived for

this research. Research methods employed included: literature review, feasibility study, concept implementation, simulation, comparative study, laboratory experiment and case study. Each of these methods contributed to answering research questions, meeting objectives and proving or disproving hypotheses. Figure 2.1 illustrates the relationships between aim and objectives, research questions and selected research methods. The overall aim is the starting point for research, it forms the objectives which in turn lead to research questions. The aim and objectives are considered to be more or less fixed for the duration of the research, but research questions can be influenced by the research process and revised at any point.

2.3.4 Literature review

Literature reviews form an important element of the research method employed, as depicted in Figure 2.1. Reviews and evaluation of the literature fed into the research continuously: when searching for answers to research questions, looking for solutions to implement and test, and when comparing obtained results with the outcome of other research.

Whilst the limits of human perception set the fundamental requirements of this research and need to be accurately determined, the initial set of constraints were derived from the technology widely in use today. For example, the frame rate of a television guides the approximate frequency for temporal quality, and the resolution of digital cameras provided the starting point in terms of visual quality. Therefore the approach to the literature survey was to begin by building up a body of literature on the subject of 3D reconstruction techniques, video conferencing, telepresence, and human modelling. This in turn led to an understanding of commonly accepted measures in human perception, for example frame rates being described as "interactive". Of particular interest were articles where these disciplines overlapped, for example human modelling through 3D reconstruction. There is a wealth of material published on the aforementioned subjects, particularly over the past fifteen years, and several hundred conference papers, journal

articles and book chapters were located. The scope of this material encompasses many disciplines including geometry, algorithms, computer graphics, maths, and photography. The initial body of knowledge shaped research questions and contained certain publications that included sufficient detail to form the first ideas for concept implementation. The wider literature survey also covered material relating to existing telepresence and ICVE implementations in order to understand the background against which future improvements will be made, this includes the technical implementation of telepresence systems as well as psychological studies relating to collaboration through such systems.

During numerous phases of literature review, key authors, conferences and journals were identified, and further material discovered through forward and backward referencing. Backward referencing helped to build a picture of the genesis of a particular research area up to the point of publication, whilst forward referencing highlighted the direction research had taken since publication. Key outlets for literature searches included IEEExplore, ACM, Google Scholar, Web of Science, Science Direct, PubGet and Research Gate.

2.3.5 Prototyping and evaluation

A major aspect of the methodology of this thesis is the prototyping and evaluation of ideas resulting from literature review or novel concept. Figure 2.1 shows how other elements of research feed into prototyping and evaluation. It also shows the methods that are used at successive stages of prototyping and evaluation. As the arrows in the diagram suggest, the process of prototyping and evaluation is progressive. Concepts first undergo a feasibility study before they are implemented, and then may progress through to later stages of evaluation. A particular 3D reconstruction approach will not necessarily proceed through all the listed steps, in fact it is unlikely that it will; concepts can be dropped at any of stages listed because they fail to meet requirements or expectations.

Feasibility study

A feasibility study precedes the implementation of a concept. Where the literature review, or a novel idea has yielded a method or concept that seems to fit the requirements and meet some or all of the objectives, the feasibility is studied in more detail in the context of a final 3D telepresence system. The feasibility of implementing a concept is considered in terms of the requirements of the concept that can be deduced from the literature and how well they fit with the available resources. Concepts evidently failing to meet the overall aim or objectives would generally be ruled out during the literature review, but this may not become clear until feasibility is considered in more detail. Available resources were an important consideration as this guided the direction of research based on what was achievable without significant outlay on new equipment. For example, a 3D reconstruction method that appears to fulfil the aim and meet some of the objectives, but requires 100 cameras might be ruled out during the feasibility study. Other resources analysed during the feasibility assessment were: estimated processing overhead, memory requirements and network bandwidth where applicable. Another means by which the feasibility study can rule a concept out before implementation is where the literature survey does not provide sufficient implementation details. The feasibility study relies heavily on figures quoted in the literature and implementation overhead is only estimated. Therefore the outcome of the feasibility study cannot be considered to be totally reliable.

Concept implementation

Concept implementation takes place when a method or concept has been deemed feasible by the feasibility study. The concept is implemented based on details derived from the literature or in some cases from novel thought and deduction. Apart from at the earliest stages in the research there is often a framework in place that can be reused for parts of the concept implementation. For example, all of the 3D reconstruction approaches that were implemented required the basics of obtaining images from cameras, and therefore such methods became core re-

usable components of a research platform that matured during the course of the overall project. Correct implementation of a concept requires an adequate degree of detail derived from the literature survey or deduced by logical thought. In some cases it does not become apparent until attempting concept implementation that there are insufficient details to complete the implementation. Often further literature searches using disparate sources to that from which the concept was derived were required to bridge these gaps and implement the complete concept.

Simulation

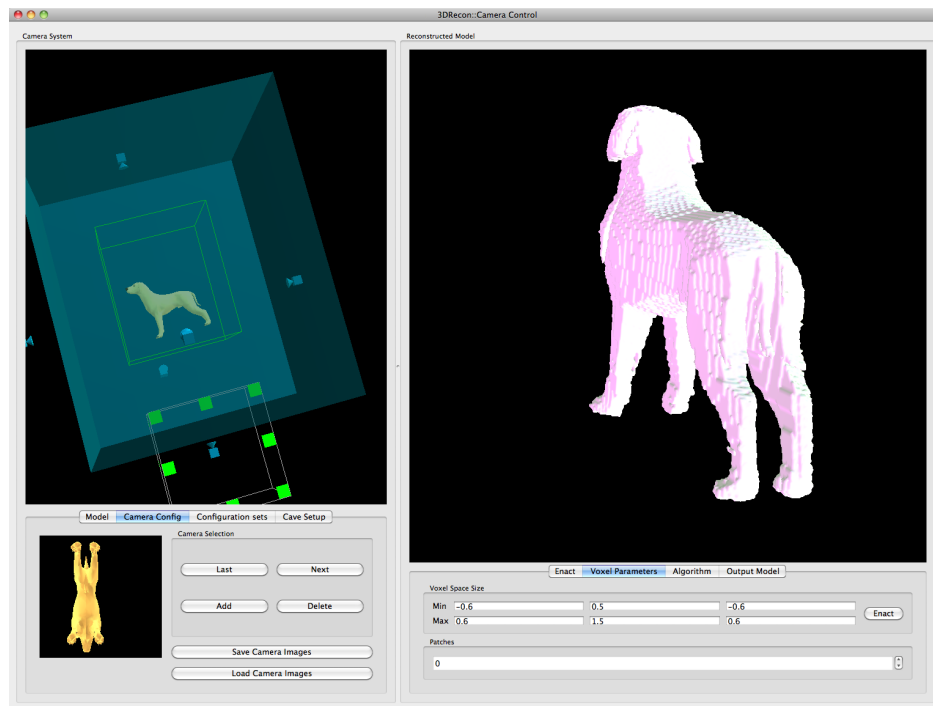


Figure 2.2: An early screen shot of the simulated setting through which many approaches to 3D reconstruction were evaluated. Shown here performing a volumetric reconstruction of a synthetic dog model from 6 virtual cameras.

Given the nature of the research problem being investigated, and the time taken to set up and calibrate real cameras, simulation became an invaluable part of the research methodology. A simulator was implemented as an integral part of the developed research platform. The simulator allows virtual cameras to be placed

around a synthetic object and for implemented algorithms to be tested using images derived from these virtual cameras, as shown in Figure 2.2. Many real-world problems are not present in the simulated setting: virtual cameras do not exhibit lens distortion, camera image noise is non-existent, and they are trivial to calibrate compared with real cameras. Therefore, incorporation of a simulator into the research platform enabled rapid prototyping of both algorithms and camera placement that would be much more time consuming in a real-world setting. The first point of testing for algorithms progressing from the concept implementation stage was in simulation.

Comparative study

Comparative study results from the simulated testing of implemented algorithms. Algorithms can be evaluated and compared based on implementation details such as memory or processor overhead, or their ability to meet the spatial, visual or temporal qualities of the end system. Most algorithms could be effectively evaluated during simulation or comparative study, only progressing to the next stages of evaluation if simulated testing and comparison did not rule them out.

Laboratory experiment

Laboratory experiment denotes the stage at which algorithms begin to be tested in a real-world setting. Real cameras replace the virtual cameras used in the simulation setting, and consequently real-world problems become apparent; camera lens choice can lead to image distortion which needs to be corrected before reconstruction is attempted. Camera image noise can make background subtraction less reliable and predictable. Lighting and shadows add to the background subtraction problems. Differences in camera image exposure or colour balance become apparent. Real cameras are much more difficult to calibrate than virtual cameras, and calibration of them can take some time to get right. The simulated setting effectively divorced the algorithm from all of these problems to do with the camera

acquisition stage of a 3D reconstruction system. It could be argued that this research could be conducted in an entirely simulated setting, focussing on algorithm performance and therefore temporal quality, and theoretically enabling improved spatial and visual quality. It would, however, be impossible to test with anything but synthetic human subjects, and consequently real human phenomena such as eye gaze could not be investigated.

Within the scope of this thesis, only one algorithm, EPVH was tested under laboratory experiment conditions. A number of variants of the algorithm were tested (sequential, parallel execution on CPU, parallel execution on GPU), but aside from differences in performance their output was identical.

Case study

A case study was used where an algorithm had been validated in real-world settings using laboratory experiment conditions, and progressed to being used as part of a true experiment. In this research, an experiment was conducted into the ability of people to estimate the eye gaze of a 3D reconstruction of a real person. A number of participants manipulated the 3D reconstructed person until they felt it was looking directly at them. The angle of presentation to the viewer of the body, the head, and the direction of gaze of the eyes in the head, were varied over a number of gaze poses to determine which of these had an impact on gaze estimation. The impact of camera placement on form reconstruction and texturing was also investigated. Further details of this study are presented in Chapter 6.

2.3.6 Interpretation

Results obtained from prototyping and evaluation are (where relevant) subject to interpretation before they can be compared to the literature. Even those approaches which did not progress through all the stages of prototyping and evaluation might be subject to interpretation as there may be something to be learned

from the findings. For example, early experimentation with volumetric reconstruction techniques proved exceptionally slow, and it was not until interpretation of the results and subsequent revisiting of the literature had taken place that the concept of octrees [57] was discovered and implemented.

2.3.7 Comparison to literature

Comparing findings of the research to the literature provides a mean by which conclusions can be formed in the wider context of the research area. Comparison can take the form of any of the measured or observed qualities or characteristics. For example, performance of the implemented EPVH algorithm was quantitatively compared with performance published in the literature. Visual and spatial qualities of the algorithm were qualitatively compared against pictures of reconstructions in the literature. Such qualitative comparisons, however, were only really meaningful where reconstructions had been performed on the same publicly available datasets.

2.3.8 Conclusions and future research

Findings from the whole cycle of the research methodology employed lead to conclusions that can identify gaps in the research undertaken and provide directions for future research. Even if the entire cycle of the research methodology has not taken place, valid conclusions can still be drawn, which may provide new directions for research. These new research directions provide an input for new research questions and hypotheses and therefore bring us back to the start of the research methodology cycle.

2.4 Experiments

The experiments undertaken fall into two categories: independent research and collaborative research. Figure 1.2 shows how collaborative experiments fit into the structure of the wider research group within which the work in this thesis was carried out. The experiments performed can be broadly categorised by research question:

- **Q1:** Which approaches to 3D reconstruction can provide the best balance between performance and model quality for representing the dynamic human form in the context of a telepresence system?
 - **E0** - Pilot experiments to investigate the suitability of various 3D reconstruction approaches to real-time 3D reconstruction of humans. - Various approaches found in the literature were implemented and their suitability tested in a simulated setting, or using real datasets from disk. These approaches were successively narrowed down until only one remained; the EPVH algorithm, which was efficient, scalable, and resulted in high quality visual and spatial reconstructions.
 - **E1 collaborative.** - Comparing the latency of an ICVE and video conference - The latency of both video conferencing, and a typical ICVE setting, where two CAVEs were connected together, was measured. The intention of the experiment was to derive results for guiding the latency expectation of a final end to end 3D reconstruction system [103].
 - **E2 collaborative.** - Synchronisation of images from multiple cameras to reconstruct a moving human - A simulated setting was used to conduct this experiment aimed at determining what the camera frame synchronisation requirements of a 3D reconstruction might be. The experiment studied the effects of relaxing camera frame synchronisation on a rotating synthetic human head model [82].

- **E4 collaborative.** - Investigating the suitability of a software capture trigger in a 3D reconstruction system for telepresence - The experiment investigated using a software trigger ("Pull" approach) compared with a hardware trigger ("Push" approach) for obtaining images from cameras in a 3D reconstruction system. The resulting sub-frame period desynchronisation was measured, and its effect on resulting 3D reconstruction studied [83].
- **E5** - Camera image synchronisation in multiple camera real-time 3D reconstruction of moving humans. - Using pre-captured datasets from disk, an increasing number of camera images up to half of the camera set were delayed by a whole frame period for a variety of human movement sequences. The effect this "Free-running" approach to camera synchronisation had on the reconstruction was presented. The motivation for the experiment was to determine what the effects might be of using commodity cameras such as webcams that do not have a hardware synchronisation capability [33].
- **Q2:** How can algorithms or their implementation be improved upon to achieve higher qualities in real-time?
 - **E3** - Accelerating polyhedral visual hull reconstruction using OpenCL. - The acceleration achieved by accelerating various discreet sub-processes of the EPVH algorithm, by executing them in parallel on a GPU, was measured [32].
 - **E6** - Parallel processing for real-time 3D reconstruction from video streams. - An end-to-end parallelisation of the EPVH algorithm tailored for local processing was tested. Its performance was compared with published figures for the network distributed parallel approach, and also between multi-core CPU and GPU execution [34].
- **Q3:** What are the requirements of a system with which the impact of camera placement on spatial and visual quality can be studied?
 - **E7 collaborative.** - Estimating the gaze of a virtuality human. - The ability of participants to judge the gaze of a 3D reconstructed human

was measured to determine whether 3D reconstruction can effectively convey eye gaze. The impact of camera placement on spatial and visual quality of the output, and how this affected eye gaze judgement, was analysed [104].

2.5 Evolution of the research

During the course of the research the direction taken was influenced by the process employed, as described in Section 2.3. Here follows a synopsis of the path taken and the reasons why.

2.5.1 Selecting an approach to 3D reconstruction



Figure 2.3: Early volumetric reconstruction performed in simulation from synthetic model and virtual cameras.

Research began by surveying literature for approaches to 3D reconstruction from images, several candidate approaches were prototyped and evaluated, and further

details are given in Section 3.4. The earliest working prototypes were achieved entirely in simulation from virtual camera images of synthetic objects, such as the simplistic voxel reconstruction of a human in Figure 2.3.

Through exploring shape-from-silhouette reconstruction techniques, the concept of direct surface reconstruction algorithms became appealing. A surface based model seemed like the best choice for a telepresence system as it could be represented by a polygonal mesh, for which many display, compression and transmission schemes already exist. Given this, any volumetric method would require a post-process to extract a surface based polygonal model, so if direct surface methods could be achieved sufficiently quickly, the intermediate surface extraction step would become redundant. The Exact Polyhedral Visual Hulls (EPVH) algorithm appeared to meet these requirements, but the ability of it to achieve real-time performance at the desired quality was unknown.

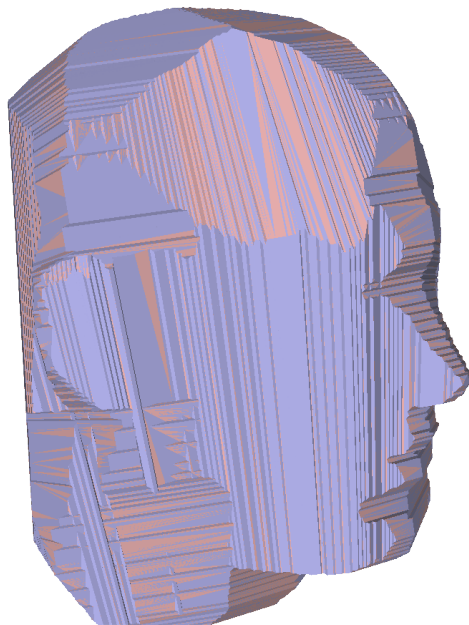


Figure 2.4: Early EPVH reconstruction performed in simulation from a synthetic model and virtual cameras.

Through the description in the literature the EPVH algorithm was implemented, Figure 2.4 shows an early output model reconstructed in simulation from virtual

cameras and a synthetic human head. However, performance of the sequential implementation was not sufficient for real-time applications at the desired output qualities.

2.5.2 Working with real data



Figure 2.5: Reconstruction of Inria's dancer from publicly available dataset using the EPVH algorithm.

Whilst simulation was an invaluable tool for rapidly prototyping and beginning the process of evaluating algorithms, many real-world human phenomena are not easily tested through this medium. Most notably movement. Whilst synthetic models can be crudely animated, they are unable to emulate the way in which a human moves without significant modelling effort. Publicly available datasets comprising camera images of moving humans provided essential material enabling algorithm testing with real-world data, prior to deploying and calibrating a set of cameras in the laboratory. Figure 2.5 shows two novel viewpoints reconstructed from Inria's Dancer dataset ¹.

¹<http://4drepository.inrialpes.fr>

Calibrating cameras



Figure 2.6: A reconstruction from real camera images from our own calibrated camera set.

Deploying and calibrating a set of real cameras can be a time consuming task, and the journey to achieving it in this research was no exception. A number of methods for calibrating cameras are identified in 3.3.9, ultimately the method described in [80] was settled upon.

In this method, cameras forming a set may be calibrated in groups, which is a significant advantage over methods such as [120] that require painstaking calibration of each individual camera. A wand is used with two clearly visible and easily differentiable markers placed a known distance apart, we used two coloured foam balls on a stick. The wand is waved in front of all the cameras simultaneously whilst they capture images. The positions of the centres of the markers in each of the camera images is determined, either in real-time or as an offline post process. The marker positions are tabulated for each frame and each camera; it is

important to maintain consistency in frame numbering between cameras in the process. For cameras where one or both of the markers is not visible, this is denoted. It is extremely important to only record the positions of markers whose visibility and centre position are known with absolute certainty because the calibration algorithm can be recalcitrant in the presence of false positives. Conversely the presence of false negatives does not affect the calibration results providing there are sufficient marker occurrences detected. The algorithm requires only the image centre in pixels and estimated focal length of each camera to begin. The calibration algorithm uses descent methods (bundle adjustment) to iteratively adjust a working model of the complete camera set until pixel reprojection errors are minimised across all of the cameras. A second advantage of this technique over other camera calibration methods is that marker correspondences only need to be present between pairs of cameras in the set, rather than present in the entire set. This means that, providing sufficient pairs of markers exist between all cameras in the set, camera sets comprising cameras pointing in diverse directions can more easily be calibrated.

Given a set of real calibrated cameras we were able to start capturing our own data and begin to conduct laboratory experiments. Figure 2.6 shows an example of an early reconstruction from our own camera set.

2.5.3 Focus on temporal quality

Through literature survey, algorithm prototyping and evaluation, the impact of camera resolution and the number of cameras on the visual, spatial and temporal quality of the visual hull was determined as:

- spatial quality increases as camera count and resolution are increased;
- visual quality increases as spatial quality and camera resolution increase;
- temporal quality increases as input complexity decreases (in general input complexity is proportional to camera count and resolution)

Therefore the relationship between visual, spatial and temporal quality is that temporal quality decreases as spatial and visual quality increase. Hence, if by some means the performance of an algorithm could be increased, it should be possible to increase the spatial and visual qualities possible per unit time. In other words, by increasing temporal quality through enhancing performance, spatial and visual qualities can be improved. An aspect of temporal quality that was explored, aside from the performance of the algorithm itself, was temporal consistency of the inputs and how this impacts on spatial and visual quality of the output. In this context temporal consistency can be regarded as the level of synchronisation of camera images. The motivation for this study came from a concurrent research project studying the camera acquisition stage, in which the question of frame synchronisation had arisen. Not all cameras offer hardware frame synchronisation inputs, and therefore signalling such cameras to provide a frame must be achieved in software, which is unlikely to be temporally consistent across all cameras. Experiment E4 (2.4, E4) was performed as part of the concurrent research project and investigated sub-frame camera desynchronisation. Experiment E5 (2.4, E5) was conducted as part of this research and investigated the effect of whole frame desynchronisation of part of the camera set on the reconstruction of moving humans. Together these experiments provided results supporting hypothesis H2.

2.5.4 Parallelisation of EPVH

Parallel processing was employed to improve the performance of the EPVH algorithm. The authors of the algorithm had described a network distributed implementation [41], which they claimed enabled real-time performance at approximately the qualities required by our application. The initial approach to parallelisation was to identify parts of the algorithm that performed a simple step numerous times with different input data (i.e. SIMD), and implement a discreet parallel processing block for them. Literature review had revealed recent research using multi-core GPUs for general purpose number crunching, and it seemed likely that GPUs could accelerate some parts of the EPVH algorithm. My initial EPVH parallelisation was implemented in OpenCL and executed three parts of the algorithm

in parallel on a GPU. The overall reconstruction time for a range of camera counts and resolution was compared to sequential execution on the CPU in experiment E3 (2.4, E3). The experiment found that the partially parallelised algorithm executed on the GPU offered improved performance compared with sequential CPU execution.

Parallelisation of the EPVH algorithm continued with the end goal of achieving an end to end parallelisation. The approach to parallelisation was completely revisited; the data entities over which the parallel processing was performed were rethought compared to that used in E3, and the parallel partitioning re-analysed. A new end-to-end parallelisation was conceived and implemented in both OpenCL and OpenMP. The new scheme improved performance over that used in E3 and also provided a means of comparison to figures published in the literature for the network distributed approach. Experiment E6 (2.4, E6) tested the new parallel partitioning, and was able to compare its execution on GPUs with that on multi-core CPUs. It was found that the implemented scheme achieved much better performance on multi-core CPUs than GPUs. The parallelisation of EPVH, and the experiments which were performed, proved hypothesis H3 - that temporal quality could be improved through parallel processing - in turn allowing for higher spatial and visual qualities.

2.5.5 Evaluation for telepresence applications

The parallelised EPVH was able to achieve real-time 3D reconstruction of humans at appropriate spatial and visual qualities, in simulation, using pre-recorded datasets from disk, and with our calibrated camera set. The evaluation for telepresence applications comprised of two parts: a one directional live end-to-end system demonstration, and an experiment to determine whether test subjects could accurately judge the direction of eye gaze of a 3D reconstructed person.

The end-to-end demonstration brought together two research projects, one focussing on the camera acquisition stage and this one. Cameras were used to cap-

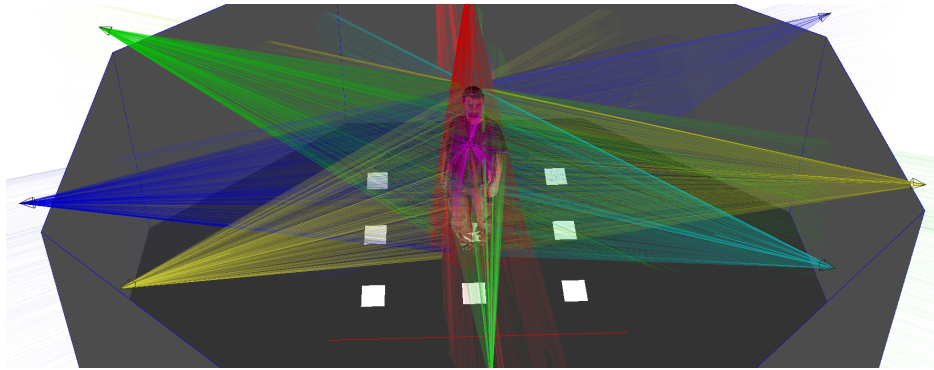


Figure 2.7: The research platform through which human interaction through 3D reconstruction can be investigated.

ture live images of a person in our laboratory, whilst the 3D reconstruction algorithm received these images and performed the reconstruction at another building in the university in front of a number of conference delegates. Skype was used for audio so that the conference delegates could instruct the person being reconstructed to perform a particular action, in order to prove that the demonstration was live. We used wi-fi for connecting the 3D reconstruction computer to the camera acquisition stage, and the receiving computer also had the burden of decompressing all the texture streams. Consequently only about 7 frames per second were achieved, but the concept was proven and latency seemed acceptable.

The collaborative eye gaze experiment was conducted using a research platform that was implemented as part of this thesis. The platform allows human interaction through 3D reconstruction to be tested and provides a means to evaluate the applicability of the implemented algorithms for telepresence applications. The eye gaze experiment showed for the first time that eye gaze could be conveyed to accuracies sufficient for human social interaction. The experiment also provided some analysis on the impact of camera placement on form and texture genesis.

Chapter 3

Background and Related work

In this chapter we review the background and literature related to specific areas of this thesis: Eye gaze and remote interaction, virtual collaboration and 3D reconstruction techniques. Finally, details of early research not described elsewhere in the thesis are provided, serving as a critical analysis of 3D reconstruction approaches presented in the literature.

3.1 Introduction

Specific conferences and journals relevant to this research were:

- Conferences of particular relevance included:
 - 3D Data Processing, Visualization and Transmission (3DPVT)
 - ACM Computer Supported Cooperative Work (CSCW)
 - ACM/IEEE Distributed Simulation and Real Time Applications (DSRT)
 - ACM SIGGraph
 - ACM Virtual Reality Software and Technology (VRST)
 - British Machine Vision Conference
 - Computer Architectures for Machine Perception (CAMP)
 - Computer Graphics, Imaging and Visualization
 - Computer Graphics and Interactive Techniques
 - Conference on Visual Media Production (CVMP)
 - IEEE Automatic Face and Gesture Recognition
 - IEEE Computer Vision and Pattern Recognition
 - IEEE Computer Vision
 - IEEE Virtual Reality
 - Joint Virtual Reality Conference of ICAT - EGVE - EuroVR
 - Pattern Recognition (ICPR)
- Pertinent journals included:
 - ACM Transactions on Graphics
 - ACM Transactions on Multimedia Computing, Communications and Applications
 - ACM Transactions on Computer-Human Interaction

- ACM Transactions on Intelligent Systems and Technology
- ACM Transactions on Modeling and Computer Simulation
- Computer Vision and Pattern Recognition
- IEEE Transactions on Pattern Analysis and Machine Intelligence
- IEEE Transactions on Visualization and Computer Graphics
- IEEE Transactions on Computer Graphics and Applications
- IEEE Transactions on Parallel and Distributed Systems
- International Journal of Computer Vision
- Journal of Mathematical Imaging and Vision
- Journal of Real-Time Image Processing
- Journal of Visual Communication and Image Representation

3.2 Telepresence

Telepresence is the ability for a human to feel, or appear as if they are present at a place other than their true location. In other words, it is the experience of presence in a disparate place by means of a communication medium. Defining virtual reality [122], [59] in terms of this user experience rather than mediating technology is the key to developing telepresence systems in which the user attains a true feeling of presence, rather than a feeling of being restricted by technology. However, this is a difficult goal to attain as technology is a requirement for telepresence.

3.2.1 Telepresence and eye gaze

The role of eye gaze in social interaction is clearly important, and according to [127] the human mind has evolved specialised methods for gaze detection and interpretation. These methods have three functions: detecting the presence of eyes

or eyelike features, computing the direction of gaze, and attributing the state of seeing to the gazer. This system is believed to play a pivotal role in everyday interaction [75]. Eye contact is a natural experience of face to face communication [6] yet video conferencing generally fails to convey it. [84] conducted experiments to study how eye contact affects the impression of telepresence in videoconferencing. [19] presents the results of experiments into human eye contact, showing that there is an asymmetry in sensitivity to eye contact. Humans are an order of magnitude less sensitive to eye contact when people look below the eye than above left or right of it, indicating that the optimal mounting point for cameras is above the screen. However, this only mitigates the problem and does not solve the problem of supporting eye contact in video conferencing.

Three approaches to supporting eye contact through video conferencing exist: Modifying the video so that the user appears to be looking into the camera [46], placing the camera behind a semi-transparent display [56], [90], or behind a hole in a front projected display [16], [107]. Modifying the video can be achieved by 3D modelling of local participants and novel view synthesis from the viewpoint of the remote participant: [25] uses a pair of stereo cameras to create a novel viewpoint enabling gaze correction in a one to one video conferencing setup. Using images of 320 x 240 pixels, the gaze corrected image can be generated approximately every 2 seconds. [63] uses a Kinect to obtain depth maps, and a face detection algorithm to cut out faces enabling synthesis of a gaze corrected face which replaces the original face in the camera image, and demonstrates that such an approach can be used to provide gaze corrected video conferencing with commodity hardware.

Whilst these techniques might improve the experience of eye contact through video conferencing, they do not provide a means for transfer of eye gaze between different places, for example in directing or following the focus of attention. Nor do they provide a means by which several participants in a video conference may perceive who is the current focus of attention as the Mona Lisa effect makes all observers feel looked at when the remote participant looks at the camera [5].

3.2.2 Remote collaboration

In [121] a desk mounted eye-tracker is used to convey eye gaze in a 3D virtual meeting room and within shared documents, but such an approach does not give the participants the freedom to move around, much like video conferencing. Collaborative virtual environments extend the concept of virtual reality into multi-user systems explicitly supporting collaboration [12]. In order to communicate gaze between people in different places, the participants need to be situated in a shared space, such as a collaborative virtual environment. In this setting, the viewpoint into the shared space and the embodiment within it can move around. This provides a 3D context through which directional information such as eye gaze makes sense.

Immersive collaborative virtual environments (ICVEs) are collaborative virtual environments in which an immersive display such as a CAVE [26] places participants inside the shared virtual world by means of surround displays. Within such a setting, and by tracking the participants, their embodiments are able to move around each other as if they were in the real world. In [102] ICVEs were extended to support eye gaze by capturing the direction of participants' gaze using eye tracking hardware. This enables direction of attention through the medium of ICVEs. However, in ICVEs, participants are represented by an avatar, which generally does not resemble the participant, and when it does is captured in an offline process so fails to provide a real-time representation.

3.2.3 Remote collaboration and human modelling

A system that adopts the benefits of both video conferencing, and ICVEs would enable the conveyance of participants' appearance and attention simultaneously. The requirements of such a system are that it captures the participants in 3D, faithfully enough to communicate attention and appearance.

There are several approaches to capturing human appearance in 3D: [42] Uses

trinocular stereo to reconstruct a dense 3D point cloud of the front of a person at 2 to 3 frames per second with end-to-end latency of 1.5 to 7 seconds. [113] and [17] transform a generic model to approximate a person's shape and [52] applies colour textures to it. [58] uses structured light to capture the human face in 3D in real-time. [106] uses a fusion of volumetric and disparity techniques to capture humans for immersive 3D video conferencing. [20] uses volumetric techniques to achieve a 15 frame per second modelling of the entire human form. [2] use network distributed EPVH to capture human hands and insert them in real-time into a virtual environment, and [92] model the entire human body in real-time for insertion into and physical interactions with a virtual environment, however this did not model the face and eyes due to the user wearing a head mounted display. More recently, [76] uses multiple Microsoft Kinect sensors to capture humans in real-time.

Other forms of human remote embodiment take a slightly different approach and use images of humans to project directly onto a real-world object: [71] captures live images of a human face, warps them and projects them onto polystyrene humanoid heads, whilst [60] uses a depth based approach to reconstruct the human and project the results onto a life sized cylinder enabling 360° viewing.

3.3 3D reconstruction approaches

There exist a number of approaches for reconstruction of a 3D form, some such as laser scanners are readily available commercial products. However when considering 3D reconstruction it is important to also consider the context the acquired 3D model will be used within. In this thesis we are interested in the real-time reconstruction of the human form for telepresence applications, and therefore only approaches which are fast, or could be made faster are strongly considered.

At the highest level, 3D acquisition methods may be classified into active and passive methods:

3.3.1 Active 3D reconstruction methods

Active methods, such as laser scanning, and reconstruction by structured light [124], [115], [53] are methods which project light onto the object surface. The 3D form of the object is recovered by analysis of the way in which the projected light falls on its surface. These methods have been widely used to obtain accurate models of static scenes, for example a room. Systems for scanning the whole body are capable of capturing the shape of a person in a fixed pose. [114] and [3] used such methods to create human body models for later animation. More recently [128] used structured light to capture the moving human face in real-time.

Shape from structured light is an increasingly explored method of real-time 3D acquisition [105], [48], [129], it is capable of achieving very high frame rates compared with other approaches, and has the advantage of being possible with a single camera. There are two major drawbacks to the approach in the context of this research: A structured light pattern must be projected onto the participant being reconstructed which can interfere with the activity taking place. Only a partial 3D reconstruction is achieved, since the structured light pattern is projected from a specific location; this limits the orientation of the participant within the environment.

3.3.2 Passive 3D reconstruction methods (Image based modelling)

Passive methods are those requiring only natural images (such as those taken with a conventional still or video camera) of the objects or scene to be reconstructed. They are often referred to as image based modelling, or image based reconstruction (IBR) techniques. This allows for greater flexibility in scene composition, and less equipment. Unlike active methods they can also allow for retrospective 3D model generation from previously captured images as first demonstrated by [30] reconstructing architectural models from images. Image based modelling

techniques can be broadly classified into single camera, two camera (stereo), and multiple camera methods:

3.3.3 Single camera image based modelling

Single camera methods may be used to create a 3D model by manually selecting feature points and matching to a known form as explored by [70], in which a method for reconstructing human body posture from a single image is described. A single camera may also be used to construct a model from a sequence of images - For example several images of an object placed on a turntable rotating in front of a camera, as described by [125] in which profiles (also known as apparent contours) of an object under circular motion are used to estimate form using epipolar constraints. Similarly, 3D forms may be obtained by analysing the movement of an object in a sequence of images, such approaches are often called structure from motion. In 1984, [87] described such a method in "Spatio-temporal modelling based on image sequences". As the title suggests, these methods obtain information about the object missing in the first images from subsequent images. They are similar to multiple camera methods but instead of images being temporally consistent, they are spatially consistent.

Single camera methods are of limited value to this thesis for a number of reasons: Firstly, the temporal approach to structure from motion does not lend itself to dynamic objects such as the human form. Secondly, it is not feasible to manually identify feature points for mapping to a known form when several images per second will be generated - This could be addressed by automatically tracking initially identified feature points through sequences of images, indeed this forms the basis for video object tracking typically achieved using easy to identify markers. However, such methods usually employ multiple cameras due to the occlusion of features in a dynamic object between frames.

3.3.4 Two camera (stereo) image based modelling

Two camera methods usually replicate the animal binocular vision system where correspondences between two slightly offset images are used to triangulate the distance to that point. Such approaches are usually referred to as stereo reconstruction. There exists a wealth of research into such methods, perhaps because it is the most obvious starting point in analysing 3D vision. Two camera stereo reconstruction methods are not capable of creating a complete 3D model of an object unless mixed with other approaches encompassing more views of the object to be modelled. A number of such hybrid approaches exist, for example [23] used a combination of photometric stereo and object silhouettes to obtain the full 3D form of an object.

On the whole, stereo methods use photo correspondences to generate a depth map representing the distance to points within the field of view from the plane in which the cameras are situated. Finding these correspondences is a computationally expensive operation, and it can take many seconds to generate the output depending on the resolution being used. [7] uses a recursive matching algorithm to achieve 25 fps performance at 720 x 576 pixels. More recent approaches have achieved near realtime performance (15 fps) by using parallel processing on a GPU [61]. [62] Uses 48 cameras arranged in pairs of black and white cameras for stereo depth computation and a colour camera for texture acquisition on a single machine. Depth maps are combined to create a 360° model of the user achieving 5-7 frames per second.

Trinocular approaches add the constraint of a third camera in right-triangular [89], parallel [91], or surround configurations [86], to reduce ambiguity in stereo matching. [85] presents a real-time approach by parallelising over four processors.

3.3.5 Multiple camera image based modelling

The widest field of image based modelling methods are those involving multiple cameras - A number of cameras are placed around the object to be modelled and a variety of techniques can be employed to reconstruct the object geometry. These methods aim to reconstruct the entire object geometry, and are therefore well suited to the requirements of this research. At the heart of most multiple camera reconstruction methods are two fundamental concepts: Shape from silhouettes, and the visual hull.

The concept of shape from silhouette was first introduced by [9] This work proposed that a 3D form could be roughly geometrically approximated from the intersection of a finite number of silhouettes of images taken around the object. [67] later proposed the concept of the visual hull as a theoretical entity which could be constructed from the intersection of an infinite number of silhouettes of the object to be modelled. The visual hull is the maximal surface enclosing the form of the object, it is unable to represent any surface concavities. Since these important contributions, the visual hull and methods for constructing it have been the focus of much study in the field of computer vision. These methods largely fall into two categories:

- Volumetric approaches reconstruct the volume of the object using voxels to represent a region of space.
- Surface based approaches reconstruct the surface of the object, often using polyhedral geometry to do so.

There exists a third category that is able to generate an arbitrary viewpoint of the object from images, but does not create a 3D model, this is known as image based rendering.

3.3.6 Volumetric modelling from multiple camera images

Volumetric reconstruction approaches [20], [112] are those which are concerned with modelling the volume occupied by an object. [77] first described a method by which a volumetric "voxel" model of an object could be constructed from multiple views. A voxel may be regarded as a three dimensional pixel, and is a discretisation of three dimensional space. As such, the conceptual step from two dimensional image pixels to voxels is a logical transition that is easy to understand.

Early approaches created low resolution, fairly rough models. This is because extending the concept of a pixel into three dimensions significantly increases memory requirements, and correspondingly CPU overhead. To overcome this and allow the generation of higher definition models, the octree representation was conceived. Octrees are the three dimensional equivalent of the two dimensional quadtree and the one dimensional binary tree, and were first described as an efficient method for representing geometric models by [57]. [22] used octrees to represent a volumetric model constructed from three orthographic projections. The octree representation of a voxel object significantly reduces the memory required to store the object by progressive subdivision of space into smaller volumes where the object exhibits more detail. This not only reduces memory requirements, but intelligent approaches can be taken to construction of the octree from images which greatly increase the speed with which the object can be reconstructed. [96] described such an approach which enabled octree models to be built from silhouette images of an object. This method was improved upon by [117] who introduced an incremental refining process resulting in a near real-time reconstruction of a 64x64x64 voxel representation of an object.

The space carving algorithm proposed by [64] is a volumetric reconstruction method that introduces the concept of the photo-hull. This solves two drawbacks of the visual hull concept: Firstly the visual hull cannot be calculated for images where there are no background pixels, for example reconstructing an entire scene rather than an object placed within the scene. Secondly, since the photo

hull is constructed by means of adhering to photo-consistency constraints, surface concavities can be represented. However, the colour consistency check across multiple views is an expensive operation and therefore this method is not usually considered as a candidate for real-time applications.

Volumetric reconstruction can be easily broken down into computational tasks that can be processed in parallel, this is because voxel occupancy can be calculated independently each other. [123] propose such an approach in which computation is distributed over a network to be calculated by several computers. Silhouette images are captured by capture nodes, which are also responsible for calculating their projection into the volume to be reconstructed. The reconstruction volume is broken down into a number of slices beginning with the "base plane". Each silhouette is projected onto a base plane using homographic principles, and its projection onto successively higher planes can then be achieved using computationally less expensive scale and translate operations. Since the volume is now broken down into many slices which can be worked upon independently of one another, groups of slices can be sent to processing nodes on the network which calculate the intersection of voxels on each plane. The resulting intersected slices are then brought back together to form the overall voxel set which constitutes the visual hull. In this way, higher resolution voxel reconstruction is achieved through distributed processing. The paper presents this as a method for capturing human motions.

The advent of general purpose GPU computing has recently meant that highly parallelisable numeric problems such as volumetric modelling can be approached in a new way. Traditional CPUs and graphics hardware have followed different evolutionary paths over the preceding decades due to the differing requirements of a general purpose CPU compared with the specific requirements of a GPU. GPUs were originally invented to remove the burden of graphics processing from the CPU and associated memory. Typically GPUs comprise many processing cores compared with CPUs and offer very high bandwidth to video memory. New general purpose interfaces to GPUs such as CUDA (Compute Unified Device Architecture) and OpenCL (Open Computing Language) are now allowing processing

unrelated to graphics to be performed on GPUs which can yield orders of magnitude performance increases for problems of a parallel, rather than sequential nature.

Two volumetric reconstruction algorithms are implemented using CUDA and compared in [65]. One algorithm pre-computes the mapping of voxels to image pixel bounding boxes and stores these in a lookup table. The second does no pre-computation, but instead downsamples the silhouette images so that each voxel approximately maps to a single image pixel. Octree variants of both algorithms are also tested. The results show that although the first algorithm is faster for small voxel sets, the size of lookup tables soon becomes too large for the GPU memory. The second algorithm does not exhibit this problem since the downsampled silhouette images are significantly smaller than the lookup tables. It can also be observed that in both cases the octree variant of the algorithm is faster. This becomes increasingly apparent as the size of the voxel set increases. Furthermore, the lookup table algorithm can only be used when the cameras are stationary, whereas the downsampling approach could be employed for moving cameras and therefore provides the best general solution of the two.

3.3.7 Surface reconstruction from multiple camera images

Surface reconstruction, or polyhedral modelling, in which the surface of the object to be modelled is directly converted into polygons has been the subject of much less research than volumetric approaches. This is perhaps because accurately calculating the intersections of projected silhouette cones in three dimensions is difficult and computationally expensive to achieve. Most research has followed the well trodden path of firstly calculating a volumetric intersection and then, if necessary, using a surface recovery approach such as the marching cubes algorithm to determine a polygonal surface model.

The first attempt at directly generating a polyhedral 3D model of an object from multiple views was in Baumgart's 1974 PhD thesis, in which the shape from sil-

houette concept was first proposed. These silhouettes were extruded away from the camera to form a cone-like shape, and the 3D intersections of these cones used to approximate the object surface. Shortly afterwards, [8] used a method in which images of a simple rotating object were used to construct a wire-frame mesh. Image contours were used to identify second order irregularities, which were tracked from view to view in order to build up a network of nodes to be joined by arcs in the final model. Such an approach works reasonably for simple objects largely comprised of a small number of corners joined by straight edges, but probably would not for more complex objects and particularly those where the edges are largely curved with few irregularities that can be tracked. [37] describes a method for constructing polyhedral models from a rotating object using a laser range finder to determine the distance to the object at various points over the surface. Since accurate depth maps can be obtained from a laser range finder, this essentially becomes an exercise in polygonising a 3D form from progressively rotated depth maps of an object. In principle the reconstruction of a surface is possible from a set of apparent contours which can be obtained by circumnavigating the surface. This was first described by Giblin and Weiss (1987), and later generalised by Cipolla and Blake (1990). Conceptually, the problem of generating the visual hull of an object by intersecting projected silhouette cones is a simple one, and some general advances were made into the efficient intersection of 3D polyhedra by [18] and later by [100] [116] refine Baumgart's earlier work on polyhedral intersection by adding a post intersection mesh simplification step and creating triangular splines which are used to control model fitting, but this work is aimed more at reconstructed object recognition than efficient reconstruction.

It is not until the realisation that 3D silhouette cones are a special type of polyhedron that significant improvements in the intersection calculation speed is achieved. Since the silhouette projection into 3D has a fixed cross section, the operation of intersecting it with another silhouette cone can be reduced to a 2D operation. A face from the silhouette cone is projected onto a 2D silhouette image from another camera, the intersection between the projected face and silhouette are calculated, and this is projected back onto the silhouette cone face. The silhouette cone face can now be intersected with its reprojection from the silhouette intersection to

calculate the 3D intersection of that silhouette cone face with the other silhouette cone, thus reducing the cost of the intersection operation and solving numeric rounding problems simultaneously. This was first suggested by [110] in which photographs of trees are used to construct 3D models, and later improved by [79] in a more general work in which polyhedral visual hulls of objects from silhouettes are created efficiently.

[68] propose a method which does not require 3D polyhedral or volumetric intersections to be calculated. Apparent contours from camera images are used to sweep out a viewing cone which grazes the object tangentially to the surface, this is used to create a "cone strip" that continuously bounds the rim of the object from a camera's point of view. These cone strips are delimited by intersection curves between two visual cones (which are easier to calculate than a polyhedral intersection). Frontier points (where rims and cones intersect at a point) are used to construct a "rim mesh", the edges of which have a one to one relationship with the faces of the visual hull mesh. Since only the position of the camera centre is required, and not the position of the image plane itself, this method can be used under weak camera calibration. However, the method is only suitable for objects with smooth curved surfaces.

A hybrid approach to reconstruction is presented by [14] in which both volumetric and surface based methods are mixed to overcome some of the shortcomings of each. A surface reconstruction approach is employed to create an irregular grid of cells close to the surface of the object to be reconstructed, these are then "carved" using volumetric methods according to silhouette information. This results in an approach which maintains the robustness of volumetric approaches, whilst providing high precision, yet efficient results. [39] improve upon this method in Exact Polyhedral Visual Hulls (EVPH) by removing the volumetric step from their algorithm and replacing it with an algorithm that recovers mesh connectivity, resulting in an approach which is quicker and produces an exact polyhedron that is consistent with silhouette images. This is an important piece of work as it also removes the topological constraints introduced by numerical instabilities which forced other polyhedral approaches to restrict themselves to simple objects. In

[41] the EPVH algorithm is parallelised for processing in a network distributed setting.

A novel approach to polyhedral reconstruction from silhouettes is presented by [73] A geodesic sphere is created by successive subdivision of the faces of an icosahedron, the centre of this sphere is arranged to fall within the volume of the object being reconstructed. Rays are traced out from the centre point to the vertex edges of the sphere. Using silhouette information, the lengths of these rays are adjusted so that they fall within the silhouette cone projected from each image. The sphere vertices are then adjusted to match the ray lengths. In this way it is possible to reconstruct certain types of objects in a very efficient manner. However, this method only works for objects which are "spherical-terrain-like" (STL) Such objects are ones where the surface can be traced out from a point within the volume, spheres and cubes would fall into this category, but a teapot would not since the handle represents a secondary set of surfaces. Although the paper suggests that the human head is a good example of an STL object, human ears are not consistent with this opinion. Nevertheless, for STL objects excellent results are obtained in less time than with other leading polyhedral approaches.

[69] is a method complimentary to EVPH and producing very similar results Silhouettes are used to reconstruct the visual hull without the need to attempt 3D polyhedra intersections, instead the visual hull is built up in an incremental manner from features such as frontier points and intersection points that are derived solely from projective and epipolar constraints on the input silhouette images and resulting silhouette cone. Consequently the algorithm is able to reconstruct an exact visual hull from weakly calibrated camera images. Performance is similar to that of the EVPH algorithm. [40] significantly improves the performance of polyhedral modelling techniques by reducing the resolution of the silhouette contour polygon, which had previously been modelled through sub-pixel methods. At the same time, it is demonstrated that despite the lower resolution representation of the silhouette contour, the output is pixel consistent with input images. Huge reductions in the execution times of both Franco and Boyer's previous algorithm, and Lazebnik's algorithm are demonstrated.

3.3.8 Image based rendering

A third type of "modelling" from silhouette images from multiple cameras is widely researched. The method, first described by Matusik in "Image-Based Visual Hulls" [79] employs an algorithm that is entirely image based. No geometric model is created, and the desired viewpoint is directly rendered from image space coordinates of the reference images. Although this set of approaches has many advantages, including low computational cost and high photo-realism, no 3D model is ever created. This means that the algorithm must be run for each desired viewpoint, and that model interactions such as collision detection with virtual objects would not be possible.

3.3.9 Camera calibration

All IBR and VBR systems require a means of obtaining camera calibration data, for fixed camera systems calibration is usually an offline process that does not need to be repeated unless cameras are moved. The process of camera calibration involves determining the relationship between 3D world coordinates and the camera image plane [120]. A wide variety of techniques exist including calibration with a one dimensional object where one end of a line is fixed [131], a wand based calibration technique [80] calibrates multiple cameras with image correspondences across views, [44] uses multi-view stereo and bundle adjustment to derive camera calibration. [55] requires images of at least three spheres to calibrate a single camera, whilst [109] describes a technique in which multiple cameras are calibrated using a single image of a globe, determining both intrinsic and extrinsic parameters. [98] uses silhouette based multiple camera calibration based on error function derived from mutual consistency of silhouettes in pairs of views. [54] derive a common homography and relate silhouettes with epipoles under circular motion, similarly [111] uses an iterative optimization of shape from silhouette under circular motion to determine camera parameters by minimizing the difference between the projections of reconstructed visual hull and the silhou-

ette images using a GPU. [95] derives calibration from unsynchronised camera pairs on a network using silhouette information, and [94] calibrated cameras using only silhouette information in video streams.

3.4 Early research

The rest of this thesis discusses in more detail quality of 3D reconstruction (Chapter 4), the focus on improving temporal quality of the EPVH algorithm (Chapter 5), and a research platform developed for evaluating these in the context of telepresence (Chapter 6). The research conducted to narrow down approaches to 3D reconstruction in partial fulfilment of objective O1, research question Q1 and hypothesis H1 is summarised in this section.

3.4.1 Initial literature survey

The initial literature survey suggested that the visual hull [67] was a form that could be created at relatively low processing overhead compared to many other image based modelling approaches. The visual hull can be created using techniques falling under the shape-from-silhouette [9] category of approaches. This provided a direction for further literature searches and also some specific techniques that could be implemented to begin evaluating suitable methods for the reconstruction of humans. Specifically three categories of shape-from-silhouette visual hull algorithms were identified: volumetric reconstruction, surface based reconstruction and image based rendering (IBR). From these three categories, volumetric and surface based reconstruction were selected as they result in the formation of a model, whereas IBR does not. Formation of a model was considered to be an important part of meeting the objectives of the research as it was anticipated that resulting models would ultimately need to be displayed in ICVE settings. Image based rendering approaches do not form a model, and are designed for 2D displays to render a novel viewpoint of the object or scene be-

ing reconstructed. Whilst immersive displays such as CAVEs with stereo capable display walls could probably be adapted to work with IBR approaches, most are already capable of displaying 3D models based on vertices, surface polygons and textures. Furthermore, it was considered that IBR methods would be considerably more computationally expensive to display in immersive settings since, not only would the novel viewpoint need to be calculated for each of the display walls, but also in order to support stereo, the viewpoint of each eye would need to be calculated. Hence for a six sided CAVE, twelve novel viewpoints would need to be calculated, whereas model based approaches would only need to calculate a single model. This rationale eliminated IBR as a candidate approach during the feasibility study part of prototyping and evaluation depicted in Figure 2.1, and no concept implementation was ever created.

Other 3D reconstruction methods ruled out during the initial literature survey, or feasibility study were:

- Active 3D reconstruction methods, such as laser scanning and structured light. - These failed to meet the requirements of modelling from camera images, as well as creating a complete (360°) model, and were therefore ruled out during literature review.
- Single camera image based modelling. - Such methods failed to meet the requirements of being able to model a dynamic form. Due to the requirement to rotate the object being modelled, or the camera around the object, for a complete (360°) model. These methods are only able to reconstruct non-dynamic objects and were ruled out during the literature review.
- Two camera (stereo) image based modelling. - Whilst it was conceivable that stereo methods could be used to create a complete 360° model by using multiple sets of stereo camera pairs and merging the results into a single model, the literature revealed that stereo matching was a computationally expensive process. The ability of stereo methods to model concave objects was however compelling, and this had to be weighed against the apparently more efficient but convex only nature of methods forming the visual hull.

Stereo methods were ruled out during the feasibility study.

3.4.2 Pilot experiments

Pilot experiments (2.4, E0) provided the means by which several candidate shape-from-silhouette approaches and algorithms were implemented and evaluated. All the pilot experiments were conducted using the simulated setting described in Section 2.3.5.

Volumetric reconstruction

An implementation of volumetric reconstruction based on the space carving algorithm [64] was tested. In this algorithm, 3D space is divided into a grid of equal sided cubes, known as voxels. Each voxel maps to a pixel, or group of pixels in every camera's image plane. Voxels that fall inside the silhouette in every camera image plane form part of the object being reconstructed, whereas those that fall inside the silhouette for some or none of the camera images do not. Iterating over every voxel defining the 3D reconstruction volume forms a volumetric representation of the object being reconstructed where set voxels fall inside the volume of the object and unset voxels do not. The resulting data can either be displayed using volumetric techniques, or the surface can be extracted using an algorithm such as marching cubes [74]. Example reconstructions from the implemented volumetric algorithm are shown in Figure 2.3 showing reconstruction of the entire human body, and Figure 3.2a a human head.

In applications where the cameras remain fixed, volumetric reconstruction can be accelerated by building tables mapping each voxel to the pixels representing it in the camera image planes. However, this becomes memory intensive as the camera count or resolution is increased, and therefore limits the scalability of such approaches. The necessity to extract a surface from the volumetric data to enable polygonal model genesis and texturing adds a further processing burden to the

workload. Whilst the volumetric implementation continued to be used for our initial synchronisation experiment (2.4, E2), alternative methods better suited to our requirements were being sought.

Voxel contour intersection

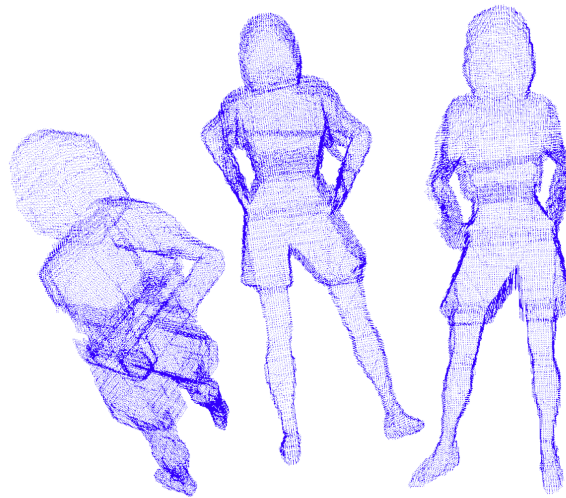


Figure 3.1: Point clouds formed using the Voxel Contour Intersection method in a simulated setting, from synthetic human model using virtual cameras.

One way in which volumetric reconstruction was a poor fit to our requirements was that it models object volumes rather than the surface between object and space. This can be remedied, as previously mentioned, by a post processing algorithm that steps around the boundary between set and unset voxels forming a polygonal surface. With this in mind we conceived our own novel algorithm for evaluating only those voxels that fall at the edge of the object. This was named "Voxel Contour Intersection" and worked using contours defining the edge of silhouettes as well as the silhouette images from the conventional space carving approach. Voxels in the final set were determined to fall at the surface of the object being reconstructed if they fell on the path of the silhouette contour in at least one image and within the silhouette in all other images. In this way we were able

to generate a point cloud representing voxels at the surface of the object, as shown in Figure 3.1.

This approach was no more computationally efficient than calculating the voxel occupancy in the space carving approach, but appeared to get a step closer to a polygonal representation without using marching cubes. Deriving a polygonal mesh from the point cloud was the next problem to be solved along this line of research. In fact marching cubes would have been no use here, because the resulting point cloud was sparse compared with a voxel set, due to the discrete nature of pixels falling along the silhouette contour. Approaches such as delaunay triangulation [31], [29] seemed an appropriate fit, and software libraries offering it were readily available. However, whilst delaunay triangulation of the sparse point cloud formed a cleaner surface than marching cubes, the performance was no better.

Direct surface reconstruction

Literature revealed that researchers had used "hybrid" methods [14] in which the form is roughly calculated volumetrically before refinement by following contour edges. However, it was becoming clear that an approach able to resolve the surface of the object directly from silhouettes without first performing a volumetric reconstruction was required. Such an approach would certainly reduce memory overhead, and could possibly be computationally more efficient.

One such technique, the Adaptive Dandelion [73], was implemented and tested. The approach begins by constructing the triangles forming the surface of an icosahedron such that the centre of the icosahedron's volume falls within the silhouettes in all images. Rays are then traced out from this central point along the line to each of the icosahedron vertices. The rays are mapped to every camera image and trimmed to the shortest length that falls within all silhouettes. Icosahedron vertices are adjusted to the 3D position at the end of the trimmed ray. The adaptive aspect of the algorithm is that each triangle can be recursively subdivided as required to increase surface detail. The algorithm was found to be very efficient, and

lacked the memory overhead of volumetric approaches. However the ability of the approach to only model spherical terrain like (STL) objects meant that whilst it was reasonable for modelling the human head, it was unable to model limbs. Figure 3.2b shows a human head reconstructed using the adaptive dandelion method. Notice how in comparison to the same head reconstructed by other methods in that diagram the adaptive dandelion head lacks the top of the ears, this is due to their geometry not conforming to the STL requirements of adaptive dandelion.

A more general purpose approach, capable of reconstructing the visual hull was found in the Exact Polyhedral Visual Hulls (EPVH) algorithm [39]. The method works by projecting camera silhouette contours into 3D space to form viewing rays, consecutive pairs of which from a particular camera form an infinite triangle. The rays are projected into every other camera and intersections with contour edges along the ray recorded. Using these intersections the minimal spans passing across the surface of the visual hull can be determined. From these initial spans, each defined by two vertices, the algorithm describes a method by which the exact connectivity to neighbouring span vertices can be determined to complete a watertight polyhedral mesh. Finally the surface polygons can be extracted by following the connected vertices of the mesh and referring back to the infinite triangles formed at the start.

3.4.3 EPVH implementation and development

The EPVH algorithm appeared to provide what had been sought through concept implementation and comparative study of other shape-from-silhouette algorithms, i.e. a method of deriving a polygonal surface of the visual hull directly from camera silhouettes without any intermediate steps. Little was known, however about the algorithm's performance compared to other approaches. Furthermore, since a reference implementation was not available, the algorithm had to be implemented from scratch based on details given in the literature. This was not a straightforward exercise, and some areas were left open to interpretation. For example, the method by which searching for intersections of an arbitrary 3D line with camera

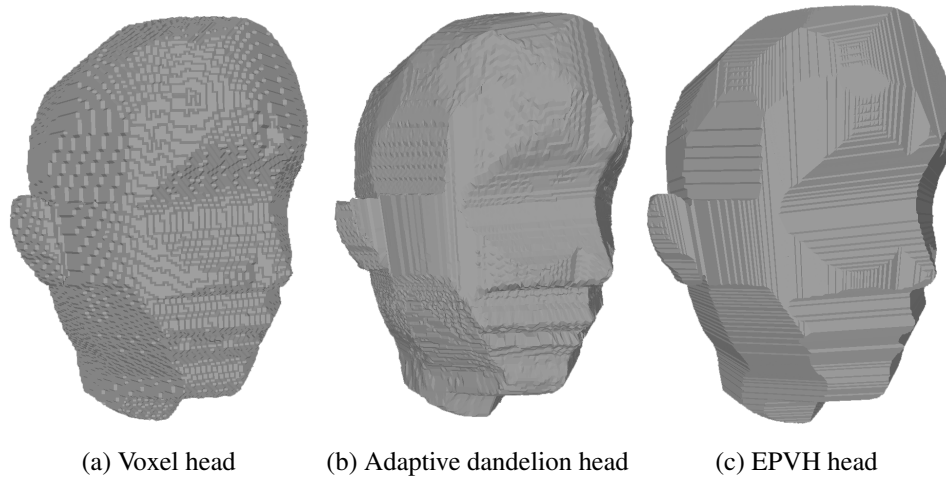


Figure 3.2: Side by side comparisons of synthetic human head reconstructed from virtual camera images by different 3D reconstruction algorithms.

image silhouette contours may be accelerated. Our initial EPVH implementation was sequential, and did not execute quickly enough for real-time performance at the desired quality. An example untextured surface reconstruction from the initial EPVH implementation is shown in Figure 3.2c.

3.5 Conclusion

This chapter began with a brief review of human collaboration through technology, focussing on the conveyance of eye gaze through video conferencing and ICVEs then human modelling for remote collaboration. This provided the setting for the research in terms of the recent developments in technology and how humans interact through it. From this is clear that it is only in recent years that techniques and technology have become sufficiently developed to achieve the goal of visually faithful real-time 3D telepresence systems.

The subject of 3D reconstruction was then reviewed to search for approaches which might be suitable for real-time modelling of the human for telepresence

applications. The review revealed that there has been a significant amount of research into volumetric 3D reconstruction techniques, but much less research in the area of direct surface based techniques. A number of approaches were implemented and compared during pilot experiments (Section 3.4), providing a critical analysis of state-of-the-art approaches from literature. This enabled narrowing of the focus to the EPVH algorithm for the remainder of the research.

Volumetric techniques have recently been accelerated by use of multi-core GPUs to achieve real-time frame rates, and both volumetric and surface based approaches have been accelerated using network distributed processing. Only one instance of a real-time surface based technique was found, however, [41] in which network distributed processing is used to achieve real-time frame rates. There appears to be no research into the use of modern multi-core CPUs or GPUs for the acceleration of surface based approaches to 3D reconstruction, and this therefore provides a gap in the literature on which Chapter 5 will focus.

Chapter 4

Quality of 3D reconstruction

In this chapter the quality of 3D reconstruction is studied. Three quality measures are defined; visual, spatial and temporal quality. The relationships between them and the characteristics of a 3D reconstruction system that influence them are discussed. Experiments are described in which temporal consistency of the input camera images is relaxed and the result on the qualities of the output model studied.

The study of camera image synchronisation presented in this chapter has been published in [\[33\]](#).

4.1 Research questions and hypotheses

This chapter aims to address the following objective:

- **O1:** Determine the requirements and approaches to 3D reconstruction suitable for a real-time telepresence system.

4.1.1 Hypotheses

- **H2:** Relaxing constraints on temporal consistency of inputs will result in a degradation of spatial and visual quality in the case where the subject is moving.

4.2 Qualities

The quality of a 3D reconstruction system is in part the faithfulness of the output 3D reconstruction to the original object being reconstructed, in much the same way as one might measure the quality of a photograph in terms of its faithfulness to the scene or objects being photographed. It is often stated that "a photograph never lies" and whilst this statement may be true at the highest subjective level, the measure of quality of a photograph extends beyond the simple notion of likeness. Lens choice, focus, depth of field, shutter speed, lighting and the speed of the film or sensor used all affect aspects of the quality of a photograph. In much the same way, choices made about the components used, and the way they are deployed, impact upon the quality achieved by a 3D reconstruction system.

For the purposes of quality analysis, we define three measures of output quality for a 3D reconstruction system: Spatial, visual and temporal, as follows:

4.2.1 Spatial quality

Spatial quality is the faithfulness of form of the reconstruction to the object being modelled. Completeness of the reconstruction, and accuracy can be regarded as independent measures of spatial quality. A particular reconstruction approach might result in high accuracy modelling of only part of an object. For example, the use of binocular stereo methods to model the human face can lead to a spatially accurate model of the face itself, but fails to model the rest of the head or body.

In the context of a 3D telepresence system, in which people are able to move around each other and use body language to communicate, completeness of model is important. The number of fingers reconstructed on a human hand surely falls into the category of model completeness, but what about the wrinkles on a person's forehead when they frown? Here the difference between completeness of model and spatial accuracy become less obvious; on one hand a lack of wrinkles could be described as an incomplete model, but on the other it could be attributed to a lack of spatial accuracy in the reconstructed surface. For the purposes of this research, we choose to differentiate between spatial accuracy and model completeness as follows:

Spatial accuracy is the metric that measures relative distance on the reconstructed model compared with the object being reconstructed. For example the ratio of arm to leg length in the reconstruction should be the same as in the human being modelled.

Model completeness is the measure of presence or absence of features in the reconstruction compared to the object being modelled. By this definition, wrinkles on a person's forehead, or lack of them, is a function of model completeness, rather than spatial accuracy.

Defining spatial accuracy and model completeness in this way also enables the adoption of model completeness as the overriding measure of spatial quality, since the accuracy of 3D reconstruction becomes an easily quantifiable measure that is

largely governed by the quality of camera calibration.

Spatial quality of the visual hull

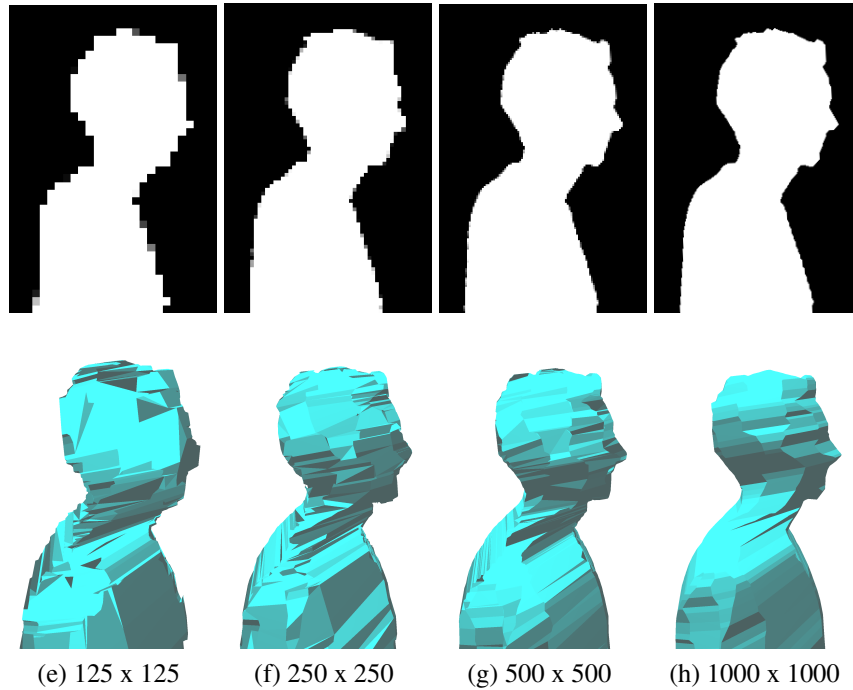


Figure 4.1: Impact of camera image resolution on the spatial quality of the visual hull. Full camera image resolution is shown after the sub-figure caption. Silhouettes shown are a sub-region of a full camera image. Models shown are part of a complete model. The region was selected in the image and model to illustrate increasing spatial quality of facial features with camera image resolution.

The ideal visual hull is a theoretical entity formed from the intersection of an infinite number of silhouette cones derived from cameras surrounding an object. Therefore all real-world shape-from-silhouette implementations using a finite number of cameras can only form an approximation of the visual hull. Since the theoretical visual hull is unable to model concavities, this limitation also applies to all shape-from-silhouette approaches that do not provide concavity modelling extensions.

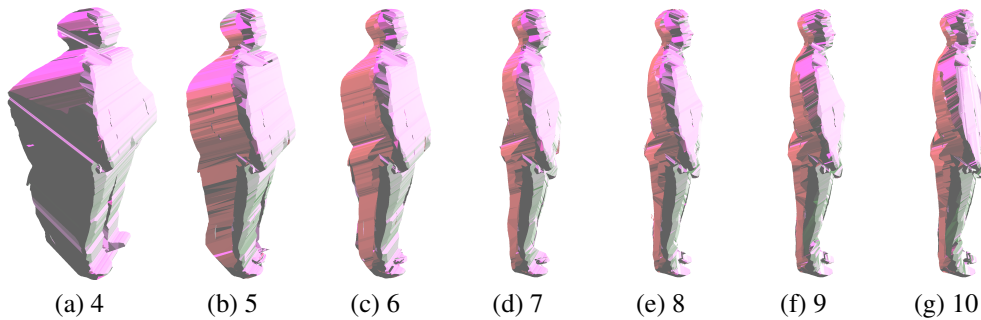


Figure 4.2: Impact of number of cameras on the spatial quality of the visual hull. The number of cameras is shown after the sub-figure caption.

In terms of the visual hull and its formation, spatial quality is largely influenced by the number and resolution of cameras. Higher resolution cameras can form silhouettes that better represent the contour of the object being modelled from a particular viewpoint as shown in Figure 4.1. The more cameras used in reconstruction, the greater the number of viewpoints on the object and consequently the better the constraint of the object's form (Figure 4.2). Increasing camera resolution or count, however, does not necessarily lead to an increase in spatial quality; the threshold of possible quality is defined by the object under reconstruction.

- Spatial quality increases as camera count, resolution and calibration quality increase.

4.2.2 Visual quality

Visual quality is the ability of a reconstruction approach to achieve a resemblance to the original object. This is certainly the most subjective of the quality measures. In two dimensions the visual quality of a copy of a picture could be measured in terms of the variance of pixels in the copy compared with the original. Similarly, in three dimensions, the visual quality of a reconstructed model could be measured against the input images. This is similar to the photo-consistency measure used by some reconstruction techniques to refine a model. However, the problem here

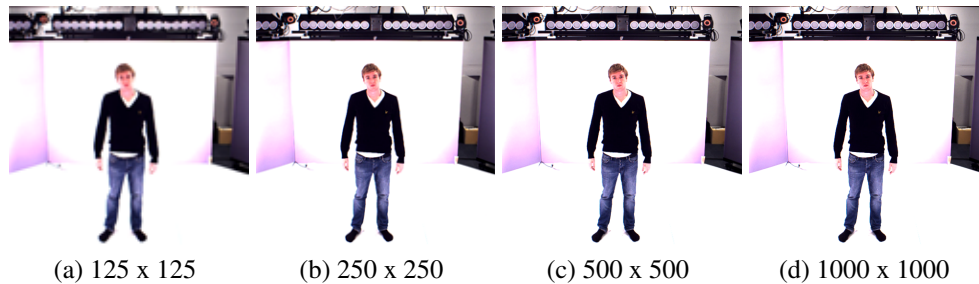


Figure 4.3: Impact of camera image resolution on the detail captured, lower resolution fails to capture important features such as the eyes. Camera image resolution is shown after the subfigure caption.

is that poor spatial quality will necessarily result in poor visual quality; therefore visual quality is dependent on spatial quality. Visual quality will also tend to increase as camera image resolution increases; more pixels will define an area being reconstructed, leading to a better likeness. Figure 4.3 illustrates how small but important features like the eyes can be poorly defined at lower resolutions.

- Visual quality increases as spatial quality and camera resolution increase.

4.2.3 Temporal quality

Temporal quality of 3D reconstruction can be summarised as the ability to quickly and consistently deliver the reconstructed model from the input images. An increase in time taken would constitute a decrease in temporal quality. Temporal quality can be subdivided into latency and frequency. The latency is the overall time taken to reconstruct the model, whereas the frequency is the number of reconstructions that can be achieved per unit time. Temporal quality is inversely proportional to latency and directly proportional to frequency. This means that a higher temporal quality is achieved by delivering output at a low latency with a high frequency.

Whilst it may seem that there is a direct relationship between latency and fre-

quency, they are actually independent measures of temporal quality. For example, a high frequency could be achieved but with a long delay (latency) in the output. Such a result could be achieved through latency in the transmission of camera images to the reconstruction algorithm. Another cause of latency can be the pipelining of processes in the reconstruction system aimed at achieving higher frequency output. Pipelining is a method of achieving stream level parallelism in which a sequence of processes are executed simultaneously on multiple processors each acting on a different frame in the stream. For example, one processor might capture a new image from the camera whilst another performs background segmentation on the previously captured frame.

- Temporal quality increases as latency decreases and frequency increases.

4.2.4 Relationship between spatial, visual and temporal quality

The three quality measures have been described, and defined in terms of the characteristics that affect them: namely camera characteristics for spatial and visual quality: and latency and frequency for temporal quality. It has already been noted that visual quality depends to some degree on spatial quality. The relationship between camera characteristics and temporal quality therefore need to be established to understand the overall relationship between these three qualities. This relationship can differ depending on the 3D reconstruction approach used, so at this stage we seek to define a generalised relationship. In general, as the number of pixels per camera and the number of cameras increases, there is an increased processing burden on the 3D reconstruction algorithm. Increasing the processing required increases the latency and decreases the frequency, hence decreasing temporal quality. Therefore, in the general case, temporal quality decreases as camera count and resolution increase. Since increasing camera count and resolution can lead to higher spatial and visual qualities, the general case can be extended to define a direct relationship between spatial, visual and temporal qualities as follows;

achieving higher spatial and visual qualities decreases temporal quality. Since temporal quality is determined by the performance of 3D reconstruction, the logic of this relationship can be reversed as follows:

- Increasing performance will enable higher spatial and visual qualities per unit time.

4.2.5 EPVH algorithm and quality

In terms of the EPVH algorithm, the general case that spatial and visual quality are proportional to camera resolution and count can be defined more specifically. Silhouettes form silhouette contours that define the edges of objects in camera images. Each contour is made up of an ordered series of points progressing around the contour, pairs of which define a straight line segment. For shapes defined by straight edges in camera images, the number of points defining the contour will be fewer than that required to define a curve. This forms the notion of contour complexity which, when combined over all of the cameras, defines the input complexity. Unlike volumetric approaches to shape-from-silhouette, in which input complexity is defined by resolution, for EPVH it is the contour complexity that becomes the impacting factor on spatial and temporal quality. For complex or curved objects, increasing camera resolution will generally increase contour complexity, and so the general rule still applies. However, for simple objects defined by straight edges, or those occupying a small region in the camera image plane, the general rule may not apply. Visual quality increases with camera resolution in the same way as in the general case, regardless of contour complexity. An object such as a cube may be defined by contours of low complexity but have an intricate surface texture, the appearance of which improves as camera resolution increases.

4.2.6 Quality requirements for telepresence

An essential part of this research is understanding the requirements for human communication through the medium of telepresence in terms of quality. The best balance of visual, spatial and temporal qualities needs to be sought to provide an effective system that does not hinder communication and feels as natural as possible. Many of the expectations are already defined by existing mediums such as video conferencing and immersive virtual reality. Video conferencing provides a visually faithful means of communication, enabling people to engage in meaningful non-verbal communication to supplement conversation; facial expressions can convey feelings and emotions that cannot be supported through immersive VR. Virtual reality provides a spatially coherent means of interacting, enabling participants to move around each other, point at each other, and generally direct each other's attention within the shared setting. However VR can be a confusing mixture of body tracked gestures conveyed through the iconic representation of an avatar that is an otherwise unrealistic soulless entity.

The requirements for 3D telepresence can be derived from combining the most effective aspects of video conferencing and immersive VR to provide a visually faithful and spatially coherent representation of the human. To quantify what these requirements are we must analyse the important aspects of each in terms of human communication, focussing on those requiring the most detail to represent. For example, the human eyes are a relatively small feature compared with the mouth; a resolution capable of representing the main features of the eyes (white, iris, pupil) will provide more than adequate detail of the mouth to convey a smile. Similarly, sufficient spatial quality to ascertain where a person is pointing will also provide for the distinction between them waving or shaking their fist. These requirements are harder to define in terms of the camera system used to capture humans for 3D telepresence as they require first defining the distance away from a camera that a human can be situated before determining the resolution that would be sufficient. Instead, the quality requirements are guided by current technology, which provides cameras of resolutions capable of capturing detail such as the eyes up to a certain distance away, and at a certain number of frames per second. In this

sense it is the manner in which this technology is deployed that will determine the quality of output achieved beyond a certain lower bound. Similarly, it is beyond the scope of this research to perform specific tests into non-verbal communication through telepresence, the aim is to provide a platform through which meaningful tests can be carried out.

4.3 3D reconstruction system components impacting upon quality

4.3.1 Camera choice

The number and resolution of cameras and their impact upon quality has already been discussed at some length. Other camera characteristics that may impact upon quality are the choice of lens used and the method of triggering and obtaining frame data.

Camera lens choice

High end cameras often provide the ability to change the lens used. Normal lenses are those where the focal length is greater than the longest edge of the image plane. Wide angle lenses are those where the focal length is smaller than the longest edge of the image plane, resulting in a wider field of view. In 3D reconstruction it can seem compelling to deploy wide angle lenses, providing each camera with a wider view on the reconstruction area and consequently expanding the working volume within which objects can move. However, inexpensive wide angle lenses can distort the camera image (Figure 4.4, turning straight edges into curved lines which, left uncorrected, will adversely affect the spatial and visual quality of the reconstruction. Camera lens distortion can be modelled and camera images can be corrected in software to remedy this. Whilst silhouette contours can be cor-

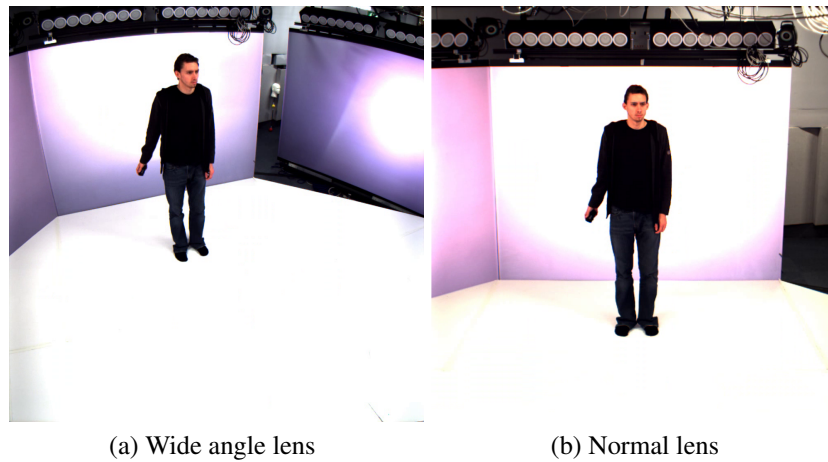


Figure 4.4: Lens choice can lead to image distortion. (a) A wide angle lens gives rise to barrel distortion, making straight lines appear curved. (b) A normal lens where straight edges appear straight.

rected with a relatively low processing overhead by simply displacing points by the corresponding adjustment in the lens distortion model, camera images used for texturing must be corrected on a per pixel basis. Such image correction requires processing, and can therefore lead to a degradation in temporal quality of the overall system. High end wide angle lenses are available that provide both a wide field of view and do not exhibit barrel distortion, however these were not available for the purposes of this research.

Camera trigger and distribution method

Some cameras provide a mechanism via which hardware synchronisation may be achieved. Typically this takes the form of a connector through which a rising or falling edge signal triggers image capture; enabling all cameras within a system to be perfectly frame synchronised. Reliance on this capability significantly reduces the range of cameras that can be used for 3D reconstruction since few cameras on the market offer it. Without a hardware trigger, perfect camera frame synchronisation is arguably impossible. Section 4.4 describes the findings of experiments performed to understand how the lack of frame synchronisation impacts on quality.

The distribution method used to provide images from cameras to the reconstruction algorithm will have some inherent latency and hence affect temporal quality of the overall system. Cameras will have an interface through which images can be obtained, which could be ethernet, USB or firewire. A reconstruction system, relying on cameras directly connected to the machine running the reconstruction algorithm, will be limited in terms of the possible number of cameras in the case of USB or firewire, since bus contention and overall interface bandwidth places a ceiling on the number of possible cameras. Scalable systems will generally provide machines dedicated to the purpose of obtaining and distributing camera images to the reconstruction algorithm. The design of such a camera acquisition stage in 3D reconstruction is outside the scope of this thesis, but has implications for the quality of reconstruction and also the temporal consistency of inputs that are discussed further in Section 4.4.

4.3.2 Camera placement

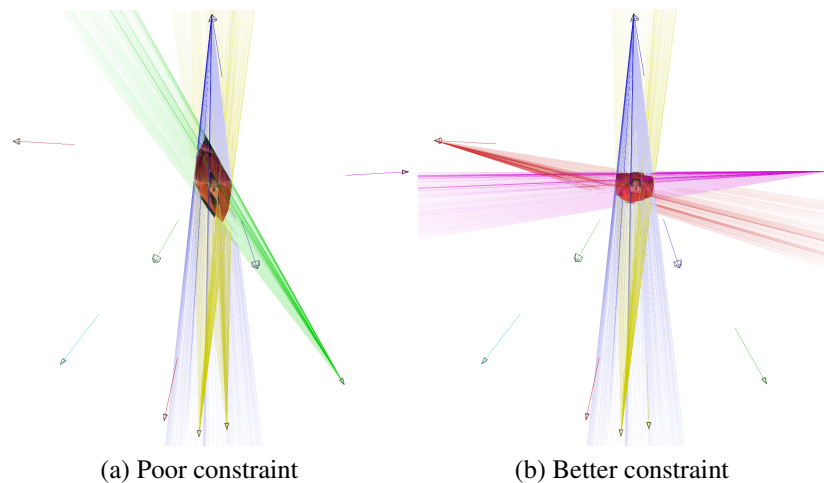


Figure 4.5: Impact of camera placement on spatial quality of the visual hull. (a) Imbalanced placement of 4 cameras provides insufficient constraint. (b) Balanced placement of 4 cameras leads to better constraint.

The placement of cameras and the shape of the object being modelled need to be considered. As a general approach, positioning cameras so that they point towards

the object from as many different directions as possible better constrains the visual hull than clustering them together. Figure 4.5 demonstrates this principle. In both cases a subset of 4 cameras from a set of 10 is selected for reconstruction. 4.5a shows the effect on that selecting cameras from insufficient directions has on the reconstructed form. 4.5b shows that, by swapping 2 of the 4 cameras for ones pointing at the object from more diverse directions, better form constraint is achieved.

Objects with holes passing through them can be modelled by shape-from-silhouette, but these holes must be clearly visible from at least one camera and also correctly masked by background segmentation. Small holes can be problematic in terms of camera placement, particularly if the size of the hole is much smaller than the thickness of the object it is passing through. Such a scenario necessitates careful alignment of camera to hole to ensure that the background is visible through the hole.

Moving objects provide further challenges for 3D reconstruction systems, particularly in respect of camera positioning. Any camera placement carefully considered for particular regions of the object being modelled, becomes meaningless if the object is to be allowed to move freely. Employing the strategy of placing cameras so that they point at the object from as many different directions as possible can only be achieved for a region within which the object is allowed to move. Furthermore, as the object moves within this region, the size of it within camera image planes will vary as the object moves towards or away from a particular camera. This makes it difficult to make strategic decisions about camera image resolution and the complexity of objects being modelled. Providing cameras are pointing towards the object from many different directions, however, this problem can be mitigated by the object's size increasing in one camera whilst it diminishes in another.

Camera placement and visual hull set definitions

By the classic definition of the visual hull, regions falling within it are defined as those projecting within the silhouette in all camera images:

$$VH = \bigcap_{\text{Images}} \left(\bigcup_{\text{Silhouettes}} \left(\bigcap_{\text{Contours}} V \right) \right)$$

where: VH is the visual hull formed, and V is the viewing cone formed by a contour

This definition gives rise to a bounding volume defined by the intersection of regions visible by all cameras simultaneously. An object overlapping the edge of this volume will be clipped in the reconstruction, and those falling outside of the volume will not be reconstructed at all. [39] defines a new visual hull set definition in which the visibility domain of each camera is considered:

$$VH^C = \bigcup_{\text{Images}} \left(\bigcap_{\text{Silhouettes}} \left(\bigcup_{\text{Contours}} DV \right) \right)$$

where: VH^C is the visual hull complement formed, and DV is the visibility of viewing cone complement relative the the image from which it is formed.

Considering the visual hull or its complement are the same, because the surface of the hull separates the region defined by regions falling inside or outside the intersection of silhouette cones. The difference lies in the complete visual hull set that is formed from all camera images in the first case, or only those visible in the second case. Regions projecting within the silhouette for all cameras, in which that region is visible, contribute to the visual hull in the second case. Using this definition relaxes the constraints on camera placement, allowing objects under reconstruction to be partially visible by a single camera, or completely outside the visible area of one or a subset of cameras.

4.3.3 Camera calibration

Camera calibration is the process that determines the relationship between the position of objects in the real world and their mapping onto the camera 2D image plane. A number of different processes are described in the literature (Section 3.3.9), resulting in the output of a set of 3×4 projection matrices P , that map 3D world coordinates to 2D coordinates on the camera image plane through the relationship in equation A.0.2 Through the process of camera calibration, two measures of calibration accuracy may be defined. The 2D pixel re-projection error defines the RMS error in camera image pixels of projecting 3D coordinates onto the camera image plane using the matrix P . The 3D reconstruction error is the spatial error in the position of a 3D point, reconstructed from a pair of 2D points, each on a camera image plane projected into 3D space. The pixel re-projection error can be used to determine the calibration quality of an individual camera. Values under half a pixel are desirable for all cameras in a set. Values over a pixel can result in the wrong camera image pixel being selected for a particular point in space, giving rise to spatial and texture distortion in the reconstruction.

4.3.4 Background segmentation

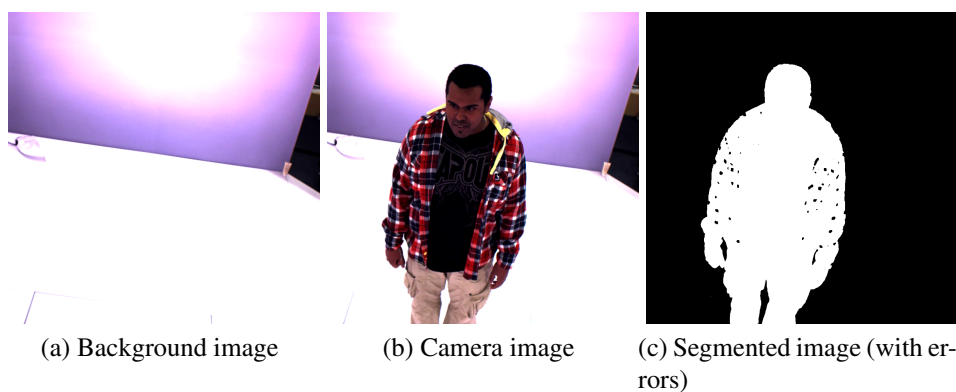


Figure 4.6: Background segmentation errors in (c) resulting from colour matching the background reference image (a) in camera image (b).

Reliance on shape-from-silhouette to form the visual hull means that the quality of the visual hull formed will be defined, in part, by the quality of silhouettes provided to construct it. Silhouettes are created from camera images by the process of background segmentation for which there are a number of approaches. Whilst background segmentation itself does not form part of this thesis, it is important to understand the factors that may affect segmentation and the resulting quality of silhouettes. A typical approach to background segmentation is to capture images of the background prior to object insertion. When later images with the object in place are used, pixels that differ from the background reference images form the object and become part of the silhouette, the remaining pixels form the background. Whilst in theory, and in simulation, this simple approach works, in practise there are many caveats that can lead to segmentation errors:

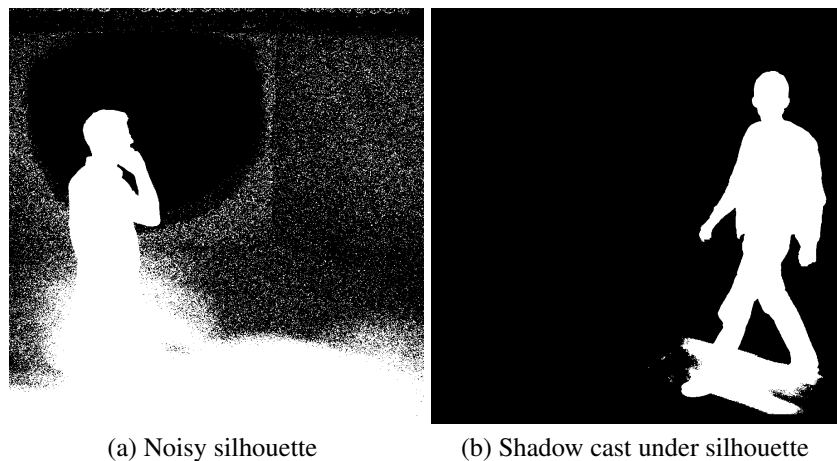


Figure 4.7: Background segmentation errors lead to poor silhouettes in the presence of (a) noise and (b) shadows.

- If the colour of a pixel is used to define how that pixel looks in the background, a pixel in the object that is the same colour will be erroneously designated as background, forming a hole or edge discontinuity in the silhouette as shown in Figure 4.6. A classic example of colour being used for background segmentation is in media and film production; known as green-screening. The weather man ensures he is not wearing any green clothes

and is filmed in front of a green surface presenting the weather. Maps reflecting the context of his weather report can then be dropped behind him by replacing pixels that only have a green component with pixels from the map.

- Camera image noise results in minute fluctuations in the colour or intensity of a pixel which can incorrectly designate background pixels as foreground pixels and vice-versa, as show in Figure 4.7a. In practise the incorrect assignment of background pixels as foreground in shape-from-silhouette approaches does not usually result in an error in reconstruction as these pixels will generally fall outside of the silhouette cone intersections of real objects within the scene, and therefore will be eliminated from the reconstruction. However, pixels incorrectly designated as background that fall within the silhouette of the object being reconstructed will result in reconstruction error in the form of holes passing through the object.
- Variation in lighting of the scene, which is particularly prevalent in outdoor settings, results in the reference background images captured before object insertion to no longer be a valid representation of the background. Parts of the background, or in the worst case the entire background becomes foreground in the silhouette, and the true shape of the object is lost as a result.
- Shadows cast by the object being modelled are a major problem in background segmentation. The reference background image pixels will differ from those where the shadow is cast and therefore get designated as foreground pixels in the silhouette. Whilst only the brightness of pixels falling in shadow will vary from reference images, and therefore colour can be used to differentiate shadow from object, the fact that shadows follow the object around the surface on which it is moving can make it difficult to determine where the object ends and shadow begins (Figure 4.7b).

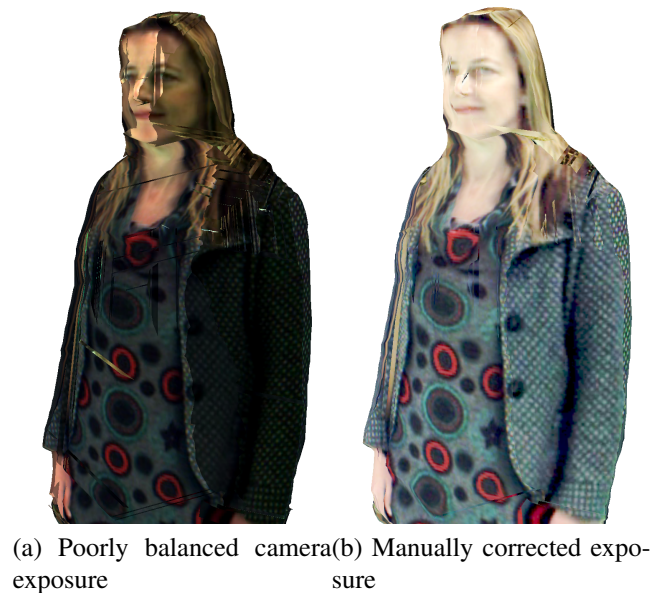


Figure 4.8: Poorly matched camera settings can lead to poor visual quality through regions textured with different cameras appearing to have different brightness and colours. 4.8a A reconstruction from poorly balanced cameras. 4.8b The same reconstruction where the images have been manually corrected.

4.3.5 Camera exposure and colour balance

Adjusting the exposure and colour balance of cameras within a set is essential for obtaining good visual quality output. Poorly balanced camera exposures will result in some camera images appearing brighter than others, which leads to a texturing discontinuity, as shown in Figure 4.8a. Texture blending can be used to blur the discontinuity over a region, but such approaches can lead to blurring of important features too. Differences in colour balances between cameras in a set can cause similar problems, and even a set of cameras from the same manufacturer and of the same model can exhibit colour variance. High end cameras often allow for exposure and colour balance adjustments to be made in software, which can be part of the camera calibration process.

4.3.6 Reconstructed model transmission

Depending on how an end-to-end telepresence system is designed, following reconstruction, the reconstructed model will either need to be displayed or transmitted to a remote location for display. One possible system design is that all the camera images are transmitted to the remote location, delegating reconstruction to the receiving end. Another design is that reconstruction is performed locally and the reconstructed model transmitted. Whilst the specifics of an end-to-end system design are beyond the scope of this thesis, the implications are briefly discussed here. The system that reconstructs from local cameras and transmits the resulting model is the preferred design resulting from this research. The reconstructed model takes the form of a polygonal mesh that can be compressed using a number of techniques [45]. Textures applied to the model will form a subset of data derived from camera images, and should therefore require less transmission bandwidth than the whole set of camera images.

4.3.7 Experiment E1 - Comparing the latency of an ICVE and video conference

In the following experiment, the latency of an ICVE and video conference were both measured. The intention of the experiment was to derive results for guiding the latency expectation of a final end to end 3D reconstruction system. The method and results are briefly summarised here, full details of the experiment can be found in [103].

Introduction

Latency can cause confusion between participants in a technology mediated conversation. The tolerance for latency in verbal communication is around 150 msecs, but this tolerance is less clear in video conferencing where the audio and video can

become out of synchronisation. Latency requirements in connected ICVEs is even less clear, especially because it is difficult to measure. The aim was to gain an insight into how an ICVE compared to a high quality video conference in terms of true end-to-end latency of the visual representation only, with the long term goal of setting latency expectation for future 3D telepresence systems.

Method

Measuring true end-to-end latency of a communication system can be a challenging problem, especially where the ends are disparately located. Whilst round trip time is relatively easy to measure, this may not give a clear indication of the latency in the presence of an asymmetry. Network timings when clocks are synchronised can give a better idea of the on way latency, but does not take into account the additional latency incurred by the capture and display systems at both end.

In order to measure latency for video conferencing and ICVE systems we used the same approach for both: 2 video cameras were frame synchronised at the start of recording and one moved to each end of the end-to-end system. A local participant was filmed moving his arm up and down in a regular motion by one, and his embodiment (either video image or avatar) was filmed by the other. Latency was measured by analysing the captured video footage from both cameras side by side and using the synchronised frame count to determine the time offset between points at which the participant's hand passed a particular point in the footage on both ends.

Results

The end-to-end latency measured between ICVEs shows an asymmetry: In one direction we measured a mean of 605ms, and in the other 414ms. This can be attributed to the different body tracking and display technologies used at either end. The end-to-end latency of the video-conferencing system measured a mean

of 120ms when viewed through a 60Hz plasma screen, and 100ms when viewed through a 102Hz DLP projector.

4.4 Temporal consistency of inputs and quality

In this section we discuss the effect that relaxing the constraint on temporal consistency of inputs has on the quality of 3D reconstruction. Inputs to a 3D reconstruction system usually take the form of synchronised camera images, obtained from cameras triggered using a hardware synchronisation signal as described in Section 4.3.1. The effect that removing hardware synchronisation has on 3D reconstruction quality appears to be un-researched in the community. Cameras featuring a hardware synchronisation interface are not common, and tend to exist only at the high end of the market. The ability to use general purpose cameras without hardware synchronisation would enable the use of a much wider variety of cameras in 3D reconstruction systems, including USB webcams.

There are three possible synchronisation scenarios for cameras in a 3D reconstruction system:

- Hardware synchronised via an external synchronisation signal.
- Software synchronised via a software trigger system.
- Free-running, the cameras are unsynchronised.

Figure 4.9 shows the three synchronisation schemes and the expected level of frame synchronisation from each.

Figure 4.9a shows frames triggered by a hardware signal. The start of each frame acquired is perfectly coincident in time. Grabbing the latest frame the has been completely acquired yields frame 2 for all four cameras.

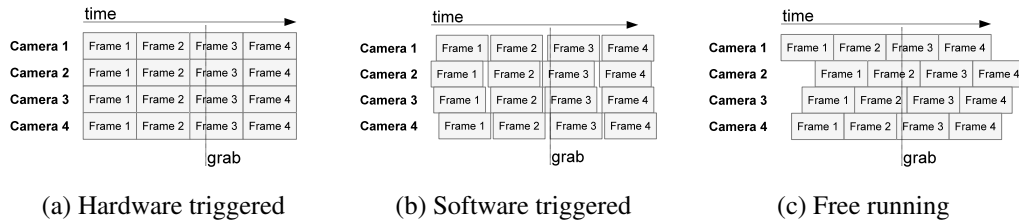


Figure 4.9: Three different camera image synchronisation schemes. Grabbing the latest complete frame for each scheme yields (a) [2,2,2,2] (b) [2,2,2,2] (c) [2,1,1,2]

Figure 4.9b shows frames triggered by a software signal. Notice the variable gap between the start of each frame acquisition, this is caused by jitter in network transmission of the trigger signal or latency in processing the signal on receipt. Grabbing the latest completely acquired frame yields frame 2 for all four cameras.

Figure 4.9c shows frames acquired through free running cameras. There is no alignment between the start of frames from different cameras, but the maximum offset between the newest frame between the whole camera set will be a complete frame period. Grabbing the latest completely acquired frame results in frame 1 for cameras 2 and 3, and frame 2 for cameras 1 and 4.

Three experiments were performed to better understand how each of these synchronisation schemes might impact upon 3D reconstruction quality.

4.4.1 Experiment E2 - Synchronization of images from multiple cameras to reconstruct a moving human

In this experiment we used simulation to capture frames from 6 virtual cameras of a synthetic human head rotating at 22.5° per second under two camera desynchronisation schemes. One based on 45° fixed rotation of the head, designed to show the most extreme effects. The second based on realistic delays expected from jitter in post-acquisition processes in the camera system; these were derived from measurements of jitter in the timings of run length encoding of the silhouette and

MP4 compression of texture.

Image acquisition from the cameras in the simulated setting can be regarded as equivalent to a hardware synchronised scheme. The synthetic model is rotated, then images are captured from each of the virtual cameras of the model in a fixed location. The jitter in time taken to compress the silhouette and texture are used to guide expectation in de-synchronisation of images reaching the reconstruction algorithm.

The model was reconstructed using a volumetric technique and the effect desynchronisation had on the resulting spatial quality was presented visually.

Results

The notation (r_{right}° , r_{front}° , r_{top}° , r_{left}° , r_{bottom}° , r_{back}°) or ($t_{right}ms$, $t_{front}ms$, $t_{top}ms$, $t_{left}ms$, $t_{bottom}ms$, $t_{back}ms$) was used where r denotes a rotation in degrees, t denotes a delay in milliseconds and right, front, top, left, bottom and back denote the position of the respective camera arranged around the sides of, and facing into, a cube within which the head was centred.

Figure 4.10 shows the results of using the camera images rotated by 45° for an increasing number of cameras in the set. It can be seen that the effect on the resultant model is dependent on which camera image is rotated. The direction of rotation also affects the deformation of the model. For example, when the top camera is delayed, the model exhibits bunching in the direction of rotation.

Figure 4.11 shows the results of using increasing realistic sub-frame period delays on cameras in the set. To put this into context, the maximum delay used by any of the cameras in this series of reconstructions is 2.5ms, which corresponds to 0.5625° of head turning, given the head was rotating at 22.5° per second. This length was determined as a realistic amount of delay that could be incurred by jitter in run length encoding of the silhouettes used to form the image.

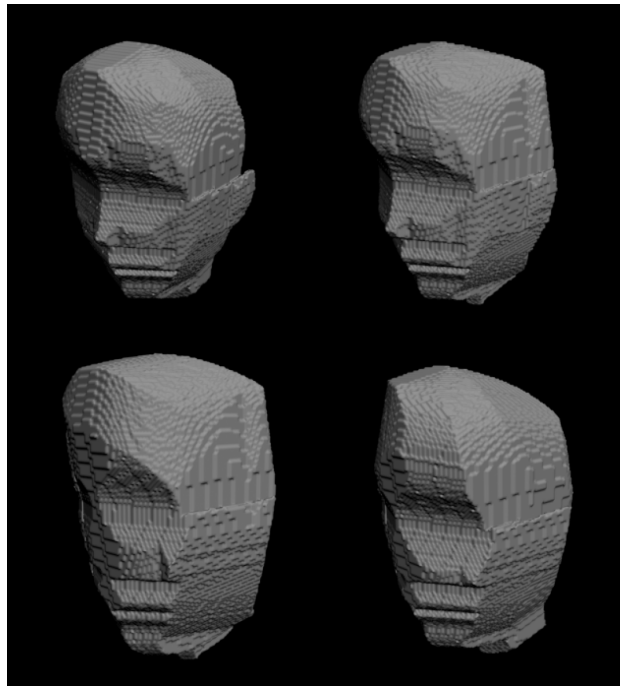


Figure 4.10: Results of the experiment showing the effect of extreme head movement. Clockwise from top left (0_{right}° , 0_{front}° , 0_{top}° , 0_{left}° , 0_{back}°) control, (0_{right}° , 0_{front}° , 0_{top}° , 0_{left}° , 45_{back}°), (0_{right}° , 0_{front}° , 0_{top}° , 45_{left}° , 45_{back}°), (0_{right}° , 0_{front}° , 45_{top}° , 45_{left}° , 45_{back}°).

Full details of the experiment, including timings of the processes of the camera acquisition system not reported here can be found in [82].

Discussion

The experiment concluded that whilst it was clear that camera synchronisation is important, for appreciable deformation to be noticeable cameras must exhibit significant discrepancies in capture time. Given the corresponding timing measurements on the camera acquisition stage, for run length encoding of silhouettes and MP4 texture compression, it was conjectured that such discrepancies would probably not be evident within a typical real-time reconstruction system. The shortcomings of the research were that it:

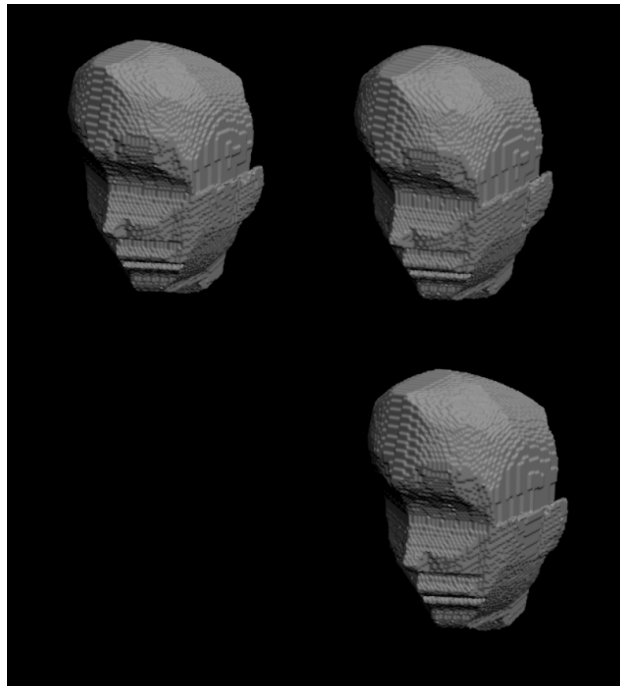


Figure 4.11: Results of the experiment showing reconstructions from synchronised and unsynchronised images. Clockwise from top left (0_{right} ms, 0_{front} ms, 0_{top} ms, 0_{left} ms, 0_{bottom} ms, 0_{back} ms), (0_{right} ms, 0.25_{front} ms, 0.5_{top} ms, 0.75_{left} ms, 1.0_{bottom} ms, 1.25_{back} ms), (0_{right} ms, 0.5_{front} ms, 1.0_{top} ms, 1.5_{left} ms, 2.0_{bottom} ms, 2.5_{back} ms).

- Only studied de-synchronisation artefacts caused by fixed-rate rotation of the human head.
- Only studied effects on spatial quality. The impact on texture and corresponding visual quality remained unknown.
- Only used post-acquisition process timings to estimate jitter in frame arrival at the reconstruction algorithm.
- Was based entirely in simulation.

4.4.2 Experiment E4 - Investigating the suitability of a software capture trigger in a 3D reconstruction system for telepresence

This experiment investigated using a software trigger for frame capture in 3D reconstruction by replacing the traditional hardware synchronisation with a software mechanism that signalled cameras to capture and provide a new frame of data.

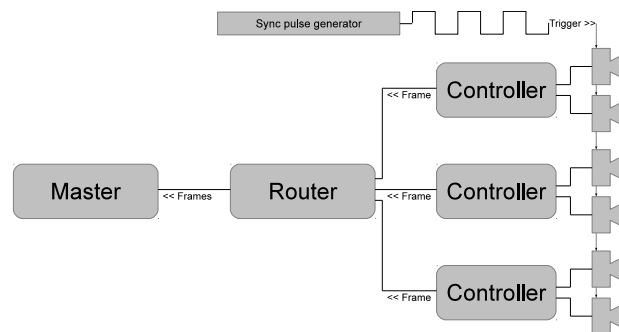


Figure 4.12: Schematic of a hardware triggered "push" camera system.

In a hardware synchronised camera system (Figure 4.12), images are captured by cameras synchronised by a common external trigger pulse. These images are then sent over a connection (usually network) to a number of camera "control computers" in a so called "push" scheme. The word "push" denotes that the cameras are sending image data unsolicited to their control computer, rather than in response to a request for an image. In this manner, images arriving on control computers are images captured by multiple cameras at the exact same moment in time.

In a software triggered camera system (Figure 4.13), the request for a new image from each of the cameras comes from a single (master) computer. This could be the computer on which the 3D reconstruction algorithm is running, or an intermediate computer. The capture request is broadcast to each of the camera control computers, which in turn request images from each of their cameras in a "pull" scheme. Depending on the mechanism by which the master computer is connected to the control computers and cameras are connected to control computers,

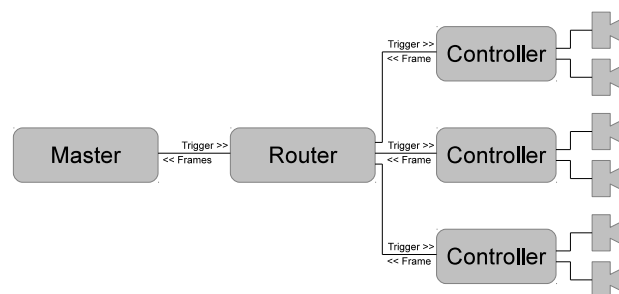


Figure 4.13: Schematic of a software triggered "pull" camera system.

the trigger request has to make at least two connection hops before each camera is triggered. Any jitter in transmission latency, or latency in the processing of the received trigger signal will result in de-synchronised image acquisition at each of the cameras.

Method

In order to measure timings of a software triggered "pull" system, 6 cameras were arranged to capture images of a free running millisecond timer projected onto the floor. The timer was projected by a 102Hz projector, providing approximately 9.8ms granularity in possible readings. Cameras were connected to control computers in pairs, and then via a network router to a master computer issuing the trigger request. Each camera had a maximum frame rate of 48Hz, giving a frame period of 20.83ms.

The master computer broadcast trigger requests over the network to control computers, which on receipt of the request acquired frames from each of the two connected cameras and stored them to disk. The captured stopwatch images were then compared to determine the time offset between frames. Internal timings were also recorded for the total round trip request time experienced by the master computer, and time spent by control computers capturing and processing each frame. By subtracting capture and processing time from the round trip time, it was possible

to calculate the network latency.

Results

48 sets of images were captured of the millisecond timer from the 6 cameras. Of these, 5 sets were too blurred to be readable, and were therefore rejected. 1 set contained a mixture of digits from one frame and the following frame, and was rejected. The remaining 42 frames of data contained images of the millisecond timer showing the same digits. This indicates that the software capture trigger operated within a granularity of around 10ms, based on the update frequency of the projector.

The average time for the software trigger request to make the round trip and provide a full set of frames was 62ms, with an average of 45ms spent by control computers acquiring and processing frames. This results in a network latency of 17ms for sending the request, and receiving the corresponding frames.

Discussion

The experiment attempted to test the suitability of a software trigger for camera image acquisition. However, the results did not achieve sufficient accuracy to determine this suitability. The method used provided an error of 10ms in measurement accuracy, which is only about half a camera frame period. In order to prove that the method was suitable for reconstruction of moving humans, an error considerably smaller than a camera frame period would be required.

In comparing a "push" system using hardware synchronisation to the "pull" system investigated in this experiment, the most useful finding has been the round trip timings. 62ms is longer than a single frame period for the cameras used, and therefore unless a pipeline were implemented whereby a new pull request is issued before the frames from the previous pull request are received it would be impos-

sible to obtain images at the camera's full frame rate. Network latency operates in two directions in the "pull" scheme; both for the request itself and transmission of the result. This combined latency was measured as 17ms, which is approaching the frame period of the cameras used. By contrast, for the hardware synchronised "push" system, network latency affects only the delivery of images. Due to this asymmetry, the "push" system should always be able to achieve the full camera frame-rate regardless of transmission latency.

Full details of the experiment can be found in [83].

4.5 Experiment E5 - Whole frame period temporal inconsistency

In this experiment a whole frame period of temporal inconsistency on camera images supplied to the reconstruction algorithm and the effect this had on spatial and visual quality of the 3D reconstruction was studied.

A full frame period of temporal inconsistency is the maximum temporal offset between frames that one might expect from the "free-running" type of camera image synchronisation scenario listed in Section 4.4. In this scheme, cameras are totally unsynchronised; they run independently of each other, capturing images at their natural frame rate. Assuming cameras are able to provide images consistently at a fixed frame rate, the maximum temporal offset between the start of a frame from one camera and any other camera in the set will be a full frame period.

The same effect might be observed in the case of cameras synchronised by hardware or software trigger, but where delivery of the frames causes significant temporal offset.

4.5.1 Method

Pre-recorded datasets from synchronised cameras were fed to a 3D reconstruction algorithm, and the effect of de-synchronising one or more frames on the visio-spatial quality of the reconstructed model was studied.

Frames were de-synchronised for successive cameras one at a time, so that the images used by the reconstruction algorithm were increasingly temporal inconsistent. This provided a perfectly synchronised set of images to use as a control against which to compare the results of the un-synchronised reconstructions. For example, if four cameras were used during capture of the sequence, we might choose frame 4 as the reference frame, and begin by providing frames [4, 4, 4, 4] to the reconstruction algorithm to produce the perfectly synchronised control model. Following this, to simulate unsynchronised cameras, we could choose to de-synchronise one or more cameras by providing the frame following (or preceding) the reference frame. For example, providing frames [5, 4, 5, 4] would result in a reconstructed model where two of the camera frames were de-synchronised by a single frame. Visual inspection of the reconstructed model can then be used to determine how the unsynchronised cameras have affected the resulting model.

Four datasets were used in the study:

- Martial¹ was from Inria's 4D repository, shot at 1624 x 1224 pixels using 16 cameras at 30 frames per second.
- Juggler was from our own facility, shot at 1004 x 1004 pixels, using 6 cameras at 10 frames per second. For this dataset, cameras were synchronised with a software trigger, adding some inherent de-synchronisation between frame starts, measured at 9ms.
- Nikos² was from the University of Surrey's i3DPost multi-view human ac-

¹<http://4drepository.inrialpes.fr>

²http://kahlan.eps.surrey.ac.uk/i3dpost_action/data

tion dataset. Shot at 1920 x 1080 pixels using 8 cameras at 25 frames per second.

- Dancer¹ was from Inria's 4D repository, shot at 782 x 582 pixels using 8 cameras at 30 frames per second.

For each dataset the correctly synchronised model from the corresponding frames is first reconstructed. Then models are reconstructed with a new camera de-synchronised by a frame each time. When half the number of cameras used by the dataset are unsynchronised, this is considered the maximum single frame de-synchronisation possible for that dataset. De-synchronising further cameras from a set by a whole frame period results in the majority of frames being synchronised to the next frame along in the sequence, and so constitutes a greater degree of synchronisation.

4.5.2 Results

Results are presented in the form of textured reconstructed output models for visual inspection. For each dataset, the correctly synchronised reconstructed model is first presented, along with a schematic showing the camera layout and numbering scheme used. This is followed by the models from the reconstructions with unsynchronised cameras. The unsynchronised models are presented in order of increasing number of de-synchronised cameras. The following notation is used to denote the camera synchronisation in Figures: (1,2,3,4,5,6,7,8) denotes that 8 cameras are used, and none of them are de-synchronised by a frame. (1,2,3,4',5,6,7,8) denotes that 8 cameras are used and camera 4 is de-synchronised by a frame.

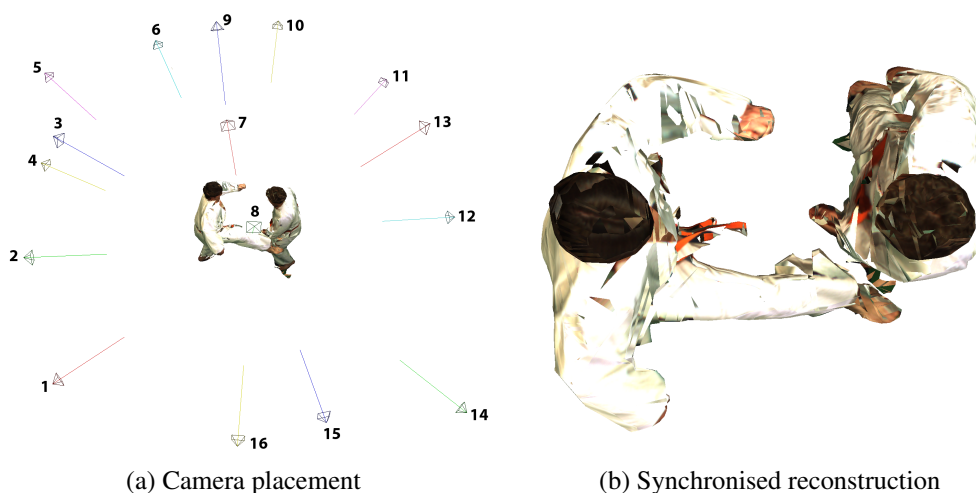


Figure 4.14: Martial, (a) Camera layout used (b) Novel viewpoint reconstructed from synchronised cameras (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16).

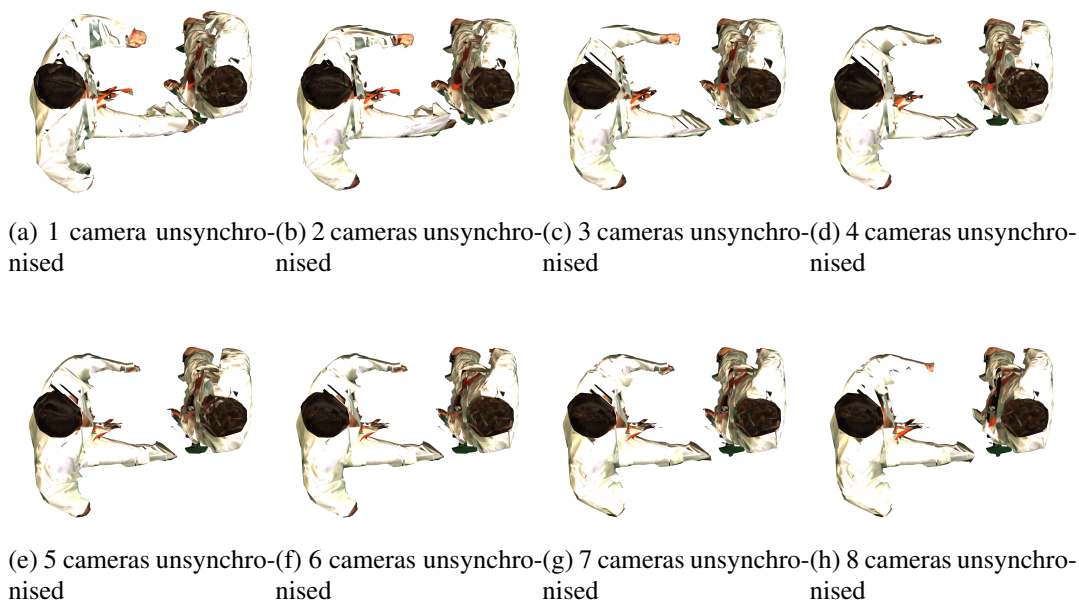


Figure 4.15: Martial, novel viewpoint reconstructed from progressively un-synchronised cameras: (a) (1',2,3,4,5,6,7,8,9,10,11,12,13,14,15,16), (b) (1',2,3',4,5,6,7,8,9,10,11,12,13,14,15,16), (c) (1',2,3',4,5',6,7,8,9,10,11,12,13,14,15,16), (d) (1',2,3',4,5',6,7',8,9,10,11,12,13,14,15,16), (e) (1',2,3',4,5',6,7',8,9',10,11,12,13,14,15,16), (f) (1',2,3',4,5',6,7',8,9',10,11',12,13,14,15,16), (g) (1',2,3',4,5',6,7',8,9',10,11',12,13',14,15,16), (h) (1',2,3',4,5',6,7',8,9',10,11',12,13',14,15',16).

Martial

The martial dataset consists of two men engaged in martial arts. We selected a frame from a section of the overall sequence where the man shown on the left

¹<http://4drepository.inrialpes.fr>

(Man A) is raising his right leg towards the man shown on the right (Man B). The movement of the leg is relatively fast. There is a corresponding balancing action with the left arm, and slight movement of the right arm too. Inspecting the synchronised reconstruction in Figure 4.14b and comparing to the progressively un-synchronised reconstructions in Figure 4.15, one can observe that with only camera 1 unsynchronised (Figure 4.15a), the reconstruction of Man A's right foot is truncated to around the ankle level. This makes sense because, being at the end of the moving right leg, the foot will be the fastest moving part of his anatomy. Also, the foot is directly in front of camera 1, which is the first to be de-synchronised in the sequence. Progressing to Figure 4.15b camera 3 is also unsynchronised. We notice minor changes to the reconstruction of Man A's right leg and foot, but more significant changes to his left arm, which has become thinned. This also makes sense as camera 3 is situated in direct sight of his left arm, but with a poor view of the right leg. The de-synchronisation sequence continues to sweep clockwise around the pair of assailants causing further truncation of Man A's right leg. His left arm becomes thinner and loses the left hand. The tassles coming from his red belt become shortened. Little change can be observed in Man B's reconstruction, and in this part of the sequence he is moving very little. On careful inspection it is possible to observe that his right arm has swung backwards, and his left hand ceases to be reconstructed correctly.

Juggler

The juggler dataset is captured using a small number of cameras, mostly situated to the sides and behind the subject. Only camera 1 faces the subject directly. In Figure 4.16b reconstructed from synchronised cameras it can be seen that there are already errors in the visual quality of the reconstruction. The juggling ball is creating a texturing artefact on the right hand side of the juggler's face. In Figure 4.17a, reconstructed with camera 1 de-synchronised, we see that the juggling ball is still present in the reconstruction, but is causing a new texturing artefact on the juggler's chest. This is because the ball itself has moved in the image of camera 1 to a position lower down, but other cameras are still capturing the ball in its

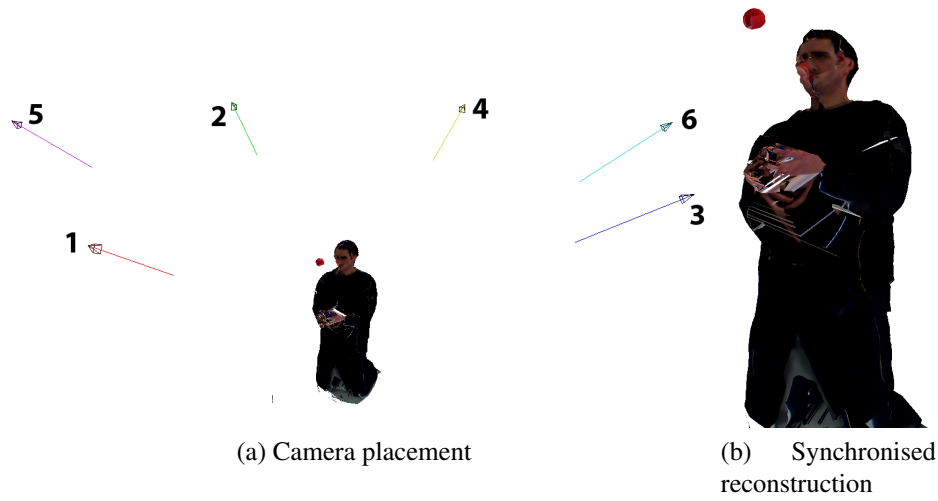


Figure 4.16: Juggler, (a) Camera layout used (b) Novel viewpoint reconstructed from synchronised cameras (1,2,3,4,5,6).

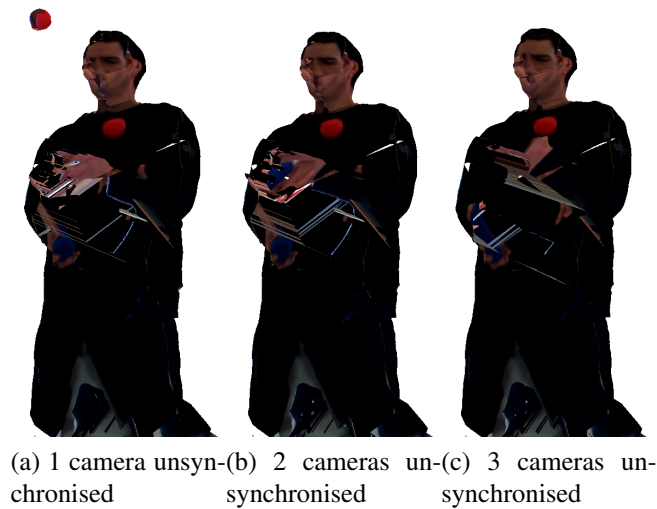


Figure 4.17: Juggler, novel viewpoint reconstructed from progressively un-synchronised cameras. (a) (1',2,3,4,5,6), (b) (1',2,3',4,5,6), (c) (1',2,3',4,5',6).

original position. The texturing artefact comes from the new position of the ball in camera 1, which now projects the ball onto the juggler's chest. We now study the changes in de-synchronised camera silhouettes to better understand the changes in reconstructed form and texturing that have taken place.

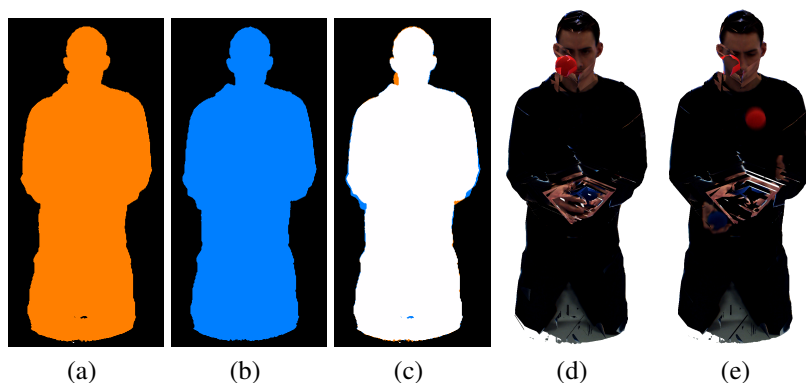


Figure 4.18: Juggler (1',2,3,4,5,6) from camera 1 viewpoint (a) Synchronised camera 1 silhouette. (b) Unsynchronised camera 1 silhouette. (c) Difference between silhouettes. (d) Synchronised reconstruction. (e) Unsynchronised reconstruction.

In Figure 4.18c very little has changed in terms of the silhouettes from which the form is reconstructed. The juggler's left and right elbows have moved very slightly, and there are slight corresponding changes in that region of the reconstruction when comparing the synchronised reconstruction of Figure 4.18d to the unsynchronised reconstruction in Figure 4.18e. A very small spot is present between the knees in the undelayed silhouette that is not present in the unsynchronised silhouette. This is likely to be a background segmentation error. Indeed, looking closely between the knees of the reconstructed juggler, it is evident that this whole region is not part of the juggler's anatomy and has erroneously been included in the silhouettes in the process of background segmentation. The likely reason for this is because the area lies in shadow. The other difference between the silhouettes is that the edge of the juggling ball is present in the synchronised silhouette, but not in the unsynchronised one. In the reconstruction we can see this more clearly. We can also see that the reconstructed ball is still present in the unsynchronised reconstruction, but it appears that part of it is missing. In fact the geometry of the ball is still intact, but the texture applied to the region of it facing camera 1 has been selected from the unsynchronised camera 1 image which now contains the juggler's face where the ball is. Looking back at Figure 4.17a this can be seen from the novel viewpoint at the top left of the ball.

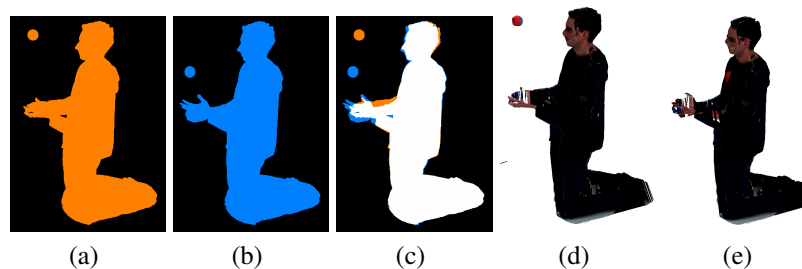


Figure 4.19: Juggler (1',2,3',4,5,6) from camera 3 viewpoint (a) Synchronised camera 3 silhouette. (b) Unsynchronised camera 3 silhouette. (c) Difference between silhouettes. (d) Synchronised reconstruction. (e) Unsynchronised reconstruction.

Figure 4.19 shows how camera 3 sees the movement of the juggling ball between the two frames. The ball has dropped significantly, so much so that it no longer appears in the reconstructed form in Figure 4.19e. It does however cause a texturing artefact as was already observed in Figure 4.17. The juggler's face has suffered from a slight tilting of the head forwards, which causes the nose and chin to become shortened. Whilst it can be seen from the difference silhouette that the hands have moved a little, it is not evident from the reconstruction exactly how the geometry is now formed. The area is confused by texturing inconsistencies.

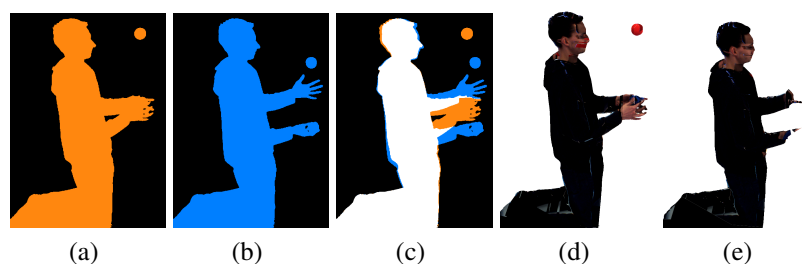


Figure 4.20: Juggler (1',2,3',4,5',6) from camera 5 viewpoint (a) Synchronised camera 5 silhouette. (b) Unsynchronised camera 5 silhouette. (c) Difference between silhouettes. (d) Synchronised reconstruction. (e) Unsynchronised reconstruction.

From the view of camera 5, it can be seen how the juggler's arms have moved between the two frames. In Figure 4.20a they are close together, and in the unsynchronised frame they are apart. This significant difference between the arms and

hands causes the juggler to lose both hands and much of his right forearm in the unsynchronised reconstruction (Figure 4.20e).

Nikos

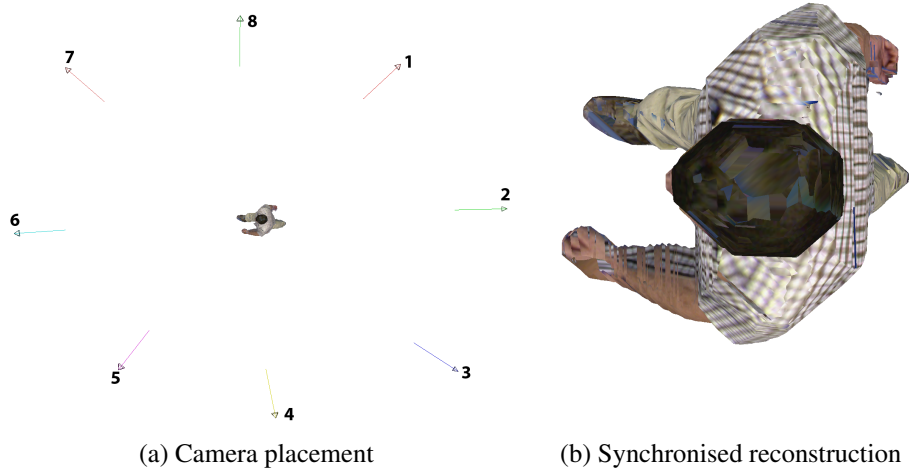


Figure 4.21: Nikos, (a) Camera layout used (b) Novel viewpoint reconstructed from synchronised cameras (1,2,3,4,5,6,7,8).

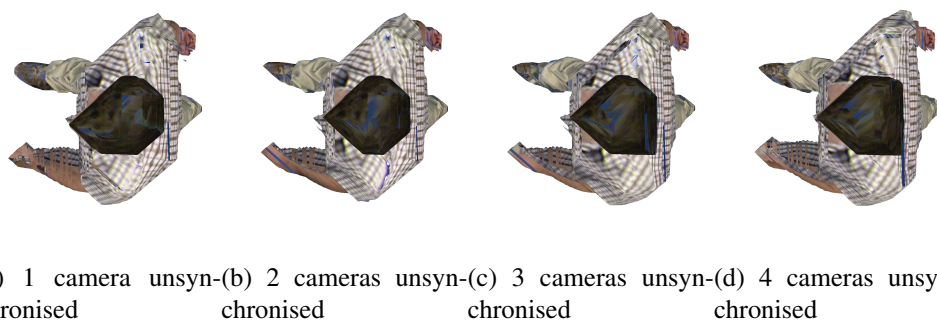


Figure 4.22: Nikos, novel viewpoint reconstructed from progressively unsynchronised cameras. (a) (1',2,3,4,5,6,7,8), (b) (1',2,3',4,5,6,7,8), (c) (1',2,3',4,5',6,7,8), (d) (1',2,3',4,5',6,7',8).

In this dataset, Nikos is walking forward at a steady pace. Figure 4.21a shows all

cameras are placed around the edge of the reconstruction area at regular angles pointing towards the centre. We selected a frame from close to the middle of the sequence so that he is of a similar size in all camera images. The reference synchronised reconstruction 4.21b is from above, where there are no cameras. A good result is achieved apart from some texturing artefacts on his left arm, which come from camera 1. In Figure 4.22 we see the results of de-synchronising cameras. Initially de-synchronising camera 1 causes the right side of his head to become flattened, this makes sense as he is moving forward, and therefore the image coming from camera 1 places him further ahead than the other cameras. The same can be seen in Figure 4.22b when camera 3 is de-synchronised, causing the left side of his head to become flattened.

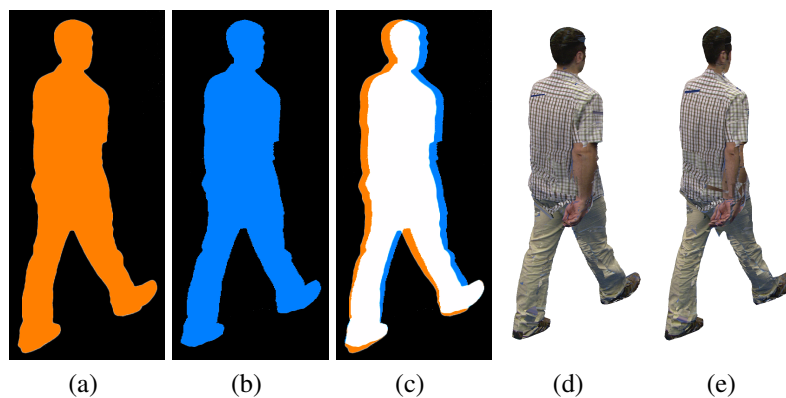


Figure 4.23: Nikos (1',2,3,4,5,6,7,8) from camera 1 viewpoint (a) Synchronised camera 1 silhouette. (b) Unsynchronised camera 1 silhouette. (c) Difference between silhouettes. (d) Synchronised reconstruction. (e) Unsynchronised reconstruction.

In Figure 4.23 it can be seen from the difference silhouette that Nikos is moving forward. His whole body advances, with only the back of his left leg remaining stationary from this angle. This results in a general thinning of the body, including his head which has become quite narrow in the reconstruction (Figure 4.23e).

The view from camera 3 (Figure 4.24) features both arms in the silhouette. The swinging movement of the arms as Nikos walks forward causes a significant thinning of them in the reconstruction (Figure 4.24e), his left hand is reduced to a

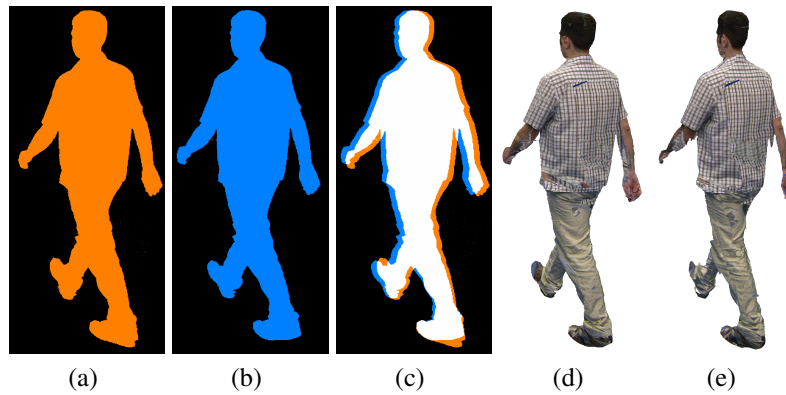


Figure 4.24: Nikos (1',2,3',4,5,6,7,8) from camera 3 viewpoint (a) Undelayed camera 3 silhouette. (b) Unsynchronised camera 3 silhouette. (c) Difference between silhouettes. (d) Synchronised reconstruction. (e) Unsynchronised reconstruction.

stub. His right foot has become pointed and this leg has also suffered considerable thinning.

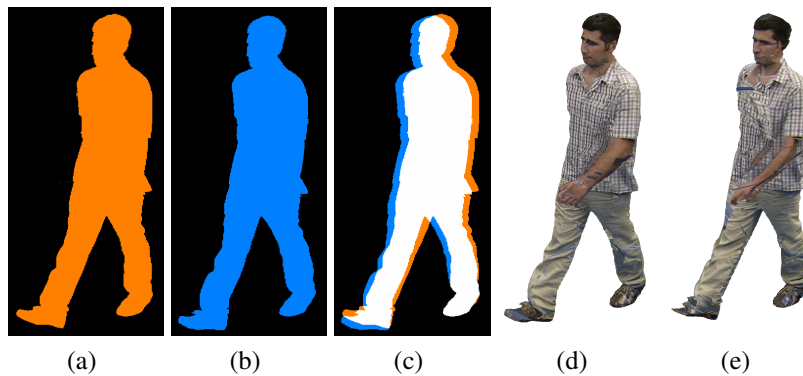


Figure 4.25: Nikos (1',2,3',4,5',6,7,8) from camera 5 viewpoint (a) Synchronised camera 5 silhouette. (b) Unsynchronised camera 5 silhouette. (c) Difference between silhouettes. (d) Synchronised reconstruction. (e) Unsynchronised reconstruction.

Camera 5 viewpoint is similar to the view from camera 1, whilst Nikos' arms are visible, neither form a feature in the silhouette, and therefore the degradation of his left arm visible in the reconstruction resulted from the de-synchronisation of camera 3. There is more general thinning of the upper body, particularly the head

and shoulders (Figure 4.25).

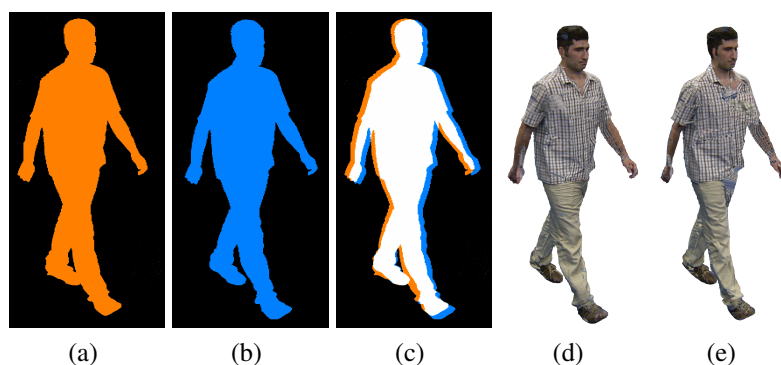


Figure 4.26: Nikos (1',2,3',4,5',6,7',8) from camera 7 viewpoint (a) Synchronised camera 7 silhouette. (b) Unsynchronised camera 7 silhouette. (c) Difference between silhouettes. (d) Synchronised reconstruction. (e) Unsynchronised reconstruction.

Figure 4.26 shows the view from camera 7. This is perhaps the least remarkable of the delayed reconstructions, particularly considering the reconstructed geometry is the same as in Figure 4.22d. Whilst the head in this figure has clearly been thinned, from this camera's point of view it is not obvious that it has suffered such severe spatial quality degradation.

Dancer

The dancer dataset is a sequence of a woman dancing captured by 8 fairly regularly arranged cameras (Figure 4.27a). Since the cameras used to film the sequence are fairly low resolution, and the dancer does not fill the frame in any of the camera images, the visual quality of even the synchronised reconstruction (Figure 4.27b) is relatively poor. For example, the facial features cannot be clearly seen. Unsynchronised reconstructions from the novel viewpoint shown in Figure 4.28 are characterised by a thinning of the arms, leading to a loss of the lower right arm. Slight thinning of the left lower leg can also be seen.

Figures 4.29, 4.30, 4.31, and 4.32 all show that the dancer's arms are subject to

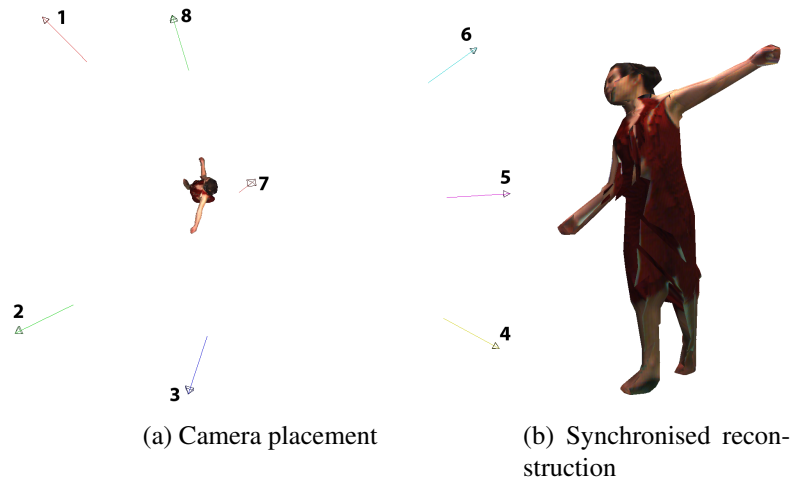


Figure 4.27: Dancer, (a) Camera layout (b) Novel viewpoint reconstructed from synchronised cameras (1,2,3,4,5,6,7,8).



(a) 1 camera unsyn- (b) 2 cameras unsyn- (c) 3 cameras unsyn- (d) 4 cameras unsyn-
 nised chronised chronised chronised

Figure 4.28: Dancer, novel viewpoint reconstructed from progressively un-synchronised cameras. (a) (1',2,3,4,5,6,7,8), (b) (1',2',3,4,5,6,7,8), (c) (1',2',3',4,5,6,7,8), (d) (1',2',3',4',5,6,7,8)

more movement than any other part of her body during this frame transition.



Figure 4.29: Dancer (1',2,3,4,5,6,7,8) from camera 1 viewpoint (a) Synchronised camera 1 silhouette. (b) Unsynchronised camera 1 silhouette. (c) Difference between silhouettes. (d) Synchronised reconstruction. (e) Unsynchronised reconstruction.



Figure 4.30: Dancer (1',2',3,4,5,6,7,8) from camera 2 viewpoint (a) Synchronised camera 2 silhouette. (b) Unsynchronised camera 2 silhouette. (c) Difference between silhouettes. (d) Synchronised reconstruction. (e) Unsynchronised reconstruction.

4.5.3 Discussion

By using whole frame periods to study the effect using non synchronised camera frames has on the visio-spatial quality of 3D reconstruction, we have shown that, for a variety of different types of human movement, important features can be lost within a single frame. Progressive degradation of the reconstructed form was observed as increasing numbers of cameras were de-synchronised by a frame, as might be expected. Up to half of the cameras were de-synchronised for each of the studied datasets. The rationale behind this was that when using whole frame periods of de-synchronisation, the maximum de-synchronisation of the camera set

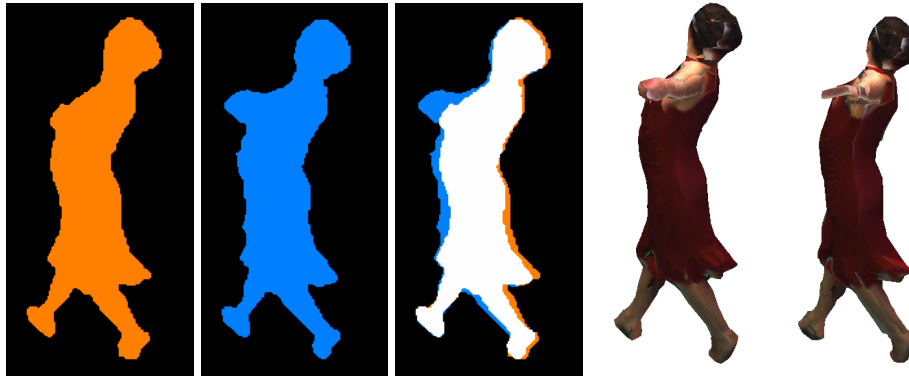


Figure 4.31: Dancer (1',2',3',4,5,6,7,8) from camera 3 viewpoint (a) Synchronised camera 3 silhouette. (b) Unsynchronised camera 3 silhouette. (c) Difference between silhouettes. (d) Synchronised reconstruction. (e) Unsynchronised reconstruction.



Figure 4.32: Dancer (1',2',3',4',5,6,7,8) from camera 4 viewpoint (a) Synchronised camera 4 silhouette. (b) Unsynchronised camera 4 silhouette. (c) Difference between silhouettes. (d) Synchronised reconstruction. (e) Unsynchronised reconstruction.

as a whole is achieved when half of the cameras are de-synchronised.

This does not mean that the maximum possible reconstruction error would be observed when half of the cameras are unsynchronised, however. As has been shown, reconstruction errors arise from cameras observing movement between the frame that would be synchronised with other cameras in the set, and the unsynchronised frame. This movement can give rise to two different types of reconstruction quality degradation: spatial and visual. Spatial quality degradation occurs when the movement between frames corresponds to the bounding edge

of the object, and therefore causes the silhouette formed to differ from the correctly synchronised frame. Visual quality is degraded both as a result of spatial errors in the reconstruction and differences in the camera image itself. It is quite conceivable for an unsynchronised camera frame to exhibit no differences in the silhouette formed, and therefore the resulting spatial quality, but for movement to be observable only within the texture enclosed by the silhouette. Figure 4.18 comes close to demonstrating this principle. In such cases, the spatial quality of the reconstruction does not suffer as a result of loss of synchronisation, but the visual quality is degraded through errors in texturing.

In terms of a free-running camera acquisition system used for 3D reconstruction, the experiment simulates the worst case scenario for camera de-synchronisation; where an entire frame period of de-synchronisation is present. It was convenient to conduct the experiment in this manner as it could be achieved with frames captured using a synchronised camera set, but the approach taken is believed to give indicative results for a free-running camera acquisition system. In a dataset captured from a true free-running camera system, one would expect to see none of the cameras perfectly synchronised, and none of the cameras de-synchronised by a whole frame period.

In terms of a synchronised camera acquisition system, an entire frame period of delay would constitute a significant loss of synchronisation. However, various researchers have reported delays of that order in the literature [78], [119], resulting from network transmission.

It is evident from this study that the combination of speed of movement and camera frame rate will determine the quality degradation observed from using unsynchronised camera images. For telepresence applications it may be tempting to assume that humans will not be moving at considerable speed. However, it has been shown that important features like the hands are most likely to suffer as these are situated at the end of limbs, and even seated meeting participants can use considerable arm waving gestures to get their point across.

4.6 Conclusion

This chapter has developed an understanding of three quality measures that can be used for 3D reconstruction. In the context of a shape-from-silhouette based reconstruction system, these output qualities have been defined in terms of the camera image inputs and temporal performance of the reconstruction algorithm.

4.6.1 Quality measures and their relationship

Spatial quality is determined by the number of cameras, their positioning and resolution. The more cameras used, the better the form constraint achieved by the visual hull. The greater the directional diversity of these cameras on the object under reconstruction, the better the form constraint of the visual hull. As camera image resolution increases, silhouettes better define the contour of the object under reconstruction, leading to a higher spatial quality.

Visual quality is determined by spatial quality and camera resolution. Poor spatial quality will not result in an accurate representation of the object under reconstruction, which alone results in poor visual faithfulness. Application of surface textures to an inaccurate form can result in distortion, further reducing visual quality. Higher resolution cameras allow for more detail to be captured, thereby creating the opportunity for higher visual quality.

Temporal quality is determined by the speed and latency of the 3D reconstruction algorithm. Increasing speed and decreasing latency result in an increase in temporal quality. In the case of the EPVH algorithm, temporal quality is determined by the number and complexity of silhouette inputs. Simple silhouettes defined by a small number of straight line segments from a small number of cameras will be reconstructed more quickly than a greater number of more complex silhouettes.

In general, therefore, as spatial and visual quality increase with the number of cameras and their resolution, temporal quality decreases. Hence, putting this into

the context of a real-time 3D reconstruction system, it follows that for a given temporal quality reconstructions from a certain input complexity can be achieved.

4.6.2 Impact of 3D reconstruction system components on quality

The choice of camera lens can impact upon either spatial and visual quality or temporal quality. Wide angle lenses lead to image distortion, which left uncorrected will result in spatial and visual quality of the reconstructed output suffering. Correction of the image distortion takes processing time, and will therefore add to the latency and consequently the temporal quality of a 3D reconstruction system.

Camera trigger and distribution method can both impact upon spatial, visual and temporal quality in a 3D reconstruction system. Using a hardware trigger will ensure that images from cameras are perfectly synchronised, resulting in temporally consistent inputs. A software trigger system cannot guarantee such temporal consistency on inputs, and can lead to spatial and visual quality degradation. Camera distribution method can also lead to temporal inconsistency on inputs in the case where frame transmission is delayed, or the order is not guaranteed by the transmission protocol.

Camera placement impacts on the spatial quality of the visual hull, and moving objects can make careful camera placement a pointless exercise. Considering contributions to the visual hull in terms of their visibility within a subset of cameras can greatly alleviate the impact of camera placement on spatial quality. Objects partially visible in a particular camera will no longer be clipped at the camera edge; objects invisible from a particular camera can still be reconstructed by the cameras from which the objects are visible.

Camera calibration accuracy determines the spatial accuracy of the system, and in turn impacts upon spatial and visual quality. Cameras with high calibration error will result in inaccurate projection of the silhouette contour and texture pixels

from the camera image plane into three dimensions. This leads to errors in both form constraint and texturing.

Background segmentation is critical for high spatial quality in shape-from-silhouette based 3D reconstruction. Poor segmentation from a variety of common causes can result in reconstruction errors in the form of erroneous objects, holes through the reconstructed form, eroded or expanded edge of objects. Lighting and shadows are particularly difficult to control in terms of their effect on background segmentation.

Camera exposure and colour balance impact upon visual quality. The camera set needs to be carefully adjusted for even colour and exposure balancing across the set. Poor balancing leads to stripy texturing of the reconstruction, adversely affecting visual quality. Real-time correction of exposure and colour balance takes processing time and can therefore lead to temporal quality degradation.

4.6.3 Temporal consistency of inputs

Use of unsynchronised cameras in a 3D reconstruction system is an attractive option because it allows for a much wider choice of cameras. Those featuring a hardware synchronisation input are few and far between, and tend to exist only at the high end of the market. The lack of hardware trigger leaves two options at the camera acquisition stage: use of a software trigger, use of free running cameras. A software trigger takes the form of a signal originating from a single computer requesting that a new image be acquired from all cameras. This signal is likely to be received or processed by individual cameras at slightly different times, and hence gives rise to a temporal inconsistency between frames from each camera. Free running cameras are completely unsynchronised, and therefore the temporal offset between frame starts for each camera can be anywhere up to a whole frame period apart.

Through a number of experiments it has been shown that temporal consistency of

inputs is important for 3D reconstruction of moving humans. In the case of a free running system, or a system in which the distribution method can result in frames arriving a whole frame period late, progressive degradation in spatial and visual quality of moving humans was demonstrated as more cameras became unsynchronised. A software triggered system was shown to exhibit relatively small temporal frame offsets compared to those possible from a free running system. However, it is clear that spatial quality degrades proportionally to the speed of movement. Therefore in the context of a 3D telepresence system, frame offsets would need to be guaranteed to be small enough for human movement within that time period to be minuscule in terms of their pixel representation.

Chapter 5

Improving performance of VBR

This chapter discusses temporal quality of 3D reconstruction, specifically the performance of the algorithm and approaches to improving it. In the previous chapter the necessity to balance visual, spatial and temporal qualities was discussed, and it was determined that increasing camera count or resolution could increase spatial and visual quality, but at the cost of temporal quality. Consequently, by reversing the logic of this statement - Improving the performance of the 3D reconstruction algorithm will enable higher camera counts or resolutions to be used, leading to higher spatial, visual and temporal qualities in real-time. The focus of this chapter is therefore on improving the performance of the 3D reconstruction algorithm. A parallelisation strategy for the EPVH algorithm that is adapted for execution on modern multi-core computation hardware such as GPUs and CPUs, which could remove the requirement for network distributed processing to achieve interactive frame rates. During the course of this parallelisation a number of optimisations that further increase the algorithm's performance are presented. The performance of our implementation is compared to the published performance of EPVH, and also between execution on a GPU and CPU.

The work presented in this chapter has been published in [\[34\]](#).

5.1 Objectives and research questions

O2: Determine whether algorithms can be improved upon to achieve higher temporal or spatial/visual quality.

Q2: How can algorithms or their implementation be improved upon to achieve higher qualities in real-time?

The temporal quality of the EPVH algorithm is determined by the complexity of the inputs, which is generally defined by the number and resolution of cameras. Therefore, it follows that improving the performance of the algorithm by some means would allow higher spatial and visual qualities to be achieved per unit time.

5.2 Introduction

Using a state of the art algorithm to reconstruct this form, researchers have previously shown that network distributed processing can be used to increase the performance of the algorithm enough to create visually faithful model humans at interactive frame rates. We propose a parallelisation strategy for the algorithm adapted for execution on multiple core general purpose computational hardware, such as CPUs and GPUs. This could remove the requirement for network distributed processing in real-time 3D reconstruction systems, which would eliminate communication and synchronisation overheads that lead to increased latency. Instead processing is performed in parallel on the cores of local compute resources, this both simplifies system design, and enables a finer grained, more targeted approach to the partitioning of work, resulting in better parallel workload balancing, and ultimately higher performance. The main contributions of this research are:

- An updated parallel partitioning strategy for the EPVH algorithm, optimised for local processing, including details of a number of optimisations.

- An alternative structure that accelerates the search for image intersections and does not suffer from performance degradation when scaled to high camera counts.
- A comparison of the performance of the algorithm under distributed and local parallel processing.
- An analysis of the performance of the algorithm on various multi-core processors.

5.2.1 Context

Recent advances in computing technology have seen the previous trend for ever increasing processor clock speeds reach a ceiling, due to limits of the materials used to build them. Instead, increasing numbers of processing cores are added to a single processing chip. Since these cores all access the same shared memory, the system can be programmed to process the data cooperatively on multiple cores in less time than a traditional sequential program. Further to this, the advent of general purpose GPU computing (GPGPU) makes it possible to shift number crunching tasks from the main processor to the GPU. Since GPUs are typically comprised of many more, simpler cores than a CPU, in some cases GPGPU computing can yield significant performance increases over processing on the CPU. Emerging general purpose languages, such as OpenCL, provide a means by which various types of multi-core processor can be employed, in theory without any code changes, and hence provides a means of targeting diverse hardware, although performance portability does not yet happen automatically. The future of processing technology appears to be continuing along the multi-core route: GPUs and CPUs will have increasing numbers of cores, and heterogeneous architectures, already seen in the likes of AMD Fusion and IBM Cell BE, are likely to be employed in the future in massively multi-core processors, possibly featuring non uniform memory architectures.

5.3 Related Work

Exact Polyhedral Visual Hulls (EPVH) [39] is a state of the art polyhedral reconstruction algorithm that creates a guaranteed watertight, manifold polyhedral representation of the visual hull. A polyhedral visual hull may have a texture applied to it derived from the camera images, special consideration needs to be made for the mapping of textures when there are several candidate cameras [66] and [97] describe the mapping of textures from multiple cameras onto such forms. Image Based Visual Hulls (IBVH) [79] is a reconstruction approach based on similar principles to EPVH, but renders a specific viewpoint rather than outputting a model. It is clear from both that the computational complexity of such approaches is largely in searching for intersections with camera images. To this end the authors of both approaches suggest optimisations that may be employed to reduce the overhead of such searches, in both cases requiring the building of lookup tables. Alternatively, spatial data searches such as the search for intersections can be accelerated by making use of a tree like structure, such as the R-Tree ([47]). The R-Tree is a structure that can be used to accelerate querying many types of multi-dimensional data, and numerous adaptations exist to meet specific requirements. One such adaptation, the R+-Tree [108] can increase performance by reducing overlapping areas, and is also quick to initialise from a static dataset. In [40], the EPVH authors determine that their method for creating the polyhedron still demonstrates silhouette consistency even when contours are modelled at camera image pixel precision, rather than at the sub-pixel precisions previously employed - This important finding significantly reduces the computational cost of the algorithm for real-time applications. [41] describes a system in which the EPVH algorithm is processed in parallel by a number of networked computers. It is found that a real-time visual hull may be generated from a number of cameras when processed in this way. In this scheme, the work is broken down into tasks that are executed by a cluster of networked computers. A three stage pipeline is employed along which a stream of frames progresses; each processing node can be assigned a pipeline stage resulting in stream level parallelism, whilst the partitioning of work within each stage of the pipeline can be further divided into units of work

that can be executed in parallel across a single frame. Tasks that are not inherently parallel are executed sequentially in between pipeline stages. The downside of this scheme is that distributed processing will inevitably introduce delays compared with a shared memory system. Data must be communicated between the nodes, and this can lead to synchronisation bottlenecks where parallel threads are stalled waiting for data, which can result in poor load balancing. Furthermore any code that must be executed sequentially, or network data that is passed sequentially could hinder the flow of parallel execution if it takes significant time to complete [4]. Whilst this might initially seem like a trivial consideration, future massively multicore processors could spread the load over many more processing units, resulting in greater impact of sequential code segments [51]. The distributed processing scheme for EPVH introduced in [41] is deployed in an end to end telepresence system [1] and [93]. With the recent advent of GPGPU computing there are numerous emerging languages and standards that can be used to target a variety of compute resources. Some languages, such as CUDA are vendor specific, and therefore useful only on specific hardware. OpenCL is a language that aims to provide a portable way to exploit the parallel nature of a number of fundamentally different types of multi-core processors from a variety of manufacturers. Identical code can be run on CPUs, GPUs or a combination of the two, making it highly portable. However, OpenCL does not yet provide automatic performance portability, so code may need to be adapted following device profiling [118] to achieve the best results. For multi-core CPUs various parallelisation libraries such as OpenMP provide a means by which programmers can easily schedule work to be executed in parallel on a number of cores. [24] provide a comparison of OpenMP and OpenCL in terms of their relative performance, and discuss the advantages and disadvantages of each: OpenCL requires greater porting effort for existing algorithms, but results in an implementation that can be executed on a range of hardware. OpenMP requires manual configuration to get the most out of the various extensions offered by a processor, whereas OpenCL achieves this automatically. We previously published work Duckworth and Roberts [32] in which we accelerated certain parts of the EPVH algorithm using the GPU. We found that for increased camera counts and image resolutions, the GPU offered accelerated performance compared with the CPU variant. However, there were a number of

shortcomings to this work: Testing used synthetic data rather than real camera images, only part of the algorithm was parallelised, the study only compared GPU performance against the sequential (single core) CPU counterpart.

5.4 Motivation and Scope of work

Modern multi-core processors could simplify 3D reconstruction system design, and improve efficiency by avoiding distributed processing overheads. Emerging general purpose computation languages enable the targeting of various processor types from the same source code, but currently performance can vary considerably. The scope of this research is:

- Adapt the distributed EPVH algorithm for local multi-core processing, An updated parallel partitioning of the EPVH algorithm for local multi-core processing is described.
- Compare the performance of the algorithm running locally to published measurements from the distributed implementation. The speed-up of the multi-core implementation over a range of processing cores and camera counts is measured and compared to the literature.
- Characterise the performance behaviour when running on different types of multi-core processors. Results are presented for the execution times on two CPUs and one GPU.
- Evaluate the performance of reconstructing real humans. Several examples of human reconstructions are presented along with timings.

5.5 EPVH Algorithm and primitives

A short summary of the EPVH algorithm and associated primitives is provided to introduce the necessary terminology and context to assist in the description of the parallelisation approach.

5.5.1 Camera Images and 2D primitives

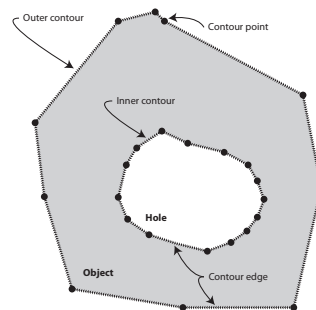


Figure 5.1: 2D Primitives: Two contours, comprising a number of contour edges each made from two contour points. Outer contours define the edge of the object being reconstructed, inner contours define holes in the object - They are differentiated by anti-clockwise or clockwise orientation respectively

Camera images are segmented to remove the background, resulting in silhouette images. The edge (contour) of the silhouette image (Figure 5.1) can be described as the points forming contour edges that progress around the contour creating a closed two dimensional polygon. The direction of progression around the contour, clockwise or anti-clockwise denotes whether the contour is an inner or outer contour edge respectively. Outer edges enclose scene objects in the camera image plane, whereas inner edges form holes within objects.

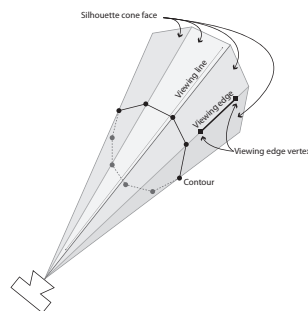


Figure 5.2: 3D Primitives: The silhouette cone - Contour points are projected into 3D space from a camera, forming a number of viewing lines and silhouette cone faces. Viewing edge vertices are situated along a viewing line, 2 viewing edge vertices form a viewing edge that will be part of the final model.

5.5.2 Silhouette cones and 3D primitives

Contours derived from camera images are projected into 3D space forming silhouette cones. Each projected contour point forms a viewing line emanating from the camera's optical centre and continuing indefinitely. A viewing line and its adjacent neighbour form an infinite triangular silhouette cone face. Silhouette cones from every camera contour are intersected in 3D space to determine areas of the silhouette cone faces that make up the surface of the visual hull. This is achieved by intersecting spans along viewing lines that fall within the silhouette cones from other cameras - when the contribution from all cameras has been taken into account, the resulting span forms two viewing edge vertices that together define a viewing edge. The viewing edge vertices and viewing edge will form part of the final model. The remaining components of the final model are triple points that are situated at the locus of three silhouette cones, and edges that connect together viewing edges vertices, and triple points along the line of intersection between two silhouette cone faces. Triple points are vertices formed at the intersection of three silhouette cone faces, they are determined by following the intersection of two silhouette cone faces across the surface of the visual hull until either the edge of one of the silhouette cone faces is reached, or a new silhouette cone faces intersects. In the former case, an edge is created that connects together the two viewing edge vertices, in the latter case, a triple point vertex is created and connected to the

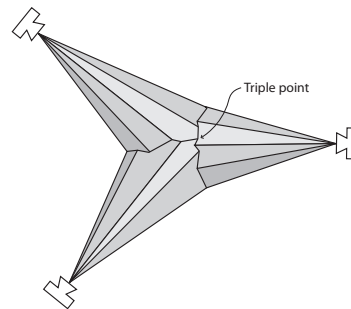


Figure 5.3: 3D Primitives: The triple point is a vertex formed at the locus of three silhouette cones, and is defined by the intersection of the three silhouette cone faces.

viewing edge vertex along the intersection line. All vertices in the final mesh must have three edges connecting them to other vertices which are either of the triple point, or viewing edge types - once all these edges have been found, the mesh is complete. Visual hull surface polygons are recovered by traversing a given edge and taking left or right turns at each vertex encountered to ensure the silhouette cone face on the left or right hand side of the edge being traversed remains consistent from one edge to the next - Once each edge has been traversed once in each direction all surface polygons have been recovered. Creation of the polyhedral model is complete, the surface polygons may now be textured if desired

5.6 Implementation

Extending upon previously published work Duckworth and Roberts [32] in which some parts of the EPVH algorithm were accelerated using the GPU, an end to end parallel partitioning is presented.

5.6.1 Methodology

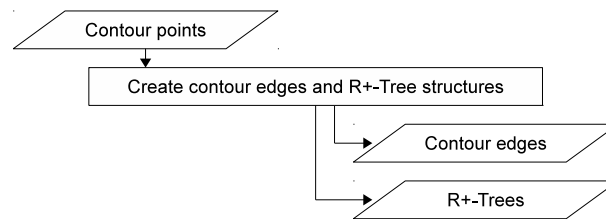
Due to the lack of a reference EPVH implementation, the algorithm has been reimplemented from the descriptions in [39] and [40]. Two such implementations have been developed: A C++ implementation that can be compiled to run on any standard computer, and an implementation in the OpenCL language that can be run on computers with any OpenCL capable device (CPUs, GPUs or a combination of both). The C++ implementation can be executed sequentially on a single core of the CPU, or can make use of multiple cores through OpenMP. OpenCL kernels can be scheduled to execute on either the GPU or the CPU.

5.6.2 Inputs, outputs and data preparation

Inputs and Outputs

There are two types of input to the algorithm, constants and variables. We choose to regard the cameras as fixed during the course of reconstruction, so camera parameters are constants in our implementation. Only one variable is required to reconstruct the polyhedral model, the lists of contour points defining the outlines of silhouettes in the camera images. Should a textured model be required, normalised texture coordinates can be generated from the camera parameters, without the actual texture.

The algorithm outputs the 3D mesh that defines the surfaces of objects being modelled, and, optionally texture coordinates. The mesh comprises a list of 3D vertices and lists of joining edges defining each surface polygon. Texture coordinates define the camera, and image coordinates that each polygon vertex maps to.



Contour processing

Two structures are created for every contour passed to the algorithm, the contour edges making up the contour, and the R+-Tree intersection data structure. Contour edges are chosen as the input for the subsequent steps of implementation because they contain not only the contour points from which viewing lines will be generated, but also the contour edges required for the search for intersections with viewing lines projected from other contours. During processing, the orientation of the contour, clockwise or anti-clockwise is determined - this, and other information such as the originating camera and contour index are stored in the contour edges data structure. An R+-Tree structure is also initialised for each contour to accelerate searches for line intersections with it. Further details of this structure can be found in Section 5.6.3

5.6.3 Parallelisation

Figure 5.4 shows the parallelisation of the EPVH algorithm, the dotted lines represent synchronisation points where the next stage of the algorithm depends upon the previous stage being completed. Such dependencies arise where algorithms cannot be regarded as a single data independent thread from start to end. These dependencies create natural synchronisation points, since the execution of the following stage depends upon all of the data from the previous stage having been processed. A subset of these synchronisation points form the basis for the stream level parallelisation adopted in Franco et al. [41]. The parallelisation strategy presented here is frame based, the synchronisation points are used to change the data entity over which the processing is partitioned, leading to an adaptive distribution

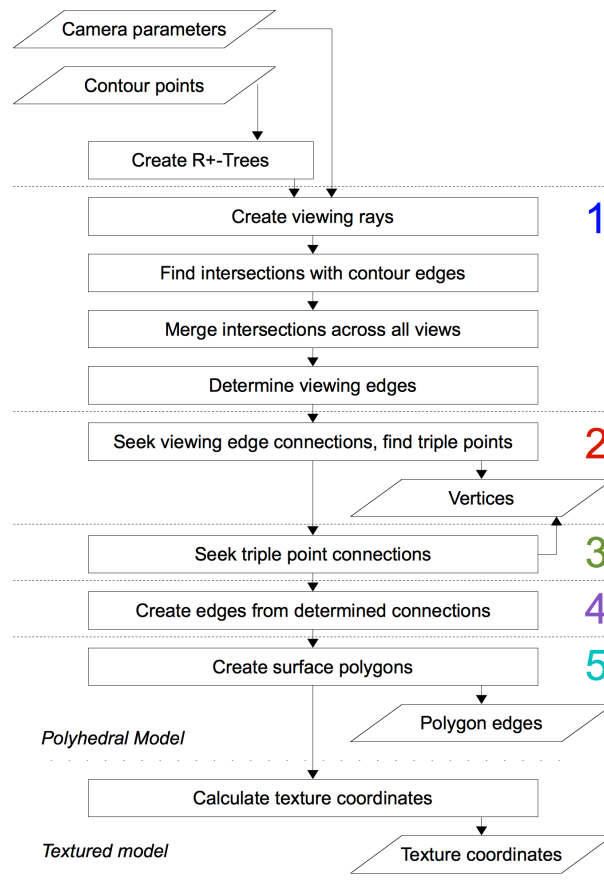


Figure 5.4: Algorithm flowchart defining inputs, outputs and processes in reconstructing the visual hull using the EPVH algorithm. Regions labelled 1 - 5 are data independent parallel regions

of processing as the algorithm progresses. By contrast, choosing not to repartition the work at these synchronisation points could lead to a poorly balanced workload among parallel threads, resulting in reduced performance.

Parallelisation differences compared to distributed approach

Figure 5.5 shows the difference between the parallelisation scheme employed in the distributed approach Franco et al. [41] and the one proposed in this research. For the first step, the distributed approach makes use of the computers hosting

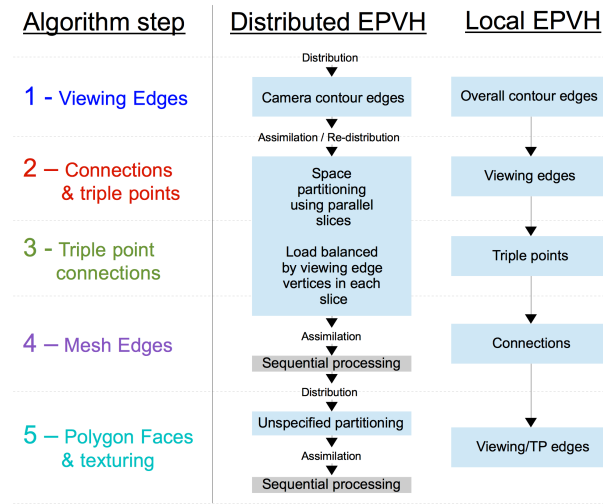


Figure 5.5: Parallel decomposition of EPVH algorithm for the distributed approach Franco et al. [41] and the local approach proposed in this research - For each of the five algorithm steps, the entity over which the parallelisation is achieved is shown in light blue boxes.

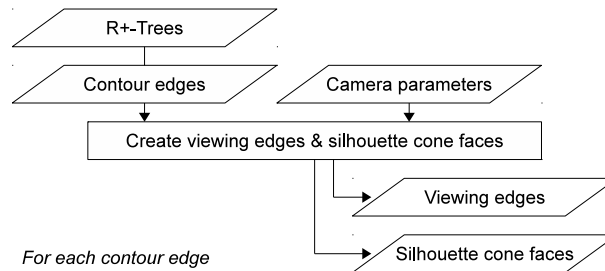
each camera to achieve distributed processing of the viewing lines into viewing edges which achieves a rough load balancing assuming cameras capture images of similar contour complexity. In order to achieve this, the contour information from each camera must first be distributed to all other computers in the processing group. The local approach parallelises over the total number of contour edges regardless of the contour complexity arising from a particular camera and should therefore achieve better load balancing as the total number of contour edges are divided amongst available processors. For the next three algorithm steps, Franco et al. [41] classically employs a space partitioning approach in which 3D space is divided into p parallel slices, where p is the number of computers over which the task is to be distributed. Load balancing is achieved by adjusting individual slice width until each slice has approximately the same number of viewing edge vertices. The space partitioning approach gives rise to boundary conditions in which viewing cone intersection curves are followed until they cross a slice boundary, at which point processing of the remainder of the curve is delegated to the corresponding computer. Parallel slices are then carefully merged together across parallel slice boundaries in a sequential computation phase to form the complete

mesh. The local processing proposed by this research individually parallelises steps 2 to 4 of the algorithm rather than using space partitioning to parallelise them in one block, this removes the requirement to explicitly handle boundary conditions and potentially gives rise to a more adaptive load balancing as the algorithm progresses from one step to the next. For example, in the distributed approach, triple points forming at the locus of three silhouette cone intersections must be handled by the computer assigned to the slice of space in which the triple point arises, and some slices may give rise to more triple points than others. Whereas in the local parallel approach the total number of triple points to search for connections can be evenly divided over the number of processors available. In step 5 of the algorithm, Franco et al. [41] does not specify how the surface extraction phase is partitioned for distributed parallel processing on each computer, only that some form of partitioning is used and that the final model is assembled sequentially. In the local processing approach the mesh formed in the previous step can be used to extract surface polygons and texture coordinates by dividing the work among a subset of the viewing edges and triple point edges determined in steps 2 and 3. The main differences between the distributed and local approaches to parallelisation are therefore:

- Distribution and assimilation of data between parallelisation steps can be eliminated in the local approach due to the shared memory nature of the local approach.
- The space partitioning used in the distributed approach can be replaced with a finer grained partitioning that should lead to better load balancing.
- Sequential processing following parallel execution steps is no longer required to handle boundary conditions arising from the space partitioning.

Parallel Step 1 - Creation of viewing edges

A parallel execution thread is created for every input contour edge. Each thread will generate a single silhouette cone face but can generate any number of output



viewing edges, or none at all.

The start and end contour points defining the input contour edge are projected into 3D space using equation A.0.3 to form two viewing lines which can be used to calculate the surface normal of the silhouette cone face. The viewing line formed from the start point is then used to create viewing edges.

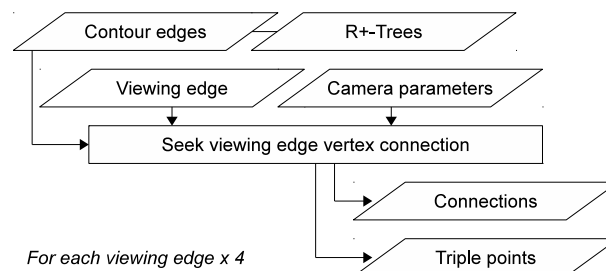
Using the input R+-Tree data structure, the intersection of this viewing line with other cameras' contour edges is tested - Since the contour contributions from each camera are consecutive in the structure, the viewing line need only be re-projected into each camera image once, which is achieved by using equations A.0.5 and A.0.4 to form the minimum and maximum epipolar extent of the line respectively in the camera's image plane. Intersections with contour edges in 2D are converted to depth intervals along the 3D viewing line using equation A.0.6, these depth interval contributions from every camera are intersected to create viewing edges. Each created viewing edge comprises two viewing edge vertices that arose from intersections between the viewing line and a contour edge from a different camera, the index of the generating contour edge is stored for each vertex.

Importantly, viewing edges created by a thread are cached until all the edges for that thread have been created, this enables sequential storage of all the edges generated from an input contour in the viewing edge table. Caching this information will enable significant optimisation of the later search for viewing edge vertices.

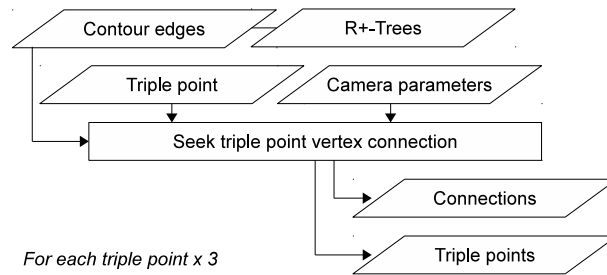
Should a textured model be required, the fact that the thread will access every other camera during the check for intersections can be used to create a list of candidate texture cameras for the polygons that will result from this silhouette cone

face. As each camera is encountered by the thread, candidate cameras are identified as those whose principle ray is pointing in the opposite direction to the surface normal - which may be easily determined from the dot product. The resulting list is sorted by dot product so that it can be used by the later face extraction step 5.6.3 to accelerate the search for the best texture camera.

Step 2 - The search for vertex connections and triple points



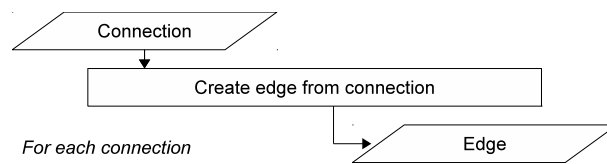
The search for viewing edge vertex connections is implemented as a thread that searches for a single connection. Since each vertex in the final mesh requires three edges, and the two input vertices of a viewing edge are already connected by an edge, 4 threads must be created per viewing edge. Each thread begins the search for a connection by locating its neighbouring vertex in the local contour, or in the contour that intersected to generate it, whichever is closest. The line forming this potential edge is tested for intersection with all other camera contours using the R+-Tree intersection optimisation described in Section 5.6.3. If any intersection is found, the closest one will be a triple point, to which the connection will be made. A thread outputs the connection it finds, which may be a triple point or a viewing edge vertex. Should a new triple point be encountered by a thread, it is created using the intersection of the three silhouette cone faces meeting it and added to the list of triple points.



Step 3 - The search for triple point connections

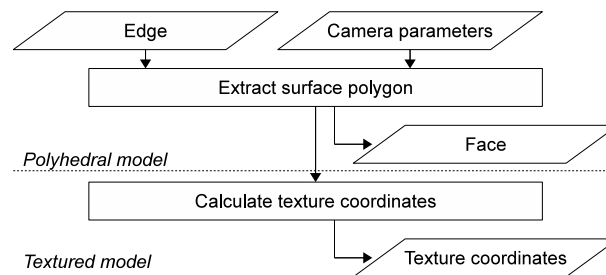
Since each triple point is a regular vertex in the final mesh it requires three connections - 3 threads are created, each of which will search for one connection. The nature of the search is very similar to that described in the previous section, except that it originates from a triple point, and as such there are no neighbouring viewing edges to search for connections to. Instead the two silhouette cone faces defining the connection being sought for the triple point are used to define the direction in which to search for viewing edge vertices that intersect the silhouette cone face. This search is significantly optimised by taking the steps described at the end of Section 5.6.3 Once a connection candidate has been located, the potential edge is searched for triple points, in exactly the same manner as described in the previous section. One triple point can lead to another, and some triple points will only be located when the search begins from another triple point, particularly when many cameras are used. Therefore, the thread outputs any new triple points that are located, as well as the desired connection. When the group of threads that were launched completes, the number of new triple points located is returned, and so a new group of threads can be launched for the new triple points, this iterative process continues until no new triple points are returned.

Step 4 - The creation of edges from connections



A thread is created for each connection determined by the previous steps. Note that exactly two connections should have been located for every edge in the final mesh, one in each direction along the edge. The purpose of this processing step is to bring together both connections to form the final edge, consequently only one of the two instanced threads will output the created edge, and the second one locates it. Execution of this thread for all located connections provides the first check on output model consistency, any errors such as connections missing in one direction indicate a problem upstream. Once all threads have completed execution, there should be $\frac{3v}{2}$ edges created, where v is the number of vertices in the mesh.

Step 5 - The extraction of surface polygons and optional texture coordinates



Surface polygons are extracted from the mesh by following edges created in the previous step and maintaining the same silhouette cone face on the left or right hand side of each edge traversed until the starting edge is reached again. Since each executed thread follows a series of edges to output a single face of the resulting model, care must be taken to only create threads for edges that are known to lead to a unique face, since launching a thread for every input edge would result in many duplicate output faces. Each viewing edge created during step 1 will create a unique face, so a thread is created for every viewing edge. When the final model is complete, each edge created in step 4 will have been traversed exactly twice, once in each direction. Therefore the step can be executed iteratively - by checking edge traversal counts in each direction after a group of threads has completed, it can be determined which edges have not been traversed, and in which direction, so a new group of threads can be created for them. The process of face extraction is complete only when all edges have been traversed once in each direction.

Texture coordinates can optionally be calculated by the thread once the surface polygon has been extracted. In step 1 a list of candidate texture cameras was created, sorted by the dot product of the silhouette cone face surface normal and the camera principle ray. The area of the polygon projected into the image plane of each candidate texture camera is calculated, and this is multiplied by the negated dot product. This provides a crude favouring of cameras with larger projected polygon areas pointing towards the camera. Texture coordinates are calculated by normalising the projected polygon vertices. Note that this scheme does not take account of occlusions that may arise between the chosen camera and the polygon to be textured. Any other polygon in the scene could obscure part or all of the polygon being textured from the camera. Such occlusions become more frequent as more objects appear in the scene, and should be accounted for. However, no adequately fast occlusion checking algorithm has yet been implemented and so has not been included as part of this research. Each executing thread outputs a list of vertex indices making up the face, and optionally a list of associated texture coordinates.

R+-Tree intersection optimisation

The main computational cost of the algorithm arises from searches for intersections of lines with camera image contour edges. This occurs in two distinct places: During the initial building of the viewing edge array when viewing rays are tested for intersection with silhouette cone faces, and during the later searches for triple points. Both scenarios are different in terms of the nature of the line along which the search for intersections is carried out. In the former case, it is known that the ray is emanating from a camera centre projected into another camera's image, in other words the line is epipolar. [79] implements a view-dependent rendering algorithm in which only the first of the two searches for intersections required by EPVH is required. The paper describes two optimisations by which the search for intersections between epipolar lines and contour edges can be reduced, both rely on the fact that only the slope of the epipolar line changes from one projected contour point to the next, and therefore contour edges can be grouped based upon the

range of epipolar lines that could intersect with them. Implementation of this optimisation requires building tables grouping contour edges by epipolar line slope for every camera pair, resulting in hundreds of tables for high camera count systems - For example, a 20 camera system would require building $(19 \times 20)/2 = 190$ tables. Since the search optimisation tables must be created for each frame of data in a real-time system, the cost of building them must be amortised against the performance gain achieved. In the second type of intersection search required, there are no assumptions that can be made about the ray's direction, but it is known to start at the visual hull surface - the epipolar line optimisations are no use in this case.

[40] explains that since the nature of the viewing ray is different in both cases it makes sense to optimise searches in a manner that is agnostic to the viewing ray direction. He describes a method in which the same data structure can be used for both cases, amortising the setup cost over both types of search. Sorted silhouette vertex lists for a fixed number of directions in the 2D plane are calculated, each search then selects the list for which the number of edge candidates are minimised - The description of the method from the paper is insufficient to re-implement a working solution, and no results are provided for comparison.

A tree structure designed for the indexing of spatial data is employed to accelerate the search for intersections in both the aforementioned cases. First introduced in [47], the R+-Tree is a structure that can be employed for indexing multi-dimensional data, and has been used in a wide variety of applications. Similar in concept to the B-Tree of [10], which organises linear data into groups within a tree structure, the R+-Tree is designed to organise multi-dimensional data. In two dimensions, the R+-Tree can reduce the search for spatial data from the exhaustive case by representing data items with a minimum bounding rectangle (MBR) that describes the minimum and maximum extent of the data entity. The tree structure places data items and their MBR at the base of the tree as leaf nodes, these are grouped together into branch nodes at successively higher levels in the tree, each time creating a new MBR representing the spatial extent of all the items in the group. The tree is complete when a single node exists at the top of the tree

representing all the data items, and whose MBR describes the spatial extent of the entire group. Search queries for items falling within a spatial range are reduced by beginning at the root (top) of the tree and following only those branches whose MBR intersects with the desired spatial range - If a search query arrives at the base of the tree, only those leaf items intersecting with the region of interest are reached.

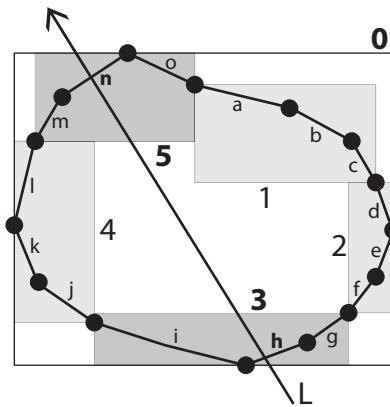


Figure 5.6: A silhouette contour with its overall MBR (0), divided into a number of groups (1, 2, 3, 4, 5). Contour edges are denoted by letters a - o. The line L is passing through elements 0, 3, 5, h and n of the contour.

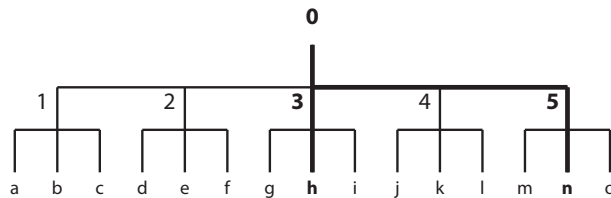


Figure 5.7: R+-Tree structure for contour in Figure 5.6. For the intersection with line L, the exercised branches are emboldened and lead to contour edges h and n, with which L intersects.

Recall that the contour defines a closed region of the camera image which can be represented as a polygon, all pixels falling within the polygon are either part of the object being reconstructed, or a hole passing through it. An ordered list of contour points defines the polygon; any two adjacent contour points in the list define an edge of the polygon, and it is these polygon edges that will need to be repeatedly tested for intersections with during reconstruction. The list of contour points is ordered such that the overall orientation (clockwise or anti-clockwise) denotes

whether the contour represents the object being reconstructed, or a hole through it. Each polygon edge is a straight line from one contour point to the next in the ordered list making up the whole contour (letters a - o in Figure 5.6). These two points can also be used to define the Minimum Bounding Rectangle (MBR) of the polygon edge. Consecutive edges around the polygon can be grouped together to form a MBR enclosing a segment of the contour (numbers 1 - 5 in Figure 5.6), or the entire contour (letter 0 in Figure 5.6). The principle behind the optimisation is that when searching for intersections of an arbitrary line through a camera image with the camera's contour edges, contour edges that fall within a bounding box through which the line does not pass can quickly be ruled out. There are two reasons why this is faster than testing for intersections with every contour edge: a line/box intersection or overlap test is much faster than a line/line intersection, the grouping of a number of contour edges into a single bounding box, and the hierarchical tree structure enables multiple contour edges to be ruled out of the intersection in a single test.

5.7 Evaluation

5.7.1 Methodology

Performance is tested on two modern computers: an Apple Mac Pro with 2 quad core Intel Xeon 5400 "Harperdown" processors running at 2.8GHz, 18GB of memory clocked at 800MHz and an nVidia GTX285 GPU, and an Apple MacBook Pro, with a quad core Intel i7 2.2GHz - 3.3GHz (turbo) processor, and 8GB of RAM clocked at 1333MHz.

Comparison with sequential EPVH (literature based)

The performance of the implementation is tested against timings for the EPVH algorithm published in [40]. In order to achieve this, untextured 3D models are built

from the same input photographs as used in [40], taking care to ensure the number of input contour edges and output vertices and faces are similar to the published figures. Since the referenced work quotes timing figures for the reconstruction of untextured polyhedrons these are what is measured. The datasets used for comparison originate from Lazebnik et al. [69] - Whilst they only provide a single frame for the reconstruction, and are therefore not typical of the dynamic data a real-time system might reconstruct, they do provide inputs of a high complexity that are useful for comparing performance.

Comparison with distributed EPVH (literature based)

The description of the distributed processing approach to accelerating EPVH in Franco et al. [41] provides some scalability results that can be used for a performance comparison; two graphs are provided indicating latency and acceleration over a range of processor counts, and for three different camera sets. The tests are repeated using a simulator in which virtual cameras can be placed around the object to be modelled. The performance data is obtained by reconstructing a synthetic object from 12, 25, 64, and 128 cameras, and measuring the time taken to reconstruct the model over a range of processor counts. This provides latency and acceleration data that can be compared to the literature.

Evaluation of parallelisation

The performance of the parallel implementation is measured when executed in a number of different processing scenarios: single core CPU, multi-core CPU, and multi-core GPU. Each step of the algorithm is measured to assess the impact parallelisation has had. The recorded reconstruction times also provide a means of assessing the performance portability of the OpenCL implementation when executed, unmodified, on two fundamentally different processing architectures, and for comparing both against the algorithm running sequentially on a single core of a conventional CPU.

5.7.2 OpenCL vs OpenMP for parallel CPU execution

The use of OpenMP on the CPU provides an alternative approach to scheduling work on multi-core CPUs, and measurements are taken of the reconstruction performance over a wide range of camera counts in order to compare with OpenCL parallel CPU execution.

R+-Tree intersection

In order to test the R+-Tree optimisation, the "Alien" dataset from Lazebnik et al. [69] was used. This provides 24 high resolution camera images, with a high contour complexity. The reconstruction time is first generated from the contours as they appear in the dataset, and then for the same images with successively simplified contours, to achieve a progressive reduction in contour complexity. The OpenCL implementation was used, running on the multiple cores of the Xeon CPU. Timings were collected for the first 3 steps of the algorithm, as the remaining steps do not search for intersections. The time taken to complete each step is measured, with and without the optimisation to calculate the change in performance over a range of model complexities. The time taken to create the structure is also measured, as this must be offset against the performance gain.

Evaluation using humans

Pre-captured datasets are used to perform these comparisons, all comprising real images of humans taken with a variety of camera configurations. In this comparison, the time taken to create the fully textured output is measured, this gives a meaningful figure for the performance of the algorithm in the context of an end to end tele-immersion system. Performance is measured for sequential CPU, parallel CPU and GPU: a 2 x Intel Xeon 2.8GHz quad core CPU, an Intel i7 2.2GHz quad core CPU, and an nVidia GTX 285 graphics card.

5.8 Results

Table 5.1: Key to graph labels

Graph label	Description
GPU 30 core	nVidia GTX 285 graphics card
i7 4 core	intel i7 at 2.2GHz running on 4 cores
Xeon 8 core	2 x intel Xeon quad core 2.8GHz
i7-seq	intel i7 2.2GHz running sequentially
Xeon seq	intel Xeon 2.8GHz running sequentially
Inria	Timing taken from literature [40]

5.8.1 Comparison with sequential EPVH

Figures 5.8 and 5.9 show the reconstructed polyhedrons and timings for the "Alien" and "Skull" datasets courtesy of Lazebnik et al. [69]. The time quoted in [40] is provided and compared to our own reconstruction times for a model of similar complexity - The output models from our implementation have slightly more vertices than the Inria reconstruction, this is due to the necessity to simplify the input contours, which are not provided in the dataset at the resolutions used in [40]. It can be seen from the graphs that the sequential implementation has a similar execution time to that cited in the literature - Whilst our timings may appear slightly faster, the Inria models were reconstructed on a 2.0GHz computer - a little slower than the computer used, and undoubtedly with slower memory.

5.8.2 Comparison with distributed EPVH

Figure 5.10 shows the effect increasing the number of processors has on latency for various camera counts. Figure 5.11 shows the acceleration achieved for the same camera configurations as the number of processors is increased. The "optimal" line is the line representing perfect parallelisation, where the task is accelerated by the same amount as the number of processors assigned to the task. It

Alien

Number of cameras	24
Total input contour points	14028
Average points per camera	584
Polygons	29794
Total vertices	14967
Triple points	4161

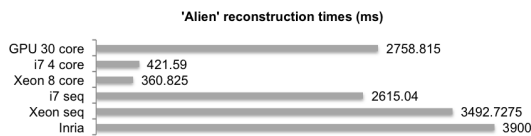
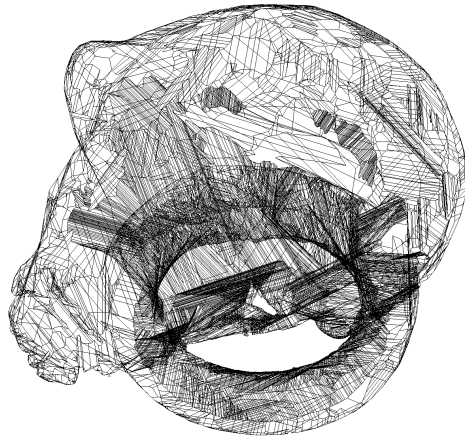


Figure 5.8: Polyhedral reconstruction and timings for "Alien" from 24 camera images averaging 1600 x 1600. Dataset courtesy of Lazebnik et al.

is clear that as the number of cameras is increased, and consequently the model complexity increases, the parallel processing acceleration tends towards the optimal line. Compared with the measurements in Franco et al. [41], our latency measurements are quite a bit lower, but this is to be expected due to the removed dependency on network data transmission. The acceleration achieved is closer to the optimal line than in Franco et al. [41] - This suggests that the finer grained parallelism employed by the partitioning of the parallel workload is more effective than the three stage stream parallelism used in the distributed approach. Alternatively, the distributed approach itself is give rise to an increased latency (as shown in Figure 5.10), which will reduce the overall speed up achieved by distributed parallelisation when compared to local parallelisation.

Skull



Number of cameras	24
Total input contour points	10418
Average points per camera	434
Polygons	46227
Total vertices	23282
Triple points	2680

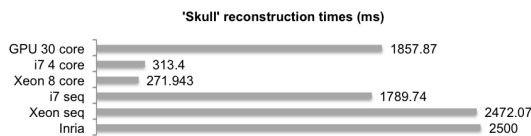


Figure 5.9: Polyhedral reconstruction and timings for "Skull" from 24 camera images averaging 1600 x 1600. Dataset courtesy of Lazebnik et al.

5.8.3 Evaluation of parallel implementation

The acceleration achieved by each step of the parallel partitioning can be seen in the following graphs, for three different processing scenarios: CPU sequential, CPU parallel, and GPU. Note that the implementation is currently not optimised for GPU execution, which could significantly improve performance on the GPU.

Step 1 - Create viewing edges

Figure 5.12 shows the execution time for step 1 of the parallel scheme. It can be seen that as input contour edges increase, the sequential CPU execution time increases more sharply than GPU parallel execution, or CPU parallel, which is the fastest. The CPU and GPU parallel executions both provide an acceleration

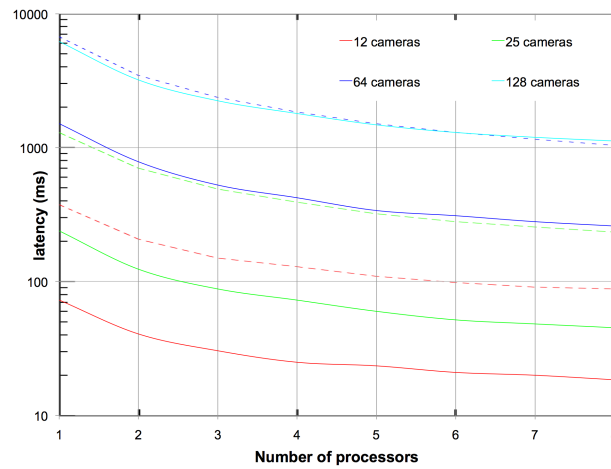


Figure 5.10: Parallel latency reduction compared to the distributed approach - Dashed lines show the corresponding latency reduction of distributed EPVH from Franco et al. [41]

compared to sequential execution on the CPU.

Step 2 - Find Connections

In Figure 5.13, the execution times are plotted against the number of viewing edges, since this is the data entity over which the step is parallelised. It can be seen that this step, and the following step (Figure 5.14) tell a different story to step 1. As the camera count is increased, more triple points are found by step 2, which leads to more connections being sought in step 3.

Step 3 - Triplepoint Connections

Triplepoint connection seeking acceleration can be found in Figure 5.14. Similarly to step 2, triplepoint connection seeking performs poorly on the GPU, worse than the sequential execution on the CPU.

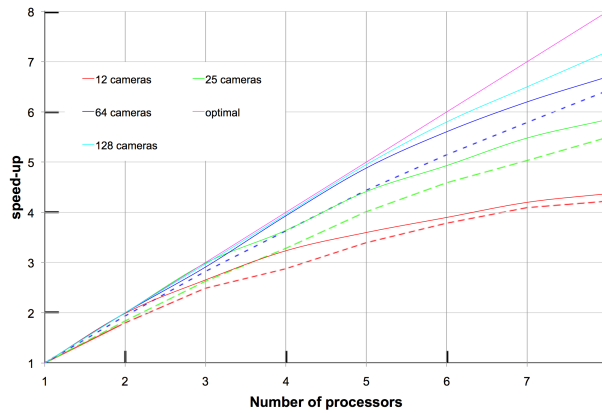


Figure 5.11: Parallel acceleration of the implementation compared to the distributed approach - Dashed lines show the corresponding acceleration of distributed EPVH from Franco et al. [41]

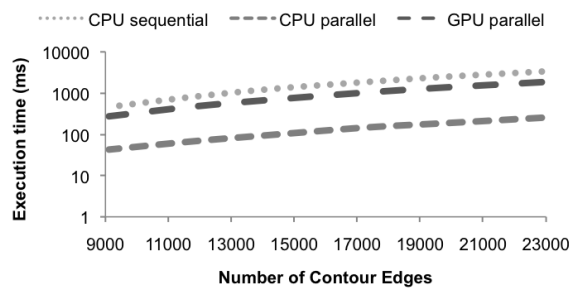


Figure 5.12: Step 1 - Create Viewing Edges execution time graph.

Steps 4 and 5 - Edges and Faces

Figure 5.15 Shows the combined time taken to extract edges and faces, including texturing. Very little gain is made over sequential CPU processing for either parallel CPU or GPU.

Overall performance

Overall, it can be seen from Figure 5.16, that CPU parallel processing is the quickest way to process the algorithm. CPU sequential, and GPU parallel execution ap-

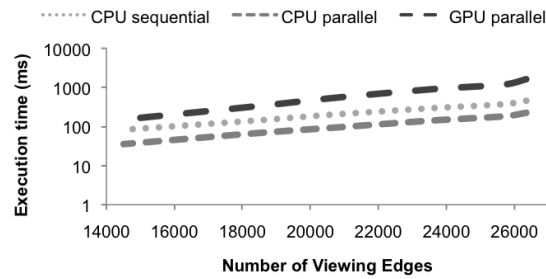


Figure 5.13: Step 2 - Find Connections execution time graph.

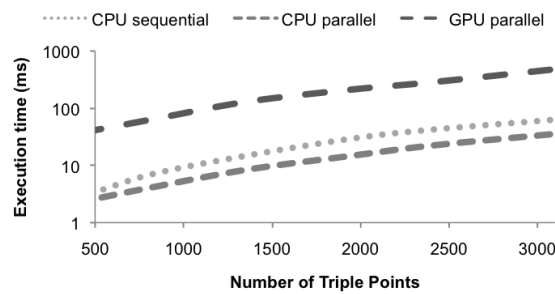


Figure 5.14: Step 3 - Find Triple Point Connections execution time graph.

pear to follow a very similar line in the graph, however, on analysis of the graphs from each parallelisation step it is evident that the GPU can process contour edges into viewing edges more quickly than the sequential CPU.

Acceleration achieved per step

Figure 5.17 shows the acceleration achieved for each step of the algorithm when executed in parallel on an 8 core CPU.

Branching analysis

GPUs are batch stream processors and as such are not efficient at executing code containing conditional statements (branches). This is because execution threads

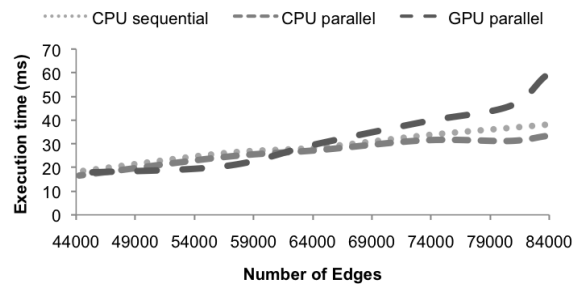


Figure 5.15: Steps 4 and 5 - Extract edges and faces execution time.

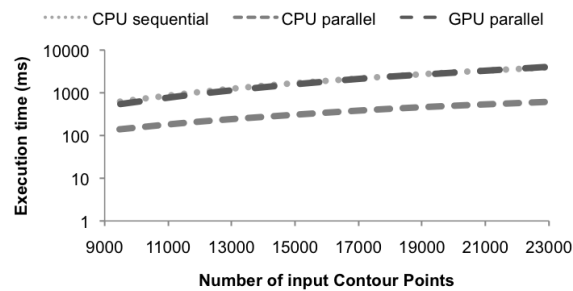


Figure 5.16: Overall execution time.

are grouped together to execute code in single instruction multiple data (SIMD) groups to save on the required number of instruction fetches and decodes. If threads encounter a branch they may diverge resulting in some threads in the group executing the branch whilst the others are stalled or executing a different code path. Effectively for each branch reached the execution group is split into two, which reduces the efficiency of both thread groups due to the loss of SIMD parallelism. For example, in a 4 thread SIMD group encountering a branch where 1 thread follows the branch and the remaining 3 do not, the original SIMD group will be split into 2 new groups, one of 1 thread and one of 3 threads. The 1 thread group will run at 25% efficiency and the 3 thread group at 75% efficiency compared with the original 4 thread group. Nested branches, where conditional code is contained within conditional code further exacerbates the problem, and SIMD execution of deeply nested branches can lead to very poor performance. CPUs on the other hand are not designed as batch stream processors, they have long instruction pipelines compared with GPUs resulting in branching generally incur-

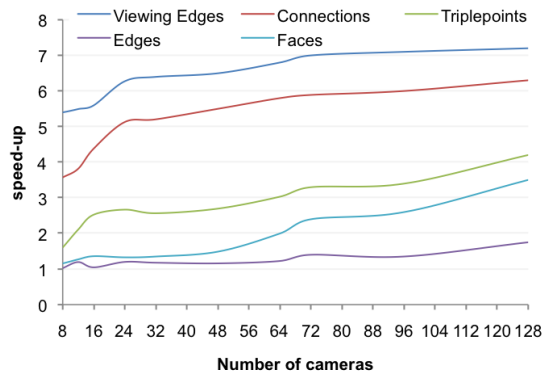


Figure 5.17: Acceleration of each algorithm step achieved on 8 core CPU

ring a small penalty for SIMD code execution. Furthermore their instruction and data cache size and logic is much deeper than their GPU counterparts and they can be flushed and reloaded in fewer cycles. Hence code containing many nested branches will execute much more efficiently on multi-core CPUs than on GPUs. The implementation of the EPVH algorithm for local processing comprises the aforementioned 5 steps, each of which has differing numbers of branches and levels of branch nesting. In order to evaluate the differences in the branching penalty that may be incurred on the GPU and CPU, each stage is analysed for the number of branches and the nesting level at which these branches occur. A simple strategy is employed to roughly assess the branching cost of each step of the algorithm. The number of branches at each nesting level is multiplied by the nesting depth resulting in more deeply nested branches having a greater weighting on the overall branching cost for that algorithm step. For example:

```

if (a) // level 1 branch 1
{
    if (b) // level 2 branch 1
    {
        if (c) {} // level 3 branch 1
        if (d) {} // level 3 branch 2
    }
}

```

}

The branching cost would be $(1 \times 1) + (2 \times 1) + (3 \times 2) = 9$. Table 5.2 shows the corresponding calculation for each step of the implemented algorithm.

Algorithm step	level 1	level 2	level 3	level 4	level 5	branching cost
Step 1	9	13	2	1	1	50
Step 2	3	8	2			25
Step 3	5	2				9
Step 4	1	4	1			12
Step 5	1	2	2	2	2	29

Table 5.2: Branching cost calculation for each step of the algorithm

5.8.4 Comparison of OpenCL vs OpenMP performance

Figure 5.18 shows the acceleration measured for OpenCL and OpenMP. Both are measured in relation to the sequential run of the algorithm. The data is shown for a wide range of camera counts, and shows that a better performance increase is achieved as more cameras are added. Since this data was collected using 8 CPU cores, the optimal acceleration would be 8 times the sequential execution time. The graph indicates that acceleration would tend towards the optimal value as the camera count increases to infinity.

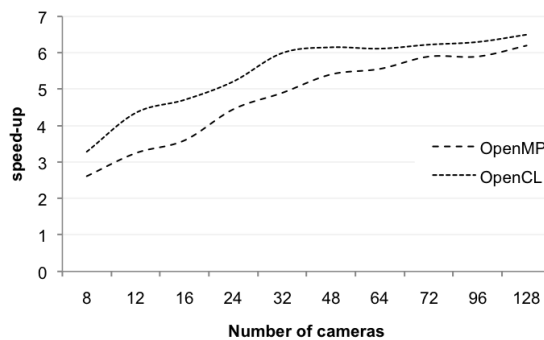


Figure 5.18: Performance increase OpenCL vs OpenMP running on 8 CPU cores.

5.8.5 R+-Tree optimisation results

Building the R+-Tree intersection data structure setup takes approximately 1 millisecond for every two thousand input contour points. Figure 5.19 shows the speed-up achieved by using the R+-Tree structure against the contour complexity. The R+-Tree performance increases as contour complexity increases. There are no published details of the intersection optimisation employed in [40], or timings to compare to.

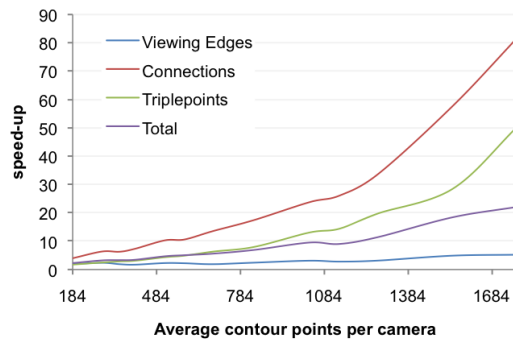


Figure 5.19: Performance gain of the R+-Tree intersection optimisation over a range of camera counts.

5.8.6 Evaluation using humans

Figures 5.20 - 5.27 show the results from tests comparing the time taken to reconstruct the fully textured model using the sequential CPU implementation and parallel implementation executed on multiple CPU cores and the GPU. The reconstructions are from a variety of datasets with varying camera configurations and resolutions. All reconstructions are of humans, enabling the building of a sense of quality expectation against reconstruction time in the context of a telepresence system.

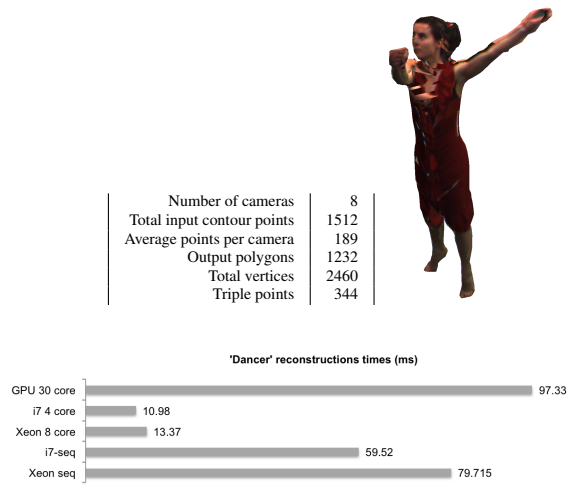


Figure 5.20: Textured reconstruction and execution times for "Dancer" from 8 camera images at 780 x 582. Dataset courtesy of Inria

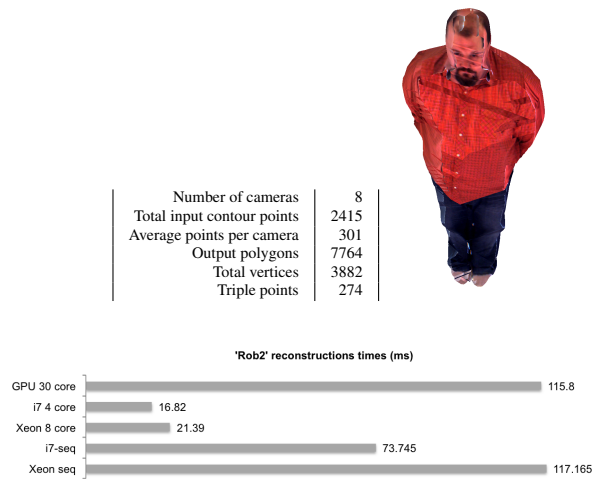


Figure 5.21: Textured reconstruction and execution times for "Rob2" from 8 camera images at 1000 x 1000.

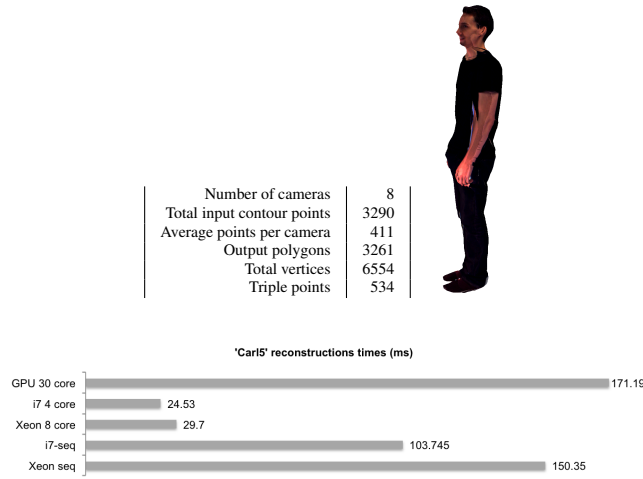


Figure 5.22: Textured reconstruction and reconstruction times for "Carl5" from 8 camera images at 1000 x 1000.

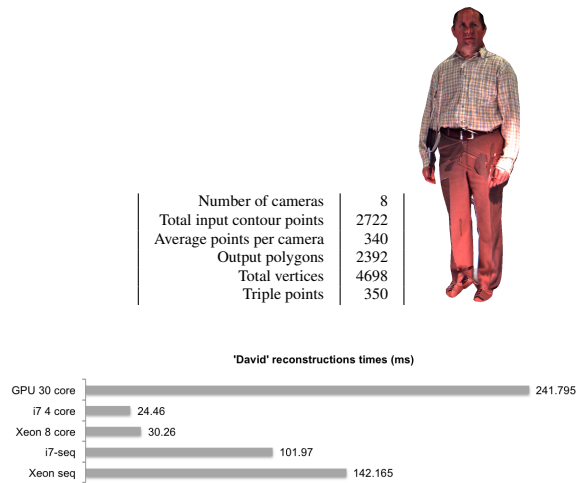


Figure 5.23: Textured reconstruction and execution times for "David0" from 8 camera images at 1000 x 1000.

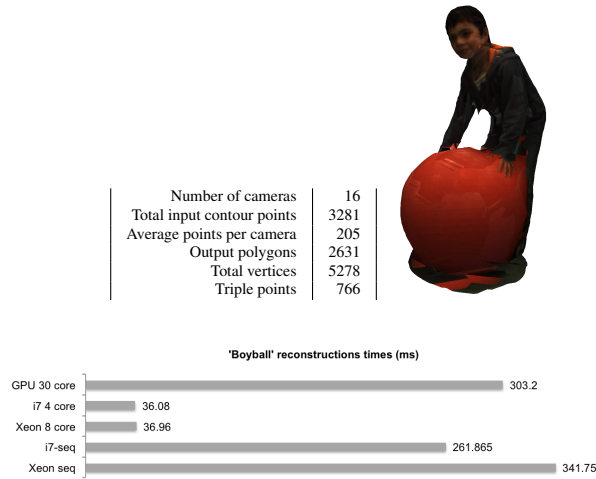


Figure 5.24: Textured reconstruction and execution times for "BoyBall" from 16 camera images at 1624 x 1224. Dataset courtesy of Inria

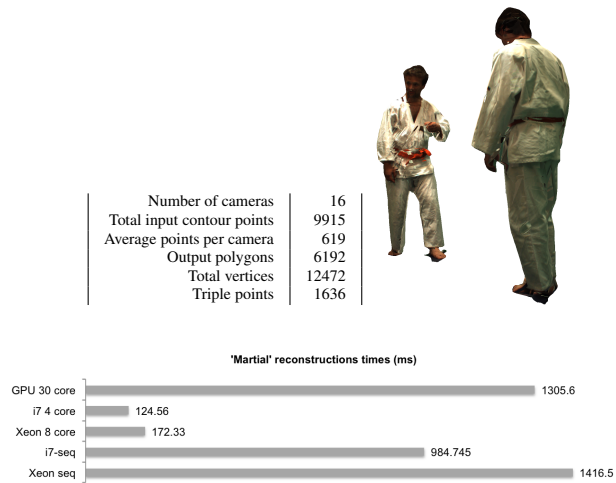


Figure 5.25: Textured reconstruction and execution times for "Martial" from 16 camera images at 1624 x 1224. Dataset courtesy of Inria

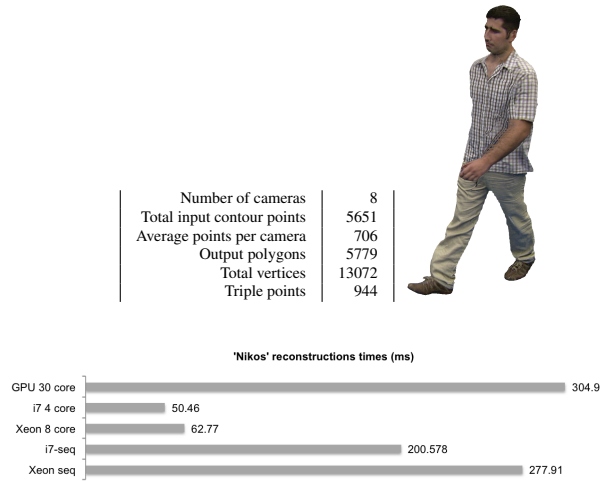


Figure 5.26: Textured reconstruction and execution times for "Nikos" from 8 camera images at 1920 x 1080. Dataset courtesy of University of Surrey

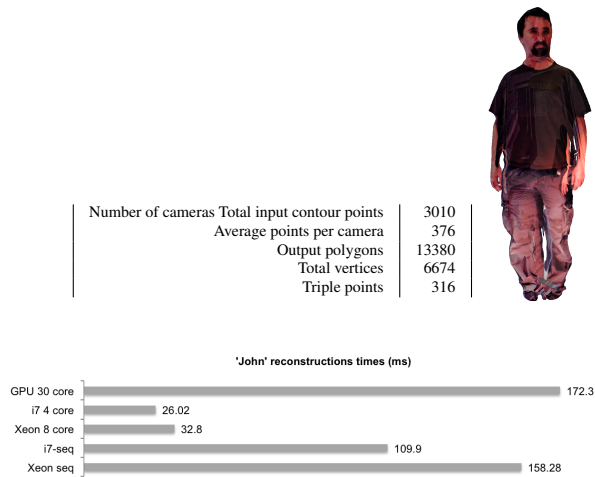


Figure 5.27: Textured reconstruction and execution times for "John" from 8 camera images at 1000 x 1000.

5.9 Discussion

5.9.1 Comparison with sequential EPVH

It has been shown that the sequential implementation of the EPVH algorithm performs as well as the state-of-the-art implementation described in the literature. This proves that the implementation provides an adequate benchmark against which to analyse optimisations provided by the parallelisation strategy.

5.9.2 Comparison with distributed EPVH

A comparison was made between the network distributed EPVH and an implementation adapted for local multi-core processing and it was found that local multi-core processing achieved marginally closer to optimal parallelism than distributed EPVH. However, since the comparison was performed from figures quoted in the literature Franco et al. [41] the observed result, which is marginally closer to optimal parallelism, could be due to the removal of network latency present in the distributed approach. The same can be said for the much lower latency results of the local processing implementation compared with the network distributed approach. What is clear is that local parallel processing achieves both good acceleration and latency results that are suitable for real-time reconstruction from video streams.

5.9.3 Evaluation of parallelisation

Figure 5.17 showed the acceleration achieved per parallel partition. The partitions that benefited were those of viewing line creation, connection, triple point seeking and surface polygon generation. Edge generation did not benefit significantly from parallelisation, even at high camera counts.

multi-core CPU vs GPU parallelisation

The parallel implementation was tested on CPUs and a GPU in its unmodified state - It is clear from the results that, if any real benefit is to be made of their massive multi-core potential, the implementation requires some specific optimisation targeting GPUs. It has been shown that using the GPU can enhance performance, but only in the first step of the algorithm since overall execution on the GPU performed similarly to the sequential implementation running on the CPU. The GPU has more cores than the CPU for the machine on which the tests were performed, but these cores are clocked at a lower speed than the CPU. Specifically, the nVidia GTX285 GPU has 30 cores clocked at 1.476 GHz, whereas the Intel Xeon CPU has 8 cores clocked at 2.8GHz. The theoretical maximum processing capability can be calculated by multiplying the core count by the clock speed, giving 44.28GHz for the GPU, and 22.4GHz for the CPU. Therefore, taking into account core count and clock speed alone, one might expect the GPU to execute approximately twice as quickly as the CPU cores, however this is not the case. GPU and CPU cores differ considerably in terms of resources available: CPUs have caches that significantly speed up memory accesses, whereas GPUs do not. Conditional code results in branching on the GPU, for which there can be large performance penalties, but the same has no impact on CPU performance. Steps 2 and 3 of the implemented parallelisation approach perform less well on the GPU than when executed sequentially on the CPU. The branching analysis performed suggests that branching is not the reason for this poor performance as step 1 performed better on the GPU than the sequential CPU and has a higher branching cost than either step 2 or step 3. The likely reason for poor performance of steps 2 and 3 is the amount of global memory access required. Both steps need to read from large intersection tables, that are created in step 1 and stored in global memory, in order to determine the correct candidate for forming the connection. Manual caching of these tables into shared or local memory on the GPU is not an option as it would limit the scalability of the implementation to a low number of cameras, or camera contour complexity (camera resolution), due to the fixed size of local or shared memory present on the GPU. Hence the poor performance of the implementation

on the GPU can be described as a combination of the algorithm design and the speed of global memory accesses on the GPU compared with general memory access on the CPU. In all cases, the OpenCL implementation running on the CPU achieves the fully textured reconstruction in a fraction of the time taken for either the sequential CPU execution, or the GPU execution - This is to be expected as no specific GPU targeted optimisation has yet been implemented. Such optimisations include branch elimination where possible, and manual caching of data to benefit from the high throughput local memory that GPUs have access to. Such caching could also reduce global memory access time, which is very expensive on a GPU compared to a CPU. The necessity to implement specific optimisations to obtain performance portability from a supposedly portable language such as OpenCL is simply an artefact of the emerging nature of general purpose computing.

5.9.4 OpenCL vs OpenMP for parallel CPU execution

OpenCL performed marginally better than OpenMP, both implementations showed a similar trend in performance increase as the number of cameras increased. Since both implementations are almost identical, the performance difference is surprising as they were measured running on identical hardware. What is also surprising is that OpenCL should, in theory, perform worse than OpenMP as there is more thread scheduling and memory management overhead. It is likely that the observed behaviour is due to OpenCL automatically targeting the correct processor instruction set extensions, whereas OpenMP generally requires extra work to ensure the executable includes both SSEx and AVX code paths. OpenCL vector primitives can help express explicit parallelism without the portability sacrifices that come from using SSE intrinsics to achieve the same in OpenMP.

5.9.5 R+-Tree intersection

It is clear from Figure 5.19 that a greater acceleration is achieved for the connection and triple point steps of the algorithm than for creating viewing edges. This is likely because in the viewing edge step long epipolar line segments that intersect many contour MBRs require more intersection tests than the lines in the connections and triple points steps, which generally give rise to very short line segments only intersecting with a single contour bounding box. This analysis suggests that algorithm performance could be further enhanced if the search for viewing edges was replaced by another lightweight and quick to build structure.

5.9.6 Evaluation using humans

The reconstructions of various humans and their associated timings for multi-core CPU processing demonstrates that high quality 3D reconstructions suitable for applications like telepresence can be generated at real-time frame rates.

5.10 Conclusion

The study proposed a method of parallelisation tailored to execution on a single machine and provided indicative results for example multicore CPU and GPU implementations. The theoretical standpoint is that by considering parallelisation without distribution, we were able to tailor the parallelisation of each step. As well as this theoretical contribution, this study has provided some useful practical results. It has been shown that the parallel EPVH implementation running on a multi-core CPU of a commodity computer is capable of producing high quality fully textured models of humans captured by 8 high resolution cameras in approximately 25ms, making it highly appropriate for real-time applications such as telepresence. However, we were not able to achieve comparable speed ups from multicore GPUs. Our findings suggest that given enough CPU cores, increas-

ingly complex models could be reconstructed in real-time without the need for network distributed processing. While our study compares results to those of the distributed approach, it does not attempt to reimplement the distributed approach on a single machine. Furthermore it does not attempt to inform how our approach might work across any CPU/GPU implementation.

The implication of our findings is that network distributed processing of the EPVH algorithm can be replaced with multiple core CPU parallel processing, using a scheme that tailors parallelisation to each stage. This simplifies system design, eliminates unnecessary network latency, and increases utilisation of processing resources. In the context of a 3D telepresence system this is an important finding, as network distributed processing creates the opportunity for jitter in latency of an end-to-end system that could confuse the flow of communication.

Chapter 6

Evaluation

This chapter describes the methods and processes by which the suitability of 3D reconstruction in the context of a 3D telepresence system has been evaluated. During the course of this research a sophisticated platform for 3D reconstruction from video was developed that has numerous practical and analytical uses. The features of the platform and how they have been used to develop understanding, validate requirements and reduce time in prototyping algorithms and camera placement are first described. Then, by means of two case studies, it is shown how the platform can be used to investigate the impact of camera placement on visual and spatial quality of 3D reconstruction. The first case study begins by investigating the impact of camera placement on the overall spatial constraint of objects, this is followed by a deeper analysis of the impact individual cameras have on the spatial and visual quality of a human head reconstruction. The second case study summarises a collaborative experiment underpinned by the platform, that showed for the first time that 3D reconstruction is able to convey eye gaze to accuracies sufficient for human social interaction.

The work describing the software utility presented in this chapter has been published in [35], and the collaborative experiment has been published in [104].

6.1 Objectives and research questions

O3: Develop a platform through which the impact of camera placement on spatial and visual quality of 3D reconstruction can be studied.

Q3: What are the requirements of a system with which the impact of camera placement on spatial and visual quality can be studied?

Given that for a particular multi-core processor temporal quality of the reconstruction algorithm will depend upon the complexity of inputs provided to it, the ability of the algorithm to achieve interactive frame rates will be largely determined by the number and resolution of cameras used. Reducing the number of cameras, or their resolution, to achieve real-time performance is likely to impact upon spatial and visual quality of the reconstruction. The spatial constraint achieved by the visual hull is known to improve as the number of cameras increases and their viewpoints of the object under reconstruction are diversified. The spatial and visual quality of reconstructions is likely to be improved as camera resolution increases. The real-time constraints of a telepresence system lead to a fixed budget in terms of the number of cameras and their resolution for a particular processor. Therefore, achieving the most faithful reconstructions within these constraints requires an understanding of the impact camera placement and resolution have on spatial and visual quality of the output.

6.2 3DRecon, a utility for investigating quality in 3D reconstruction

In order to develop a deeper understanding of many aspects of the VBR process, 3DRecon was developed. 3DRecon is an interactive utility application that can be used to further understanding of many aspects of the VBR process, including the impact of camera placement on form and texture genesis. The tool allows in

depth analysis of individual frames and sequences of frames forming a 4D animated sequence. Similar tools already exist for the analysis of pre-reconstructed geometry, but to the best of our knowledge this is the first such tool that includes the reconstruction back-end, enabling interactive experimentation with the camera set in terms of which cameras contribute to generation of form and texture. Furthermore, a built in simulator provides an environment for rapid prototyping of algorithms or camera configurations prior to real-world deployment.

The principle features of 3DRecon are:

- Interactive control over camera selection for form and texture genesis.
- A built in simulator enables rapid prototyping using virtual cameras and synthetic objects.
- Runs from pre-recorded datasets on a filesystem, or with live cameras.
- Provides an algorithm "plugin" API, enabling rapid prototyping and evaluation of 3D reconstruction algorithms.

6.2.1 Related work

LucyViewer¹ is an open source application capable of viewing 4D data in a manner similar to 3DRecon. The primary difference being that LucyViewer takes as input pre-reconstructed models in the form of geometry files and textures. Hence LucyViewer can not be used to determine the effect individual cameras have on the form of the reconstructed object. Control over texture application from particular cameras is however possible. Lacking the entire reconstruction backend, LucyViewer is also not able to provide the simulated setting provided by 3DRecon.

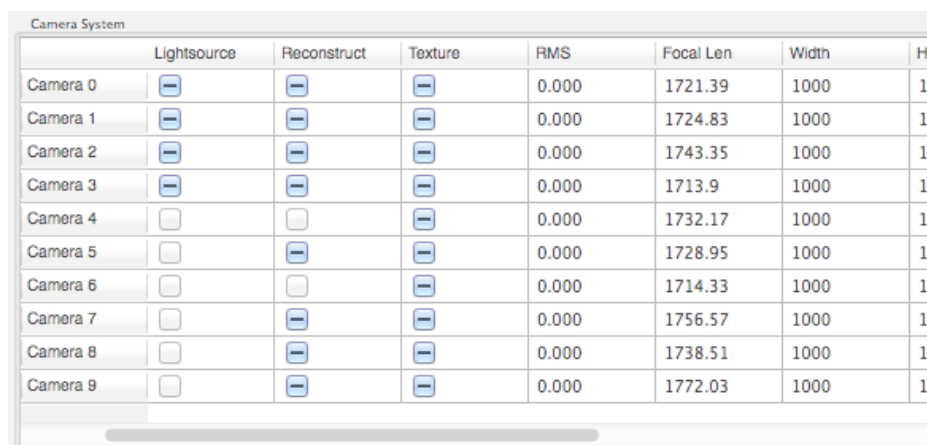
¹http://4drepository.inrialpes.fr/lucy_viewer

6.2.2 Overview of 3DRecon's features

3DRecon consists of three main components:

- Camera interface.
- Viewer.
- Simulator.

6.2.3 Camera interface



Camera System	Lightsource	Reconstruct	Texture	RMS	Focal Len	Width	Ht
Camera 0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.000	1721.39	1000	1000
Camera 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.000	1724.83	1000	1000
Camera 2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.000	1743.35	1000	1000
Camera 3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.000	1713.9	1000	1000
Camera 4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0.000	1732.17	1000	1000
Camera 5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.000	1728.95	1000	1000
Camera 6	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0.000	1714.33	1000	1000
Camera 7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.000	1756.57	1000	1000
Camera 8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.000	1738.51	1000	1000
Camera 9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.000	1772.03	1000	1000

Figure 6.1: The camera manger

The camera interface provides a means by which multiple cameras may be connected to the reconstruction algorithm. Cameras may be added individually, or in groups from XML files defining their properties and positions, or configured through a network connection. Additionally, each camera may derive its image data from a filesystem or through a network interface, allowing for both pre-recorded and live data. Cameras may be individually switched on and off for the purpose of contributing to both the form reconstruction and texturing. Colours are used to identify each camera, and in high camera count systems close colours

can be disambiguated by using the "Look from" function which views the reconstruction from the selected camera (see Figure 6.2(a)).

6.2.4 Viewer

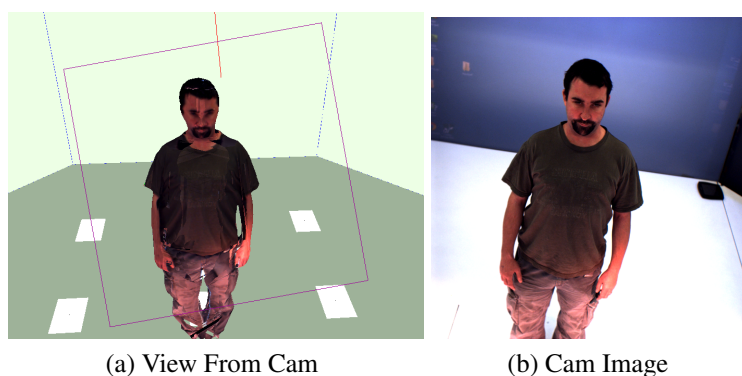


Figure 6.2: Reconstruction viewed from a camera, and the original camera image

The viewer component provides the visual interface through which the user interacts with 3DRecon. A general purpose scene graph is employed to display 3D geometry from the reconstruction algorithm and camera manager, and this can be manipulated by the user to view the scene from arbitrary viewpoints. The viewer does not implement any form of volumetric rendering, all primitives must be based on vertices, lines and triangles or polygons. Volumetric algorithms are therefore required to derive a surface before they can be rendered.

Camera images (Figure 6.2) can be loaded from pre-recorded datasets on a filesystem, or streamed live via a network connection. The viewer provides an interface with which to select individual frames from a sequence, or to start and stop streaming the whole sequence. When used in network acquisition mode, the viewer can be paused for inspection of a particular frame.

At the level of an individual frame, the viewer provides a number of auxiliary primitives that can be used to inspect the reconstructed form. Such primitives may vary from one algorithm to another, so an API is provided for defining custom

primitives specific to a particular algorithm implementation.

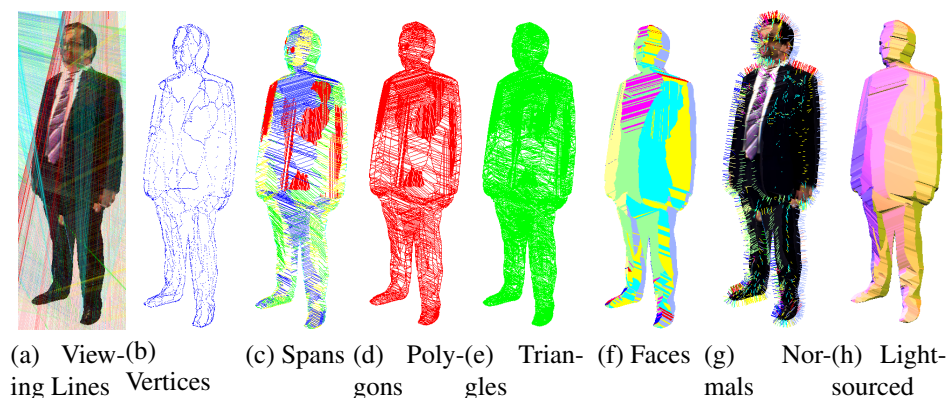


Figure 6.3: Auxiliary primitives for the EPVH algorithm

For the EPVH algorithm the viewer implements the following primitives, some of which are illustrated in Figure 6.3: Viewing lines, vertices, spans, triple points, polygon edges, polygon surface normals, polygons, triangles, triangle edges, faces, textured polygons.

Viewing lines are of use in understanding the shape-from-silhouette process in the formation of the visual hull. They are the projection into 3D space of the points defining the contour that is the edge of the silhouette image of the object being reconstructed. The complete set of these rays emanating from a camera is known as the silhouette cone. Provision is made to switch individual camera and contour silhouette cones within the viewer, which can be very useful in understanding the contribution of a particular camera.

Vertices, spans, polygons and triangles are largely of use for debugging and verifying that the underlying primitives upon which the output is based are sound. Polygons and triangles can be useful when trying to understand how a reconstructed surface is composed. Polygons can be coloured to denote from which camera silhouette cone the surface polygon is derived, which can be useful in determining individual camera contributions to the visual hull.

The remaining primitives are useful for application of a texture to the recon-

structured form. Faces provides a solid geometry in which each polygon can be coloured to denote the camera from which texture is derived. Normals displays the surface normal of each polygon, and also uses colour to denote the texturing camera. Lightsourced provides a solid untextured model that is illuminated by lights positioned at each camera, and shining with the colour associated with that camera. The resulting coloured model is useful in understanding the scope of each camera's view as occluded geometry does not get illuminated by that camera's coloured light.

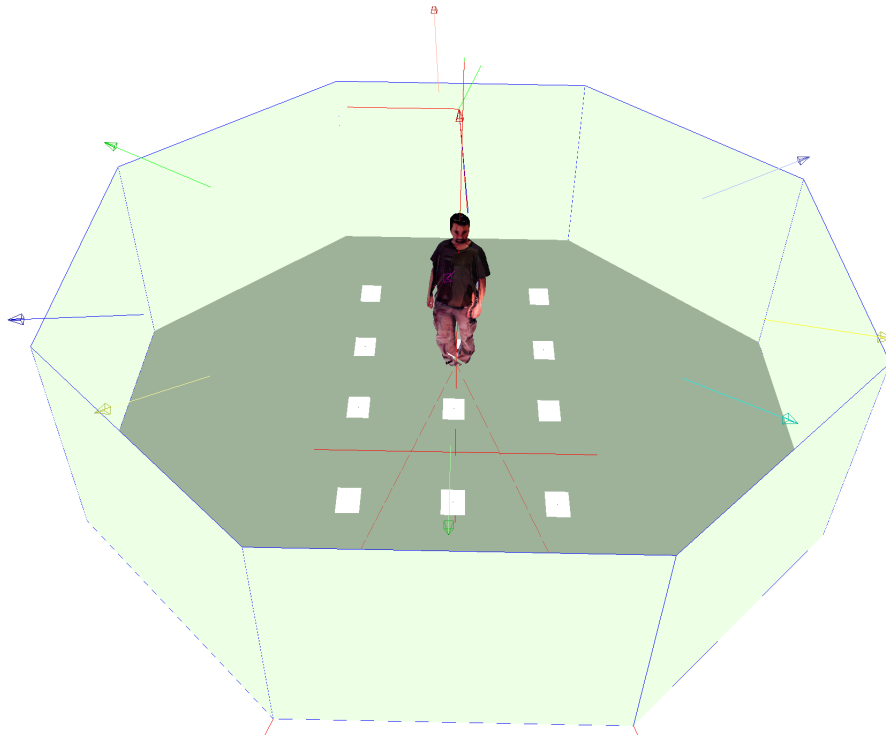


Figure 6.4: The reconstruction viewer displaying a reconstruction, environment and cameras

The viewer also provides a mechanism through which a virtual environment may be programmatically defined, or imported from a file and oriented into the reconstruction coordinate system. Linear and point features present in camera images can be used to define axis unit vectors and an origin respectively, providing the necessary scaling and orientation information to place the reconstructed form within a pre-defined environment. Figure 6.4 shows the viewer displaying a re-

construction of a human within a programmatically defined model of the Octave laboratory at Salford University. The cameras used in forming the reconstruction are also shown. The positions of these cameras, and the reconstructed human relative to the model of the Octave are accurate.

6.2.5 Simulator

The simulated setting allows 3D reconstruction to be studied without any real cameras or datasets. Any number of virtual cameras can be arranged around a synthetic object and their images used to drive a 3D reconstruction algorithm. The simulated setting proved extremely useful early in research to accelerate the process of prototyping and evaluation of different 3D reconstruction techniques and algorithms.

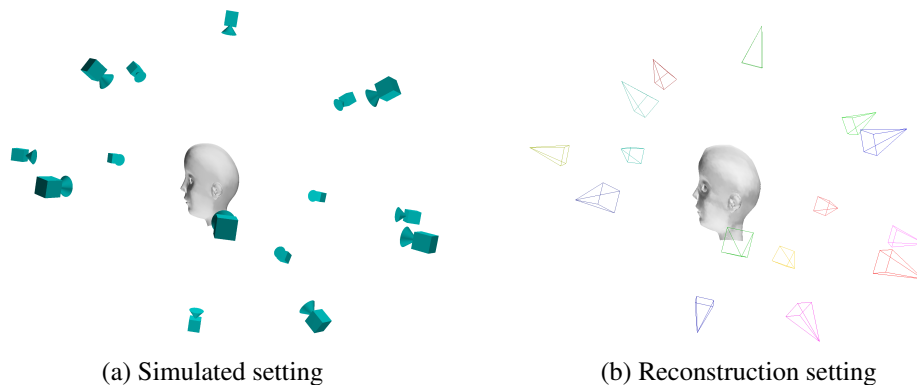


Figure 6.5: The simulator. (a) Virtual cameras surround a synthetic object (b) Reconstructed model

Figure 6.5 shows the simulated setting and the corresponding reconstructed output. A 3D model is loaded from disk and virtual cameras are placed around it, images from these cameras are used to create the reconstructed output in the same way as with real cameras or datasets.

The virtual cameras used in the simulator do not suffer from image noise, and can therefore provide perfect background segmentation for synthetic objects. The

lack of calibration errors, image noise, lens distortion, background subtraction artefacts and issues relating to physical camera placement in the real world makes the simulator an ideal tool for prototyping and evaluation of reconstruction algorithms.

The simulator is also useful for rapid prototyping of camera arrangements prior to real-world deployment, which can be a time consuming process. Placements can be tested for their impact on reconstruction quality, refined and evaluated numerous. Images of the object from the virtual cameras can be viewed during placement, allowing fine grained adjustments to their placement.

6.3 Case study 1: Camera Placement

6.3.1 Spatial quality: overall constraint of the visual hull

Recall that the reconstructed form is constructed at the boundary of volume(s) enclosed by the intersection of silhouette cones formed from the back-projection of points defining the silhouette's contour edge. Using the viewing line primitives provided by 3DRecon, these silhouette cones can be visualised, which can help in selecting appropriate cameras for reconstruction of the form.

Using a reconstruction of a human as an example, a subset of three cameras is selected for reconstruction. Figure 6.6 (a) and (b) show the different reconstructed forms achieved by changing a single camera. In (a) the 3 cameras are all situated to one side of the object being reconstructed and lead to an asymmetrical reconstruction due to the nature of silhouette cone intersections. In (b) the 3 cameras are symmetrically arranged around the object being reconstructed leading to a tighter silhouette cone intersection constraining the reconstructed form.

Low camera counts will generally yield a poor approximation to the form of the object, and even in the case of non-symmetrical camera placement increasing the

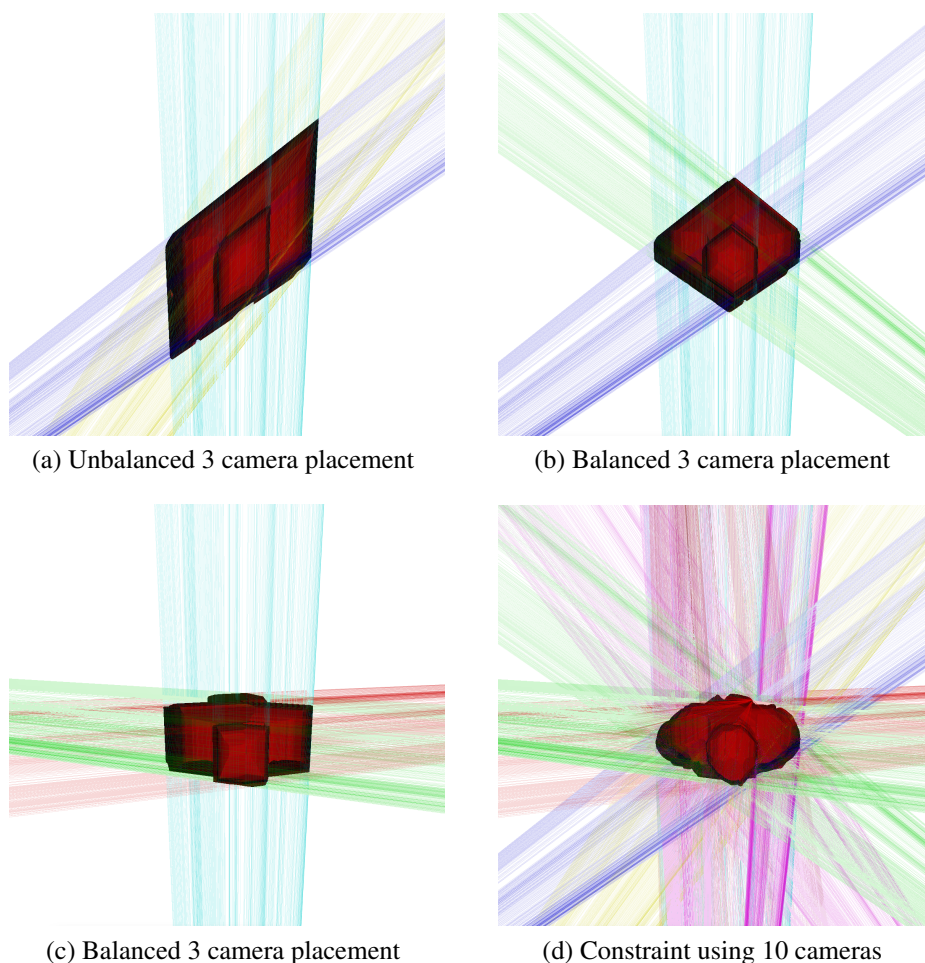


Figure 6.6: Camera placement. (a) Unbalanced placement of 3 cameras leading to reconstruction asymmetry, (b) Balanced placement of 3 cameras but angular reconstruction, (c) Balanced placement of 3 cameras without angular features (d) Better form constraint using 10 cameras.

camera count will minimise reconstruction asymmetry providing the angles between silhouette cones are not all acute. The reason for this is that the silhouette cone is an entity that diverges moving away from the camera along the principle ray. Consequently, the intersection of two silhouette cones arising from a pair of cameras whose principle rays subtend an angle of less than 90 degrees will give rise to an elongated intersection.

Figure 6.7 illustrates this principle - Pairs of 2D silhouette cones are seen inter-

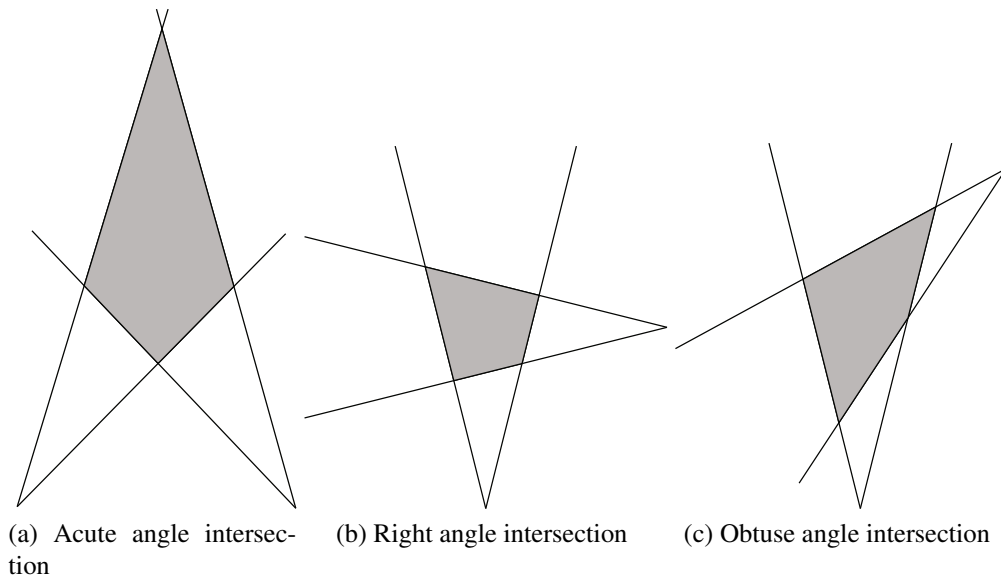


Figure 6.7: Pairwise silhouette cone intersection in 2D.

secting in three differing principle ray intersection angle scenarios: (a) acute, (b) right angle, and (c) obtuse angle. The area of the shaded intersection is a function of distance between the two silhouette cone origins and angle formed by the principle ray intersection. Notice that for acute and obtuse intersections the area increases as the principle rays approach parallelism, but that for obtuse intersections the intersection is always bounded by the other camera whereas acute intersections become infinite. Also, whilst distance between silhouette cone origins tends to be proportional to intersecting area for all three cases, as the acute intersection approaches parallelism the relationship becomes inverse: the intersection area increases as the distance between the cone origins decreases.

The optimal (most constraining) intersection is formed where principle rays intersect to form a right angle. Based on this pairwise 2D analysis it is quite clear that the principle translates into multiple cones and 3D; the example in Figure 6.6a follows the same principles. When the scene is reconstructed by cameras all pointing in the same direction spatial quality degrades significantly. The best results will be achieved when cameras are placed all the way around the object being reconstructed as this will result in greater cone intersection angles, which

give rise to more constrained silhouette cone intersections.

Now consider again the constraint formed in Figure 6.6b, whilst a symmetrical reconstruction is achieved, the spatial quality is clearly lower than that achieved in Figure 6.6d. Most notably, the front and back of the reconstructed human are poorly constrained, leading to angular features that are not present in the real human. The angular features arise through lack of any other constraint for the visual hull in this region. Figure 6.6c shows the reconstruction formed from 3 balanced cameras, but where two of the cameras from the previous balanced case have been replaced with a pair of cameras that are almost directly opposite each other. This results in a constraint that does not exhibit the angular features of Figure 6.6b because the two cameras now form an intersection that is closer to linear, which is a better constraint for the human body in this particular orientation. However, notice that the head has now become cuboid in shape, compared to a more regular but somewhat angular appearance of the previous balanced reconstruction. This is because the constraint offered by the 3 cameras does not provide any surfaces with which to model the curved nature of the head. Their arrangement gives rise to approximately right angular silhouette cone intersections which are only able to form roughly recti-linear reconstructions.

The general principle that can be drawn from this is that whilst cameras whose principle rays are perpendicular to one another will provide the best volumetric constraint of the reconstructed form, those with principle rays closer to parallel will provide the best surface intersections with which to model it.

Figure 6.6 (c) Demonstrates better form constraint by using 10 cameras surrounding the object being reconstructed.

6.3.2 Reconstruction of the face

The human face, and facial expressions are an important part of human non-verbal communication, and therefore it is important to understand what impacts upon

quality of the 3D reconstructed face.

It has just been shown that camera placement plays an important role in spatial quality of the reconstruction in its entirety, especially where there are a small number of cameras contributing to form reconstruction. However, the general rule derived: that cameras should be placed all the way around the object under reconstruction; does not tell us a great deal about the impact an individual camera can have, and tells us nothing about texturing.

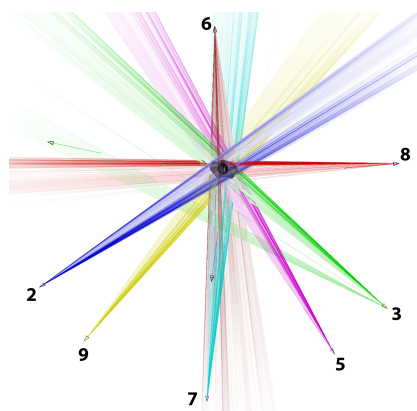


Figure 6.8: Camera arrangement used for the study.

The impact of individual cameras on the spatial, visio-spatial, and visual quality of a reconstructed human face is now investigated. A human is reconstructed from a set of 7 cameras, configured as shown in Figure 6.8: Camera 6 is situated behind the human being reconstructed and is always enabled for form reconstruction. Camera 7 is directly in front of the human. Cameras 2 and 9 are to the left of the diagram and capture the right hand side of the human. Cameras 5 and 3 are to the right of the diagram and capture the left hand side of the human. Camera 8 is at 90° to camera 7 and captures the left hand profile of the human.

The camera images are shown in Figure 6.9 to demonstrate that the intention of the study was to capture and reconstruct the entire human form, rather than just the face. In this way, the reconstructed facial quality can be investigated in a way that is meaningful in terms of a telepresence system in which participants are captured in their entirety.

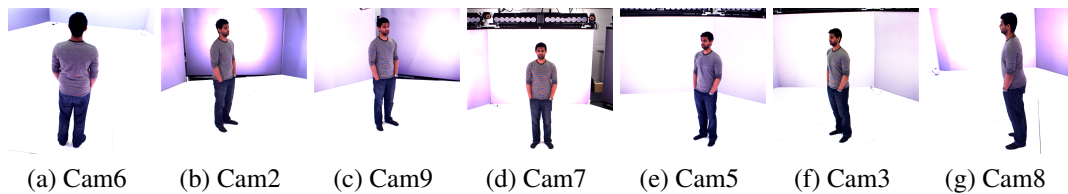


Figure 6.9: Camera images used in the study, all 1000 x 1000 pixels

In order to investigate the impact each camera has on spatial, visio-spatial and visual quality, the participant is reconstructed with one of the front, or side facing cameras removed each time. A series of figures follows that show the results. The figures are grouped in sets of four; each set illustrates the impact of a particular camera. The first figure in a set shows the cameras that contribute to the reconstruction; silhouette cones are drawn for each contributing camera. The second figure in a set shows the contribution of each camera to spatial quality; surface polygons are coloured according to the camera forming the silhouette cone they were derived from. The third figure in the set shows the visio-spatial quality; the spatial model has texture applied to it. The fourth figure in a set shows the impact on visual quality alone; the form is constructed from all cameras, but texturing is only derived from the active cameras in that set, coloured surface normals denote texturing camera for a polygon.

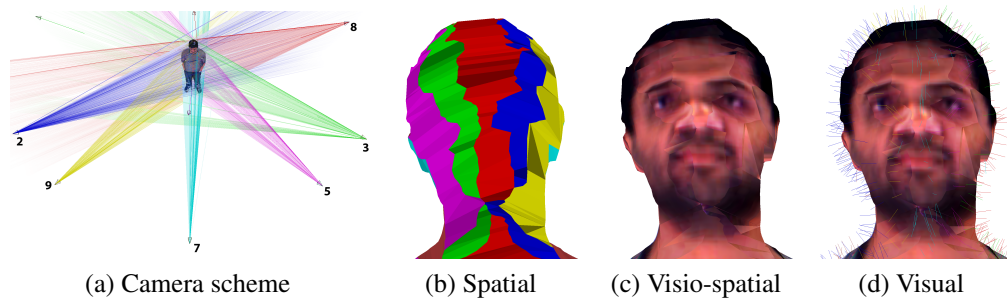


Figure 6.10: Reconstructed human face from 7 cameras [2,3,5,6,7,8,9] (a) Camera scheme used (b) Visual hull coloured by contributing cameras (c) Textured hull (d) Visual quality with surface normals showing texture camera source

Figure 6.10 shows the reference reconstruction in which all cameras contribute to form and texture. In the spatial sub-figure (6.10b) it can be seen how each

contributing camera's silhouette cone forms a strip of polygons oriented vertically down the face. Notice that the red polygons correspond to camera 8, that can be seen in Figure 6.9g to capture the participant from the left hand side. The silhouette cone formed from the image of camera 8 creates a strip of polygons that define the edge of the visual hull in the region defining by the edge of the participant from that camera's viewpoint. All cameras can be seen to form strips of polygons that are tangent to the object's true surface. Studying Figure 6.10d, it can be seen that the surface normals from the red polygons in the spatial figure are coloured cyan. This means that they are closer to parallel with camera 7's principle ray than any of the other cameras in the set. The same principle applies to the other visual hull strip and texture camera pairs: polygons formed by camera 5 are textured by camera 2, which is at approximately 90° to it, polygons formed by camera 3 are textured by camera 2, and so on. In other words, the camera providing the most direct view (assuming no occlusions) of a polygon will be the one situated closest to perpendicular to the surface of the silhouette cone formed at any one point.

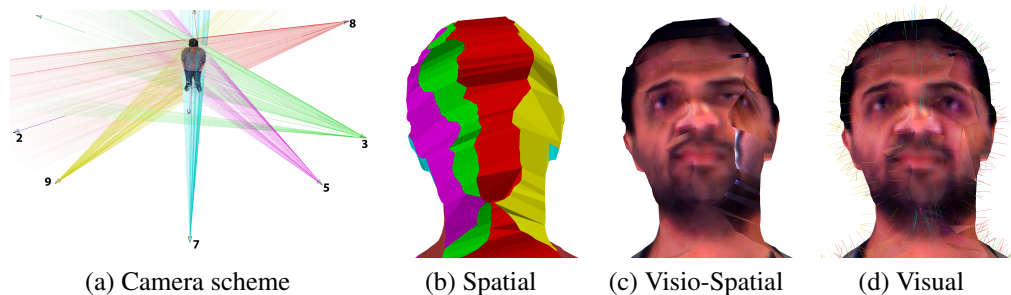


Figure 6.11: Reconstructed human face from 6 cameras [3,5,6,7,8,9] (a) Camera scheme used (b) Visual hull coloured by contributing cameras (c) Textured hull (d) Impact of camera 2 on texture provenance

Figure 6.11 shows the effect disabling camera 2 has on the reconstruction. Notice in Figure 6.11c that the participant's left cheek and eye have become badly deformed. From 6.11b and comparing to the reference reconstruction, it can be seen that the blue visual hull strip defining the left cheek, eye and forehead has disappeared. The polygons making up this region are now defined by the extension of silhouette cones emanating from cameras 8 and 9, and forms a prominent

ridge running down the seam of the visual hull between them. In terms of spatial quality, or model completeness, the presence of the ridge represents poor quality as the ridge is not present in the participant's face. The distorted texturing visible in this region is the by-product of poor spatial quality, as the camera images used are mapped to geometry that is non-existent in reality.



Figure 6.12: Enlargement of the intersection of silhouette and image from camera 3 in the region of an observed texturing artefact.

The texturing artefact visible next to the participant's mouth is a result of a background subtraction. The visual hull should be silhouette consistent, meaning that the reconstructed geometry will always reproject back to the interior of silhouettes from all cameras. The white pixels observed are clearly background pixels, so this would appear to violate the silhouette consistency constraint. Figure 6.12 shows an enlargement of this region in camera 3's image; the image has been cut out using the silhouette formed during background segmentation. Significant "ghosting" is visible in around the participant's head and face in the cut-out. The phenomenon arises through background segmentation errors, and since these pixels are within the silhouette itself this does not constitute a violation of silhouette consistency. Nevertheless, they are ugly, and degrade visual quality. The use of texture pixels close to the edge of the silhouette in any camera image is an indication that geometry is poorly constrained in that region.

Studying Figure 6.11d it can be seen that the source of polygon textures on the right of the participant's face has changed from the reference reconstruction. Camera 2 provided most of the texture for this region of the face previously, but it is mostly derived from camera 9 in this reconstruction. It is somewhat difficult to

notice any degradation in visual quality as a result, but looking closely around the participant's right eye in the reference reconstruction reveals a slight colour imbalance that creates an edge between the bottom of the eye and cheek bone. This edge is not present in Figure 6.11d.

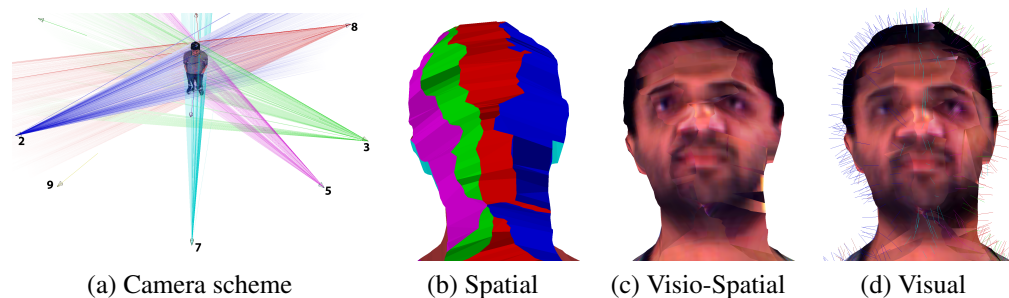


Figure 6.13: Reconstructed human face from 6 cameras [2,3,5,6,7,8] (a) Camera scheme used (b) Visual hull coloured by contributing cameras (c) Textured hull (d) Impact of camera 9 on texture provenance

In Figure 6.13, the effect disabling camera 9 has can be seen. The region defining the left most side of the participant's face, ear and neck was define by visual hull strips formed from camera 9 in the reference reconstruction, but in Figure 6.13b these can be seen to now be formed by camera 2. Because the region affected falls at the edge of the reconstruction from our viewpoint, it is not possible to state the full impact disabling camera 9 has, since there may be adversely affected geometry further around the reconstruction. However, it can be observed from Figure 6.13c that the geometry that is visible has changed sufficiently to cause a change in the contribution to texturing of the upper neck. In terms of the visual impact alone, Figure 6.13d shows that the majority of the right side of the face is now textured from camera 2. This causes no apparent deterioration in visual quality from this viewpoint.

Figure 6.14 shows the impact removing camera 7 has on the reconstruction. Comparing Figure 6.14b to the reference spatial reconstruction, the contribution of camera 7 at both ear lobes is no longer present. Because camera 7 is in the same lateral orientation as the viewpoint of these reconstructions, its contribution to visual hull polygon strips is almost invisible from this orientation, but the effect this

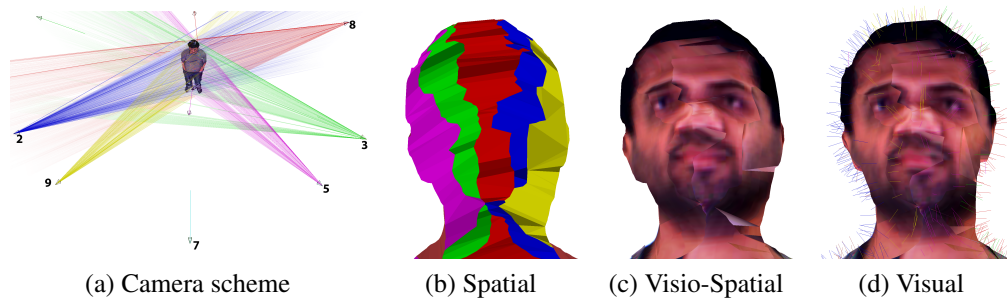


Figure 6.14: Reconstructed human face from 6 cameras [2,3,5,6,8,9] (a) Camera scheme used (b) Visual hull coloured by contributing cameras (c) Textured hull (d) Impact of camera 7 on texture provenance

has on the geometry of neighbouring strips can be significant. This is evident in the lack of constraint at the left side of the participant's neck, resulting in a general thickening in this region. The impact on visual quality in Figure 6.14d is also significant; the camera providing the majority of contribution for the front of the face has been removed. This results in a combination of camera 5 and 9 being used to texture the mouth, nose and forehead, which causes an apparent flattening of the nose.

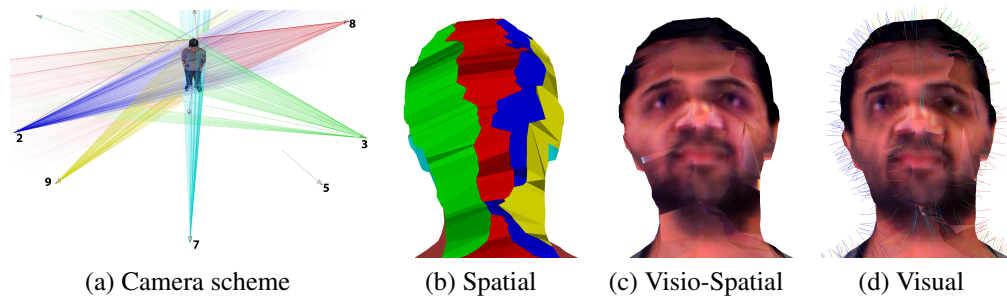


Figure 6.15: Reconstructed human face from 6 cameras [2,3,6,7,8,9] (a) Camera scheme used (b) Visual hull coloured by contributing cameras (c) Textured hull (d) Impact of camera 5 on texture provenance

Figures 6.15 and 6.16 show the impact of removing camera 5 and 3 respectively. The results cause the right hand side of the participant's face to suffer, but not as considerably as with camera 2 or 9. Camera 5 caused the right most side of the face to be better constrained, and its absence results in a facial asymmetry.

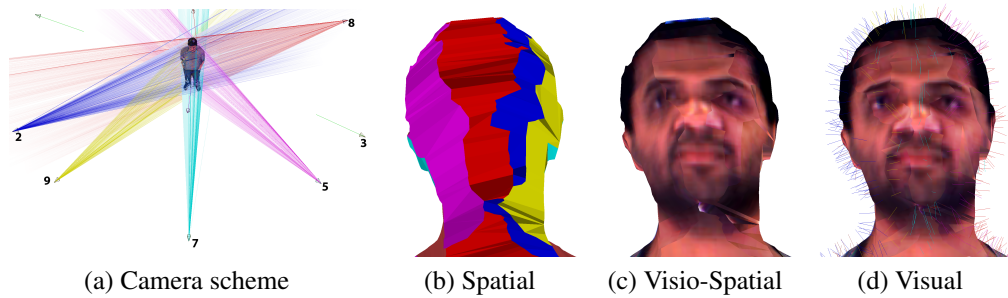


Figure 6.16: Reconstructed human face from 6 cameras [2,5,6,7,8,9] (a) Camera scheme used (b) Visual hull coloured by contributing cameras (c) Textured hull (d) Impact of camera 3 on texture provenance

Camera 3 appears to have little effect on the spatial quality from this viewpoint, but results in a visio-spatial degradation in Figure 6.16c. Since the appearance is worse than that in Figure 6.16d, the difference must be attributal to a deterioration in spatial quality.

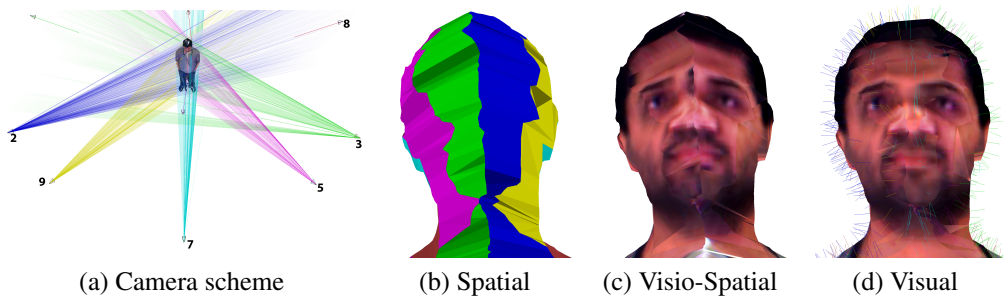


Figure 6.17: Reconstructed human face from 6 cameras [2,3,5,6,7,9] (a) Camera scheme used (b) Visual hull coloured by contributing cameras (c) Textured hull (d) Impact of camera 8 on texture provenance

In Figure 6.17, the effect of removing camera 8 can be observed. It is immediately clear from Figure 6.17b that the visual hull strip defining the front of the face has disappeared. Instead, the front of the reconstruction is formed from the locus of silhouette cones coming from cameras 2 and 3. This gives rise to an apparent ridge forming down the centre of the face. Viewed from above (Figure 6.18) the degree of spatial quality deterioration can be clearly seen. Due to the location of camera 8, and the lack of a symmetrical camera on the opposite side, or one from

above, it is the only camera providing constraint of the participant's profile.

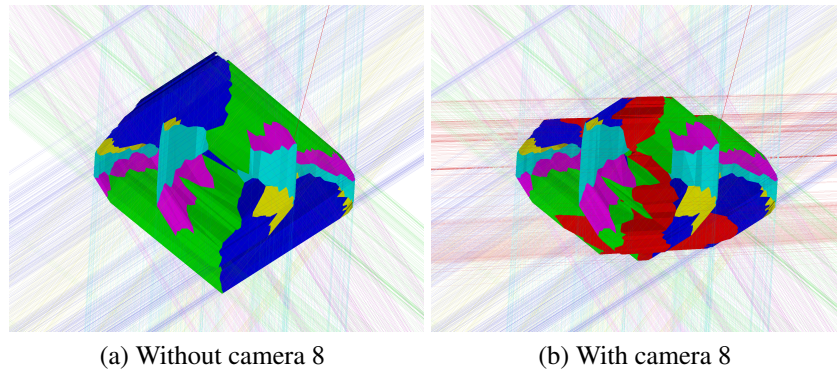


Figure 6.18: Reconstruction with and without camera 8, viewed from above

6.3.3 Discussion

This study investigated the impact of camera placement on quality of reconstruction in two ways: firstly the impact of different angular configurations of camera sets on spatial constraint of the entire form was investigated, secondly the impact of individual cameras on the spatial and visual quality of a human face was studied.

The main findings were:

Whilst close to perpendicular camera configurations provide the tightest constraint on overall form of the visual hull, they result in insufficient variation in angular intersection of silhouette cones to model curved or flat features. Placing cameras opposite each other will provide close to parallel silhouette cone surfaces that are better able to model curved surfaces.

The impact of individual cameras on visual, visio-spatial and spatial quality is most apparent in visio-spatial quality. spatial deterioration in the reconstruction often led to texturing artefacts that would be disconcerting in a telepresence context. On the whole disabling a single camera for texture contribution did not have

a significant impact on visual quality when the form was generated from all cameras.

6.4 Case study 2: Investigating Eye Gaze

6.4.1 Introduction

In this study the communication of eye gaze through 3D reconstruction is measured through a collaborative experiment. As shown by the previous study, with shape-from-silhouette, the polygonal qualities of the reconstructed head, and visual quality of face and eyes, depend on the arrangement of cameras used to generate form and texture. The position of eyes within the head affect gaze direction, but little is known about the impact of camera placement or indeed 3D reconstruction on gaze estimation of gaze.

Contribution

We provide the first empirical evidence of the reliability of gaze reproduction in 3D reconstructed virtual humans. Reliability of gaze estimation is measured across various gaze poses and camera arrangements, likely in every day social interactions and settings. This is the first work to provide evidence that social eye gaze can be reliably communicated through a virtual 3D reconstructed medium. Furthermore, it provides the first insight into the importance of the relationship between turn of eyes, head and body, method of recreating and texturing 3D form, and arrangement of capturing cameras.

6.4.2 Method

The aim of the experiment was to determine if eye-gaze can be estimated from a 3D reconstructed human to within the accuracies necessary for social interaction; and reliably across various gaze poses and camera arrangements.

Capture of the subject

The subject was captured by 10 cameras in 7 different gaze poses. Capture took place within an octagonal CAVE-like display system designed for telepresence research. During capture, defuse/ambient light from the immersive displays fourteen projectors (6 ceiling and 8 wall) was combined with defuse spotlights to achieve clear lighting and contrast of the face. Defuse spot lighting was bounced off the floor in front of the captured subject in order to remove much of the shadow around the eyes, without causing glare in their face and reflected highlights in the pupils. Spotlights were also placed behind the screens to increase ambient light. The rooms strip lighting was turned on. Five different arrangements of texture cameras were used. Each was a subset of the ten cameras from which silhouettes from images were used to create form.

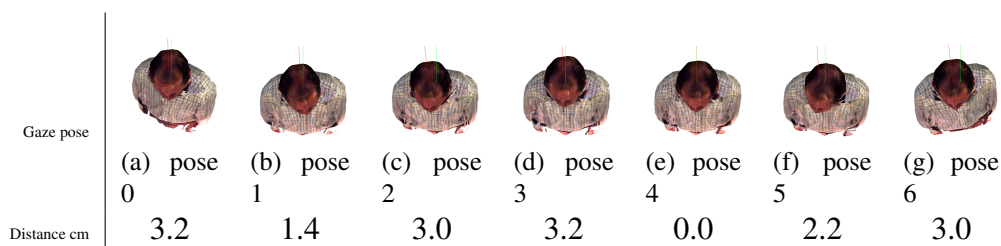


Figure 6.19: Determining the rotation axis offset between poses. The green line is the fixed axis of rotation used in real trials. The red line is the adjusted rotation axis.

During capture, the subject stood on a mark on the floor on which he theoretically remained centred across the gaze poses; this mark would later form the axis of rotation about which the participants rotate the reconstructed subject. In practice,

the exact position of the subject being captured varied slightly between gaze poses. This was corrected in software afterwards by analysing the 3D reconstructions of the subject in each different pose and determining the rotation axis offset. Figure 6.19 shows the calculated distance offsets, and original and adjusted rotation axes.

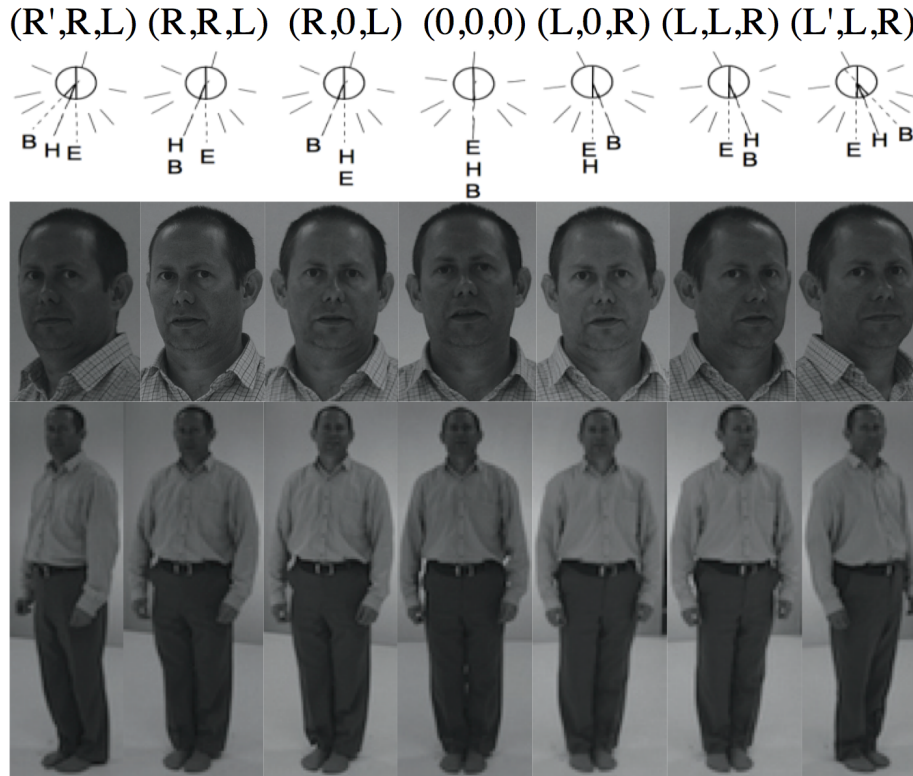


Figure 6.20: The seven gaze poses, each shown diagrammatically above photographs of head and whole body from the gaze target. Normal of body head and eyes (B H E) are shown in relation to the cameras.

The seven gaze poses adopted by the subject represented all possible combinations of eyes, head and body; being either centred or turned, Figure 6.20. The subject stood on a mark on the floor 4 metres away from three gaze targets; one 18° to the left, one directly ahead and one 18° to the right. The gaze angle was chosen at this distance as it was one that appeared natural, was comfortable for the observer, and at the whites of the eye were barely discernible on both sides. Each gaze pose can be described both through a triple of body, head and eye orientation and diagrammatically showing the normal of each with respect to the full set of

cameras. The ordered triple describes (rotation of body relative to the eyes, rotation of head relative to the eyes, rotation of eyes relative to the centre line). For example (R,0,L) indicates that the body is rotated to the right from the eyes, that the head and eyes are in line, and that the eyes are rotated to the left. Note that the first two values thereby give the relative arrangement of the body, head and eyes within the target participant and the third value gives the spatial orientation of the assembly as whole. L and R denote a rotation of 18° to the left and right respectively and L' and R' denote rotations of 36° . Left and right are those of the captured subject rather than an observer.

Participant method

22 participants were each presented with a series of life size static 3D reconstructions of a previously captured human subject and asked to rotate the viewpoint around each until they felt most looked at. Each participant stood on a mark on the floor 2 metres away from a 4 metre by 2 metre projected display screen. The height of each participant's eyes was measured at the start of the experiment, and entered into the reconstruction software to calculate the correct viewing frustrum for that participant on the display screen, and to render the subject's eyes at the same level as the participant's. The subject was rendered on the projection screen as if they were 3.5 metres away from the participant.

Each participant was asked to rotate 3D reconstructions of the subject until they felt the subject was looking directly at them. Rotation was achieved using the cursor keys on a wireless keyboard in increments of $0.25/\text{degree}$, and the subject pressed space when they settled upon feeling closest to making eye contact. The seven different gaze poses were reconstructed using five different texturing camera configurations, forming 35 different combinations, for which the presentation order was randomised. To simplify the experiment, and in particular the analysis, and to maximise repeatability, static reconstructions were used. To demonstrate results were valid for not only image based reconstruction but also VBR, the virtual human was kept still by reusing a single frame from each camera at an

interactive frame rate.

The accuracy with which participants could orient the 3D reconstructed human such that they felt it was looking directly at them was measured in degrees. If the participant oriented their view directly along the line between the captured subject and his gaze target, the error in accuracy would be 0° .

Cameras used for texturing

To test the impact of texturing on gaze estimation the subset of cameras from which images were used for texturing was varied.

Table 6.1: Camera arrangements

Name	Cameras	Description
Shallow	1	Single front 15° V to face
Pair	2,8	38° , 342° H, 15° V to face
Arc	1,2,3,7,8	38° - 342° H
Surround	0-9	Around and above
Steep	9	Single front 30° V to face

To evaluate the impact of reducing texture cameras we compared camera arrangements of Single, Pair, Arc and Surround; with one, two, five and ten cameras respectively. To evaluate the impact of steepness of texture camera to face we compared two single frontal cameras, both just above 2m from the floor with one about 2m and the other 4m in front of the subject. The five camera arrangements are described in table 6.1. Each gaze pose was captured in ten synchronised frames, each from a different camera facing the target participant from a unique angle. Ten silhouettes were created, one from each image. The form was created using all ten silhouettes and textured from various subsets of the ten images. The placement of cameras with respect to the captured subject and display walls is shown in figure 6.21. Example images and corresponding silhouettes are shown in Figure 6.22.

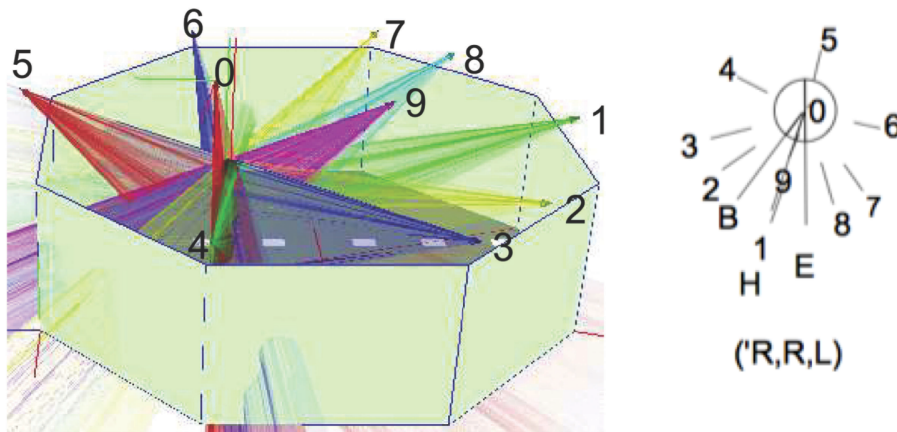


Figure 6.21: Camera placement with silhouette cones to capture subject (left) and schematic of cameras with respect to a given gaze pose (right).

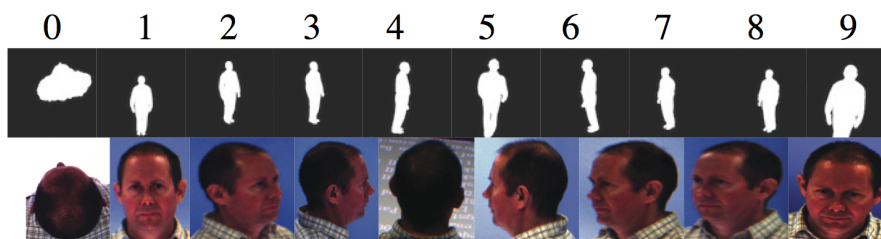


Figure 6.22: Close-ups of silhouettes and head from images, for gaze pose (0,0,0) where head, eyes and body face front.

Method of analysis

We use three metrics for significance: statistical; practical; and qualitative. In the quantitative analysis we investigate statistical and practical significance, whereas significant corroborating evidence is investigated in the qualitative analysis.

Accuracy of gaze estimation in terms median and interquartile range of absolute errors is shown in box plots. Where statistical significance lies, it is described in the text. In the box plots: horizontal line marks 4° accuracy; upper and lower box, respective quartile; line in box, mid quartile; whiskers, min and max; numbered dots and circles, outliers.

Statistical significance is measured using the Wilcoxon Signed Rank Test. We take: p values of less than .05 to indicate significance. Where data is aggregated for an overall view, for example combining all gaze poses where eyes are centred and all that are turned, or combining all camera arrangements, we report median of medians.

Practical significance compares accuracy of gaze estimation to that typical of social gaze in the real world. Our test subjects were stood 3.5m from the virtual humans. 4m is the extent of social gaze distance and we have chosen 3.5m as it falls comfortably but not excessively within this. From this distance, centre head to end of shoulder are about 4° apart. Thus we argue 4° is the critical accuracy necessary for two people stood shoulder to shoulder to determine which is being looked at from this distance. For this experiment we consider practical significance to be the movement by more than 1° of one of the quartiles or median error in estimation that takes it across the 4° threshold.

6.4.3 Results and analysis

Impact of gaze pose on gaze estimation

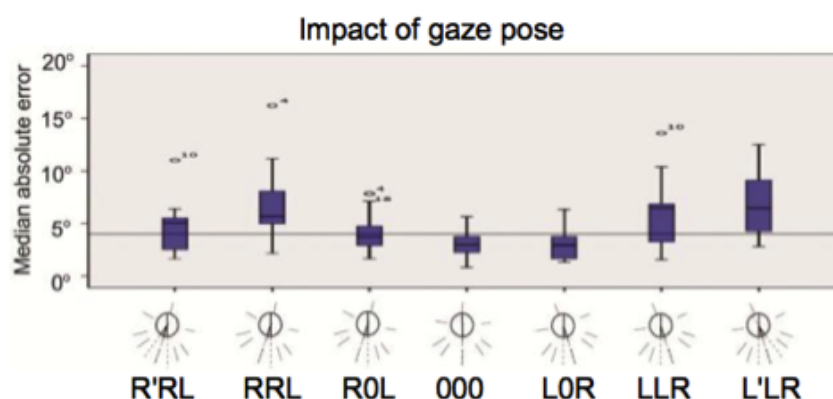


Figure 6.23: Taken across all camera arrangements, median of median estimations always and only below 4° when eyes centred (x0x).

Figure 6.23 shows the results for the impact the subject's gaze pose had on the participant's ability to estimate his gaze. The best accuracy is achieved for (000), where the eyes, head and body are all facing forwards. Whether or not the head and eyes are turned relative to each other results in a significant difference. Poses with the eyes centred (L0R and R0L) result in an estimation error with median 3.82° and inter-quartile-range of 1.68° . Whereas those with eyes turned (L'LR, LLR, R'RL, and RRL) result in median estimation error of 5.9288° with inter-quartile-range 2.67° . The relative orientation of head and body did not make a statistically significant difference. Head and body aligned (LLR and RRL) resulted in median estimation error of 5.45° , inter-quartile-range 2.67° , showing no significant difference with head turned (LLR and RRL) (Median 5.87° IQR 4.89°).

Therefore, it is clear that gaze poses in which the eyes are centred in the head are the only ones for which gaze could be accurately estimated at the 4° criterion of acceptable error.

Impact of texturing cameras on gaze estimation

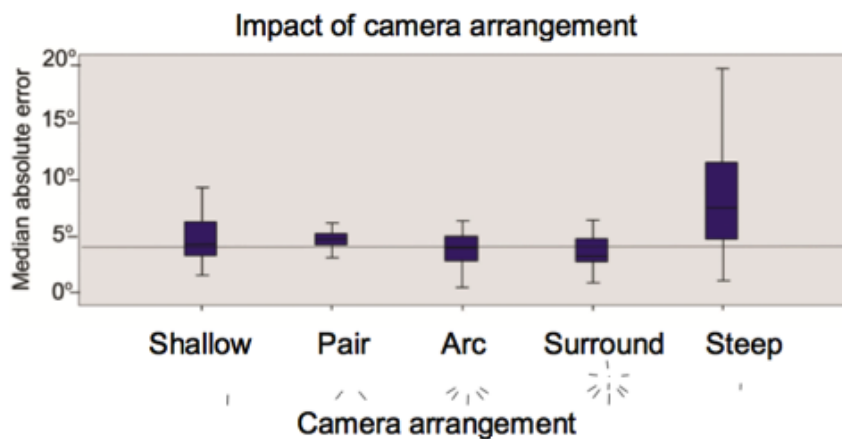


Figure 6.24: Over all gaze poses a median accuracy of 4° is achieved by surround and arc with both shallow single and pair within 1° of this limit. Steep single stands out as clearly worst performer.

Figure 6.24 shows the impact of texture camera arrangement on accuracy of gaze

estimation. The graph shows the error in estimation averaged across all gaze poses. An accuracy meeting the 4° criterion is achieved by the "surround" and "arc" configurations, with "shallow" and "pair" falling just above. The "steep" configuration clearly resulted in the worst gaze estimation.

Impact of number of texturing cameras on estimation across gaze poses

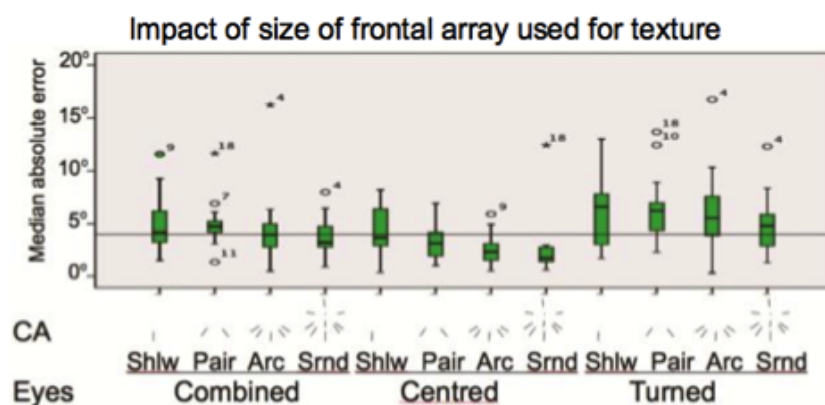


Figure 6.25: Median accuracy increases with number of texturing cameras both when eyes centred and turned, being within 4° when centred.

Figure 6.25 shows the impact the various camera texturing configurations had on gaze estimation categorised according to different eye directions: combined is the combination of poses, centred is for poses in which the eyes are centred in the head, and turned are poses in which the eyes were turned relative to the head.

For combined eyes centred and turned poses, texturing from surround and arc configurations significantly outperforms both pair and shallow configurations. For eyes centred poses only, surround, arc and pair configurations significantly outperform shallow. For eyes turned poses, none of the camera configurations resulted in estimation at the 4° criterion. Overall, it can be summarised that the arc and surround camera configurations resulted in the best gaze estimation.

Impact of steepness of texturing camera across gaze poses

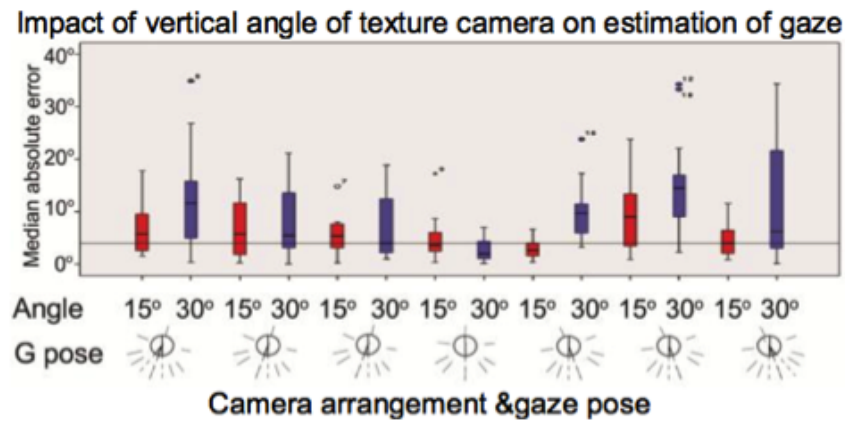


Figure 6.26: Estimations when texture camera close to eye level outperform those when a texture camera is looking down at a steeper angle. At worst when eyes turned and camera angle steeper.

Figure 6.26 compares the single camera texturing configurations "shallow" and "steep" across the different gaze poses. For gaze poses in which the eyes are turned in the head, shallow outperforms steep, but for eyes centred poses, the perform similarly. Overall, "shallow" approached the 4° criterion, whereas "steep" did not. Using "shallow" rather than "steep" resulted in statistically significantly better performance overall, Shallow (Median= 4.19° IQR= 3.13°) outperformed steep (Median= 7.45° IQR= 7.34° , $Z=2.156$, $p=.031$).

Qualitative analysis

Figure 6.27 shows close ups of headshots of all reconstructions taken from the target of gaze. We now look at why the impact on subset of cameras was not significant when eyes are turned. Reconstructions of gazes with eyes turned right suffered stretching of the dark of the rightmost eye, considerably greater than that seen when eyes were centred.

Closer inspection of images from source cameras and of reconstruction showed

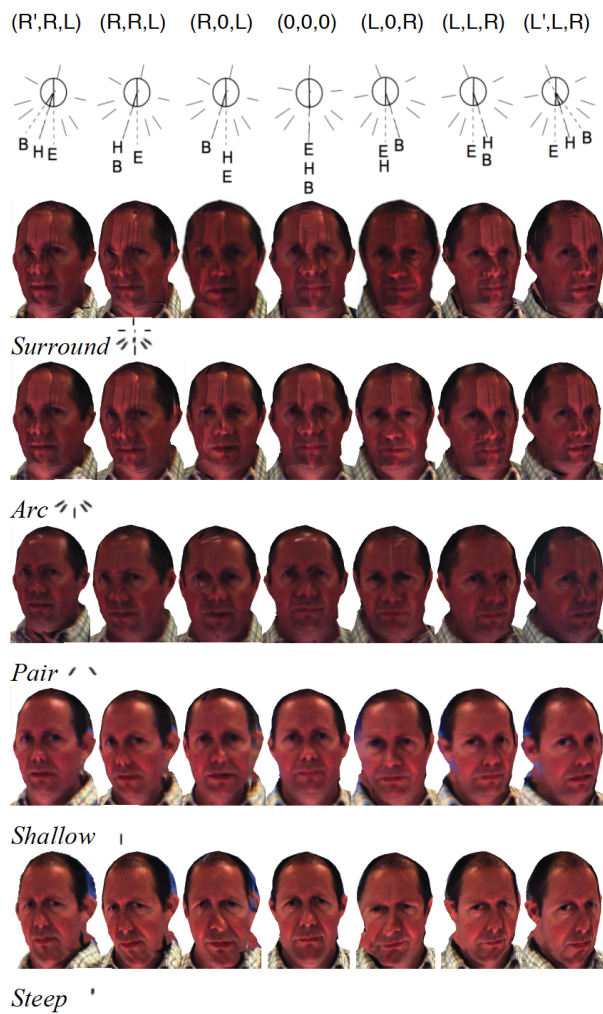


Figure 6.27: Close up of head from target of gaze. Ranked from top to bottom in order of accuracy of gaze estimation across all poses.

the cause to be a combination of shadow in the corner of the eye and insufficient camera resolution to clearly distinguish this from the dark and white of the eye, Figure 6.29. During analysis we reproduced a similar result through blurring in an image editor, Figure 6.29. Interestingly, the problem was most apparent for the single central camera.

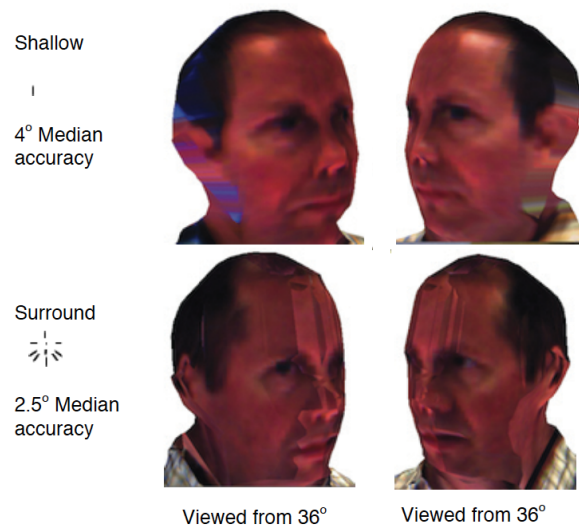


Figure 6.28: Rotating past the gaze target exaggerates the differences between surround and single texture. The dark of the eye appears stretched when taken from a single texture but not when taken from a texture from the camera close to this viewpoint. (0,0,0).

Steepness of texture camera to face

Texturing from a camera angled steeply to the subject's face caused a significant drooping effect to the eyes and an apparent twist to the end of the nose. The bridge of the nose, the inner corners of the eyes and the top of the mouth are not drooped, whereas the droop of the eye increases gradually toward the outer corners, and drooping appears at the end of the nose and under the chin. In all cases the cause appears to be texturing beneath an overhanging polygon that should have occluded the camera. The reason eye droop gradually increases from nothing at the bridge of the nose, is from shape-from-silhouette's inability to capture the concavity of the eye socket. When viewed from the steepness of the texture camera, the reconstruction looks reasonably correct, Figure 6.30.

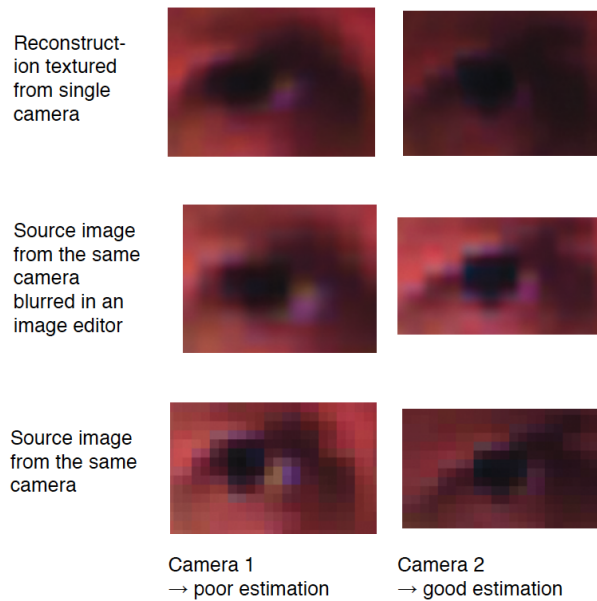


Figure 6.29: Close up of right eye, Reconstructed, blurred through an image editor, and source. The image used for texture from one camera but not the other appears to stretch the dark of the eye. The underlying cause seems to be insufficient camera resolution to resolve the dark of the eyes from shadow when image blurred by texturing. The effect of texturing is similar to that of blurring in an image editor.

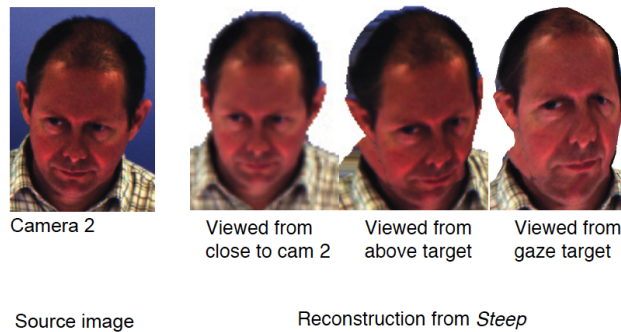


Figure 6.30: Reconstruction textured from camera looking down steeply on face looks reasonable from a similar vertical angle but distorts as viewpoint lowered to eye level. (L,L,R).

6.4.4 Discussion

Interestingly, and perhaps conversely to expectation, reconstructions textured using a surround set of texturing cameras were found to improve gaze estimation compared with those textured with a single camera. This was tested with both steep (distorted) and shallow angles of inclination. It had been expected that texturing artefacts arising from difference in the relative brightness would result in poorer gaze estimation than those textured with a single camera. However, this was clearly not the case as shallow performed worse than all other configurations except steep for gaze poses with the eyes centred in the head.

The experiment showed that for gazes poses in which the eyes were centred in the head, participants were able to estimate the subject's gaze to within the 4° criterion. For gaze poses in which the eyes were turned in the head this was not found to be the case, but estimation was significantly improved if the reconstructed human was textured from a surround set of cameras.

6.5 Conclusions

In this chapter a research platform, 3DRecon, was presented that aimed to facilitate analysis of the impact of camera placement on spatial and visual quality of 3D reconstruction. The objective was to develop a platform through which this impact could be investigated, and in order to do so the requirements were determined as follows:

Firstly the platform needed to be able to provide an interactive setting within which users could reconstruct frames from pre-recorded datasets and manipulate the result to analyse areas of the reconstruction in closer detail. In order to investigate the impact of camera placement it was essential that control over the contribution of individual cameras for form and texture genesis be provided. The ability to draw camera silhouette cones on a per-camera basis greatly provided great en-

hancement to the understanding of shape-from-silhouette based algorithms. To separate the analysis of spatial versus visual quality, it was also necessary to provide a means by which each may be studied independently. To meet this end rendering of the visual hull without any texturing was implemented, and further to this, the visual hull could be drawn coloured by which camera's silhouette cone had contributed to surface polygons. In order to study contribution of cameras to visual quality, the provision of coloured surface normals enabled the provenance of texture for a particular polygon to be determined. These were the primary requirements of the system in terms of meeting the research objectives.

The platform also provided a number of other useful features, most notably the simulated setting. Whilst this was not a primary requirement as it was not necessary for the investigation of spatial and visual quality, it provided a means by which algorithms and camera placements could be prototyped and evaluated without the hindrance of real-world camera problems. This greatly accelerated both the ability to investigate possible camera placement, and investigations into their impact on spatial and visual qualities.

The efficacy of the implemented platform was demonstrated by means of two case studies. The first demonstrated how the aforementioned features could be used to investigate the impact of camera placement on both spatial and visual quality in the reconstruction of a human face. The second demonstrated that the platform has been used as the core tool in both the conducting of an experiment and subsequent analysis of results in a collaborative experiment investigating the conveyance of eye-gaze through 3D reconstructed humans.

Further to the investigation of visual and spatial quality in 3D reconstruction. In the context of a real-time telepresence system, 3DRecon allows for the connection of live cameras which can be used to reconstruct humans in real-time. In this sense, the platform also provides the ability to investigate the impact of cameras on temporal quality, although this has not been demonstrated here.

Chapter 7

Discussion and Conclusions

This thesis concludes with a discussion of the research carried out: The setting, aim, objectives and research questions are revisited, followed by a retrospective review of how the methodology employed has shaped the research. The hypotheses are then reviewed in terms of the outcomes from experiments that were used to prove or disprove them. Shortcomings of the research are identified, followed by the formation of conclusions, leading to some ideas for future research directions.

7.1 Discussion

7.1.1 Aim and point of departure

The aim of this research was to study and improve the balance of temporal, spatial and visual quality of virtual humans, reconstructed from multiple video streams, to support most kinds of non-verbal communication in telepresence.

The research followed on from previous work on eye-gaze in immersive collaborative virtual environments [102], which enabled participants to see what fellow participants were looking at, but not what they looked like. In that research, a user's head and hand were motion tracked, and the direction of their eye gaze captured using an eye tracker. Local and remote participants were depicted in a shared virtual environment using avatars whose eye direction was controlled in real-time by the eye-tracker, but whose facial features were otherwise static. Whilst this research proved that eye-gaze could be conveyed through ICVEs, the obvious shortcoming of it was that the avatars did not look or move like the real participants. A natural progression from this research was to attempt replace the avatar with a lifelike real-time virtual copy of the participant, described by the term "virtuality human" in [104].

7.1.2 Review of objectives and research questions

The objectives and research questions fall into three categories:

Exploratory

O1: Determine the requirements and approaches to 3D reconstruction suitable for a real-time telepresence system.

Q1: Which approaches to 3D reconstruction can provide the best balance between performance and model quality for representing the dynamic human form in the context of a telepresence system?

These aim to identify through literature review, prototyping and evaluation, 3D reconstruction methods suitable for the purpose of real-time reconstruction of moving humans; whilst isolating the quality measures and relationships between them through which these may be compared.

Evolutionary

O2: Determine whether algorithms can be improved upon to achieve higher temporal or spatial/visual quality.

Q2: How can algorithms or their implementation be improved upon to achieve higher qualities in real-time?

These aim to focus on improving performance of a 3D reconstruction algorithm. Given the quality relationships determined through the exploratory research it is believed that this will enable higher quality 3D reconstructions in real-time.

Evaluative

O3: Develop a platform through which the impact of camera placement on spatial and visual quality of 3D reconstruction can be studied.

Q3: What are the requirements of a system with which the impact of camera placement on spatial and visual quality can be studied?

These aim to set the requirements for, and develop a system enabling the impact of camera placement on visual and spatial qualities to be investigated.

7.1.3 Review of methodology

The methodology described in Chapter 2 was applied to meet objectives, answer research questions and prove or disprove hypotheses. The specific research process applied was described in Section 2.3, and here we review how this process contributed to the research.

7.1.4 How literature guided the research

Initial literature review developed a body of knowledge that provided essential background information on the subject of machine vision, multiple view geometry and 3D reconstruction [81], [27], [28], [49], [38]. Surveys of 3D reconstruction methods [101], [15], [88] provided a useful review of the range of methods available. Research papers detailing specific algorithms [64], [36], [11], [21], often presented algorithm performance timings, which provided a broad indication of methods approaching real-time capability, or presented an indication of visual or spatial quality. In these ways the literature contributed to the exploratory objective phase of research

As research evolved to focus on improving temporal quality as a means to achieving real-time 3D reconstruction at sufficient spatial and visual qualities, literature provided details on parallelisation of existing techniques in two ways: distributed over a network [126], [13], [72], and processed locally using multi-core GPUs [65], [130], [50], [112].

The EPVH algorithm [39] had been encountered, and an efficient implementation towards real-time processing described in [40]. The authors described a network distributed approach aimed at real-time applications in [41]. Other than this publication a gap in the literature was identified in the area of parallelising surface based 3D reconstruction methods; and further to this there appeared to be no research at all into using local multi-core processing such as GPUs and CPUs to achieve it.

7.1.5 Contributions of the research to literature

Research contributed to literature through a combination of collaborative and individual research:

Collaborative contributions

- [103] provided latency measurements comparing existing video conferencing and immersive collaborative virtual environments, giving the background in terms of latency expectation for future communication across a distance technologies.
 - The method for measuring latency was devised by this research, and carried out for both video conferencing and ICVEs.
- [82] and [83] Contributed the first results of removing camera image synchronisation for shape-from-silhouette algorithms, and provided an analysis of the suitability of using a software capture trigger in terms of how this might affect camera image synchronisation.
 - 3D reconstruction algorithms prototyped and evaluated in this thesis were used to present output models from the unsynchronised camera images.
- [104] Showed for the first time that eye gaze could be conveyed to accuracies sufficient for human social interaction.
 - The research platform that was developed (Chapter 6) was used to carry out the experiments, take and refine accurate measurements, and develop a better understanding of how camera placement contributes to form and appearance of reconstructed humans.

Individual contributions

- [32] presented an initial parallelisation of parts of the EPVH algorithm, providing the first measurements of EPVH performance executed on a GPU.
- [33] furthered previous collaborative study of the effect that removing camera image synchronisation has on 3D reconstructions of moving humans. Unlike earlier work, which was simulation based and presented only the effects on spatial quality, this research used datasets of real humans and presented fully textured output models.
- [34] presented an end-to-end parallelisation of the EPVH algorithm tailored for processing on local multi-core CPUs or GPUs. This provided a different approach to parallelisation than the network distributed method, and also compared execution between GPUs and CPUs.
- [35] introduced the research platform that was developed, which unlike other similar utilities, included the reconstruction back-end enabling analysis of both form and texture genesis from different cameras. This also enabled a simulated setting, which could be used to prototype camera placement, potentially saving a considerable amount of time.

7.1.6 Prototyping and evaluation

The process of prototyping algorithms described in 2.3.5 provided an essential means by which methods derived from the literature survey could be implemented and evaluated. Development of a simulated setting (2.3.5), in which implemented algorithms could be tested without the hindrance of real-world phenomena, (such as camera calibration, camera image noise, background segmentation, etc) provided the key component necessary for rapid iterative prototyping, development and refinement. The steps up to and including comparative study (2.3.5) enabled investigation of the key aspects of **O1**, and provided sufficient answers for **Q1** through pilot experiments **E0** (2.4), for progression of the research.

During optimisation by parallelising of the EPVH algorithm the simulated setting and comparative study enabled rapid testing of implementations. Use of publicly available datasets enabled testing of non synthetic humans, captured by real cameras, without the need to go through the processes involved in setting up and calibrating a camera system. Together with the simulation, this enabled the parallel EPVH implementation to be debugged, optimised, refined and evaluated in a continuous development cycle, employing simulation or real-world data as appropriate. This approach, and the evolution of the parallelisation of EPVH for local processing through experiments **E3** and **E6** (2.4), helped meet objective **O2** and answer question **Q2**.

Development of a platform for further research

Building on the evaluation and visualisation methods implemented during the course of research, a platform was developed that enabled interactive control of the inputs to the 3D reconstruction algorithm. This enabled camera images to be loaded from disk, or streamed from live cameras and selected in real-time for contribution to form and texture genesis, thus providing a means of evaluating their impact on visual and spatial quality. The ability to use live camera images extended upon the use of the simulated setting, or pre-captured datasets enabling laboratory experiments (2.3.5) which could be used to investigate human non-verbal communication through the medium of 3D reconstruction. Whilst conducting experiment **E1** provided some of the expectations of a communication system, determining the requirements for and developing this platform met the objectives of **O3**, and conducting collaborative experiment **E7** (2.4) proved that the platform was an effective means of performing such investigations.

7.2 Review of hypotheses

H1: Approaches to 3D modelling from images can be used to model moving humans at sufficient qualities for telepresence applications.

By parallelising the EPVH algorithm we were able to achieve real-time performance (20 - 30 frames per second) using 10 cameras of similar resolution to those used in typical desktop video conferencing. One key difference is that in desktop video conferencing the camera tends to frame the upper torso and head in the camera image, whereas in our 3D reconstructions we captured the entire human body in the camera image. This difference results in fewer pixels representing the facial region, and therefore the possibility of insufficient detail being represented in comparison to video conferencing. However, by conducting collaborative experiment **E7** we proved that eye gaze could be conveyed at accuracies sufficient to support human social interaction. This is important because eyes are small in comparison to other features, and are therefore perhaps the most challenging to represent.

H1 was proved to be clearly true, that modelling from images can be used to model moving humans at sufficient qualities for telepresence applications.

H2: Relaxing constraints on temporal consistency of inputs will result in a degradation of spatial and visual quality in the case where the subject is moving.

Through experiments **E2** and **E5** we tested what the effect of removing hardware camera synchronisation would be on reconstruction of moving humans. We found that for whole frame periods of de-synchronisation of a subset of cameras, moving parts of the human anatomy would become truncated. This was particularly true for extremities at the ends of limbs, which are the most likely areas to move considerably between cameras frames. Spatial quality was most adversely affected, and as a consequence visual faithfulness too, but specific visual quality degradation was also noticed in terms of texturing artefacts that arose as a result of inconsistencies between camera images.

H2 was shown to be clearly true, that relaxing constraints on temporal consistency of camera images results in a degradation of spatial and visual quality when the subject is moving.

H3: Temporal quality can be improved upon through parallel processing using multi-core GPUs or CPUs. Such an improvement will allow for higher spatial and visual quality in real-time.

Temporal quality was improved through parallelisation of the EPVH algorithm, enabling real-time performance using 10 cameras at a resolution of 1000 x 1000 pixels, when processed on a modern 4 core multi-core CPU supporting hyper-threading. The parallelisation was shown to continue to tend towards optimal as the input complexity increased, meaning that for higher resolutions or number of cameras used the performance enhancement tended towards the number of parallel threads executed. Given that spatial and visual qualities are a function of input complexity for the general case, it follows that if temporal quality is improved through the parallelisation, then the spatial and visual qualities achievable in real-time will also be improved.

H3 was shown to be partially true, that temporal quality can be improved through parallel processing on multi-core CPUs, allowing for higher spatial and visual qualities in real-time. The hypothesis was not, however, shown to be true for execution of the parallelised EPVH algorithm on GPUs.

7.2.1 Shortcomings of the research

Whilst the overall aim of the research has been met, objectives realised, and research questions answered, the directions taken have led research down a certain path that has resulted in a lack of investigation in certain areas. These are acknowledged and listed here.

Simplistic approach to texturing

The main shortcoming in terms of the quality of output is probably the simplistic approach to texturing employed, leading to poor visual quality manifested as texturing errors due to occlusion, and camera exposure or colour balance differences. The texturing approach used does not take into account the occlusion of regions in the process of selecting the best camera for texturing a particular polygon. Whilst this was implemented, a solution for real-time performance that was sufficiently computationally inexpensive was not achieved. Other researchers have used processes such as blending together the textures from the three best candidate cameras [40] to reduce such errors. This technique will reduce the impact that occlusion, camera exposure, or colour balance has on visual quality, but not eliminate it entirely since the occluded camera will still feature in the blended texture. The reliable method for eliminating occlusion artefacts involves calculating the visibility of polygons in each camera image, which can result in partial visibility for partially occluded polygons. In this case polygons are split into visible regions for a particular camera. Once the visible polygons for each camera have been determined, the optimal camera can be selected based on surface normal and size in much the same way as in our implementation.

Lack of end-to-end latency measurements

Whilst the end-to-end latency of video conferencing and ICVE systems were measured through experiment **E1** early in the research, the corresponding latency of an end-to-end 3D reconstruction system, to provide a means of comparison, was never measured. The measurements for VC and ICVE latency were conducted over a local network, and could be repeated for the 3D telepresence implementation, however due to their nature they are time consuming and awkward to conduct.

Many challenges still need to be solved for a complete end-to-end system operating over anything but a local network, including sufficient compression of mesh

geometry and textures to enable transmission over an internet bandwidth connection.

7.3 Conclusion

Research began by surveying existing methods for 3D reconstruction using the quality expectation of current communication systems such as video conferencing and immersive collaborative virtual environments to guide direction. Achieving sufficient quality of human reconstruction quickly enough to be used as a communication medium is a balancing act of performance versus quality, and whilst many techniques are documented in the literature, they are generally too slow or too crude in terms of quality, to be used for non-verbal communication purposes. Other researchers had tackled this problem by using heavy weight network distributed processing to reconstruct humans with sufficient performance and at a high enough quality for real-time interactions. However, such an approach does not provide an obvious path for the wide scale adoption of the technology.

With this in mind, the requirements and approaches to 3D reconstruction suitable for a real-time telepresence system were determined. Through a number of pilot experiments various approaches to 3D reconstruction, and their respective quality characteristics, were explored. This enabled the scope to be narrowed down to a polyhedral visual hull reconstruction algorithm, EPVH. In Chapter 4 three quality measures of 3D reconstruction were defined: spatial, visual and temporal. Their relationship, and factors affecting them in a complete 3D reconstruction system were investigated. One of the major hindrances to the widespread adoption of 3D reconstruction for communication purposes is the inability of commodity cameras to be synchronised at the frame level. To investigate the effect that use of such cameras might have, experiments were performed that are, to the best of our knowledge, the first indication of how a variety of human movement would be reconstructed from unsynchronised video streams. This proved that, for any appreciable movement within a typical frame period to be captured, frame level

synchronisation is necessary. Therefore, commodity cameras are not currently suitable for 3D reconstruction of highly dynamic human movement, but could be used to extend video conferencing scenarios into 3D, providing support for conveying eye-gaze.

The EPVH algorithm appeared to provide suitable spatial and visual qualities, but not at sufficient temporal quality. In order to improve temporal quality to achieve sufficient spatial and visual qualities, in Chapter 5 a parallelisation of the EPVH algorithm was presented. The parallelisation was tailored for processing on modern multi-core CPUs or GPUs, to improve the performance of 3D reconstruction achievable on commodity hardware. The local approach to parallelisation provides a better load balancing than the previous network distributed approach could, and also reduces the opportunity for network processing artefacts such as jitter in latency to confuse human interactions. Full 360° reconstructions of moving humans with a visual faithfulness comparable to video conferencing from 10 cameras at 20 to 30 frames per second were achieved using a modern 4 core CPU. There has been a great deal of research recently into using depth based cameras such as Kinect to achieve similar results. These either only produce a partial model from a single camera, or when used in combination to produce a 360° model interference between cameras causes artefacts that must be eliminated with a post-processing step, and this interference increases as more cameras are added to a system. Shape-from-silhouette based techniques, such as EPVH, on the other hand are eminently more scalable since increasing the number of cameras reduces artefacts and increases the quality of the reconstructed model.

In order to investigate the impact of camera placement on spatial and visual quality of 3D reconstruction, a platform was developed with which the balance of qualities could be investigated. The platform presented in Chapter 6 provides a number of features aimed at understanding the impact of camera placement on reconstruction qualities, and a means by which placements can be rapidly prototyped in a simulated setting in order to reduce the time spent in deploying real cameras. A collaborative experiment was carried out that validated that the research platform was suitable for its purpose by providing both the experimental

platform, and analysis tools used in determining the impact of cameras on estimation of eye gaze through 3D reconstruction. This proved for the first time that eye-gaze could be conveyed at accuracies sufficient for human social interaction.

7.3.1 Future research directions

The scope of the research undertaken provides several directions for future investigation:

Two way end-to-end 3D telepresence system

Whilst this thesis proved that in principle a two-way end-to-end 3D telepresence system could be developed, the focus was very much on the 3D reconstruction algorithm itself. A concurrent thesis studied the camera acquisition stage and provided useful findings in terms of the best methods for delivering numerous camera images to the reconstruction algorithm in a timely manner. Together we were able to demonstrate a one-way local end-to-end system in which a person was captured, reconstructed and displayed through a local network connection. Turning this into a two-way system is not a great leap and would entail replicating the one-way system in the opposite direction. What is missing from this research in order to create a true telepresence system is the ability to transmit reconstructed humans over the internet without a huge bandwidth requirement. The use of polygonal 3D meshes goes some way to achieving this as there is already significant research into efficient mesh compression algorithms. Our current system transmits camera images as compressed video streams, which limits scalability at internet bandwidths. Since textured polygonal models only use a small fraction of the overall video data transmitted, it should be possible to send only the regions used for texturing, reducing the bandwidth requirements, possibly resulting in a more scalable system.

Commodity cameras with frame synchronisation

We identified that frame level synchronisation was important for capturing human motion, but that at present the only cameras offering hardware synchronisation exist at the high end of the market. This hinders wide adoption of 3D reconstruction from multiple cameras as the next consumer communication medium. Furthermore, whilst we succeeded in removing the requirement for network distributed processing to achieve the actual 3D reconstruction from images, our complete 3D reconstruction system still uses a network distributed approach for camera image acquisition; networked computers are each responsible for one or two cameras, and collect, compress and transmit the image data to the reconstructing computer. Until very recently, network distribution of camera images was probably the fastest, and most scalable way to achieve this. Recent advances in locally hosted interfaces such as USB are now capable of achieving speeds well in excess of copper networks and this trend looks set to continue. USB 3.0 is capable of speeds up to 5 Gb/s, Thunderbolt can currently achieve 10 Gb/s, and future Thunderbolt is claiming 100 Gb/s.

The concept of a camera, providing hardware frame synchronisation, and using a high speed locally hosted interface such as USB or Thunderbolt is a compelling one when considered in combination with the local parallel processing presented in this thesis; especially as the number of parallel processing cores looks set to continue to rise. Cameras could include hardware video compression, and possibly background segmentation implemented on an FPGA, eliminating the need for a carefully thought out camera acquisition stage in 3D reconstruction system design.

Depth based cameras for background segmentation

Robust background segmentation is perhaps the most critical step in the 3D reconstruction system pipeline when using shape-from-silhouette approaches. Even in laboratory conditions, background segmentation can be obstreperous in the pres-

ence of variations in lighting and shadows. In the home, cameras including depth sensors, such as Kinect have proven robust in the presence of natural lighting and shadows. It is conceivable that such sensors could be used to replace the background segmentation process for the purposes of obtaining silhouettes. Currently Kinect itself is low resolution, the depth sensor is VGA resolution (640 x 480) which this research suggests would not provide sufficient spatial quality to pick out the individual fingers on a hand when the entire person is in the camera image for example. Furthermore, the edges of the depth image are somewhat noisy, giving rise to variable silhouette outlines even when the object being modelled is completely stationary. To add to this we have already mentioned the interference problems associated with using multiple Kinects in combination. Kinect can also be sensitive to certain materials, such as polyester and hair which scatter the projected infra-red dot pattern resulting in depth computation errors.

What is required here is a higher resolution depth based camera that can be used as part of a set without interference. These cameras would perform the background segmentation part of the 3D reconstruction process, replacing the current image based methods with a much more reliable and robust method. Image sensors could be included, or form a different array of cameras. Frame rate time-of-flight cameras, also known as Lidar may overcome some of the shortcomings of Kinect. For example the Mesa SR4000 enables multiple simultaneous cameras to be used by enabling different frequencies in the projected light and therefore eliminating interference for up to three cameras. However, the image resolution at 174 x 144 pixels does not yet meet the requirements for 3D telepresence. Similarly, the PMD CamCube offers high frame-rate (40 FPS) depth imaging at 200 x 200 pixels. Future resolution enhancements in time of flight image sensors could provide the key component required for accurate background segmentation in diverse settings and agnostic to the materials making up the object being modelled.

Bibliography

- [1] Jérémie Allard, Jean-Sébastien Franco, Clément Menier, Edmond Boyer, and Bruno Raffin. The GrImage Platform: A Mixed Reality Environment for Interactions. In *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (ICVS '06)*, New York, January 2006. IEEE Computer Society.
- [2] Jérémie Allard, Clément Menier, Bruno Raffin, Edmond Boyer, and François Faure. Grimage: markerless 3D interactions. *Proceedings of the 34th annual conference on Computer Graphics and Interactive Techniques (SIGGRAPH '07)*, August 2007.
- [3] Brett Allen, Brian Curless, and Zoran Popović. Articulated body deformation from range scan data. In *Proceedings of the 29th annual conference on Computer Graphics and Interactive Techniques (SIGGRAPH '02)*, San Antonio, July 2002.
- [4] Gene M Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the spring joint computer conference (AFIPS '67)*, Anaheim, April 1967. ACM.
- [5] Stuart M Anstis, John W Mayhew, and Tania Morley. The perception of where a face or television'portrait'is looking. *The American journal of psychology*, 82(4):474–489, 1969.
- [6] Michael Argyle and Mark Cook. *Gaze and Mutual Gaze*. Cambridge University Press, January 1976.

- [7] Nicole Atzpadin, Peter Kauff, and Oliver Schreer. Stereo analysis by hybrid recursive matching for real-time immersive video conferencing. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(3):321–334, 2004.
- [8] H Baker. Three-dimensional modelling. *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 649–655, 1977.
- [9] Bruce Baumgart. A polyhedron representation for computer vision. In *Proceedings of the national computer conference and exposition (AFPIS '75)*, pages 589–596, Anaheim, May 1975.
- [10] R Bayer and E McCreight. Organization and maintenance of large ordered indices. In *Proceedings of the ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control (SIGFIDET '70)*, Houston, November 1970.
- [11] P N Belhumeur. A binocular stereo algorithm for reconstructing sloping, creased, and broken surfaces in the presence of half-occlusion. In *Proceedings of the Fourth International Conference on Computer Vision (ICCV '93)*, pages 431–438, Berlin, May 1993.
- [12] Steve Benford, John Bowers, Lennart E Fahlén, Chris Greenhalgh, and Dave Snowdon. User embodiment in collaborative virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*, Denver, May 1995.
- [13] E Borovikov and L Davis. A distributed system for real-time volume reconstruction. In *Proceedings of the Fifth IEEE International Workshop on Computer Architectures for Machine Perception (CAMP 2000)*, pages 183–189, Padova, September 2000.
- [14] E Boyer and J Franco. A hybrid approach for computing visual hulls of complex objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, pages 695–701, Madison, June 2003.

- [15] J Butime, L Galo, and I Gutierrez. *Application of Computer Vision to 3D Reconstruction: A Survey of Reconstruction Methods*. A Survey of Reconstruction Methods. VDM Publishing, 2010.
- [16] WAS Buxton, A J Sellen, and M C Sheasby. Interfaces for multiparty videoconferences. *Video Mediated Communication.*, pages 385–400, 1997.
- [17] Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. In *Proceedings of the 30th annual conference on Computer Graphics and Interactive Techniques (SIGGRAPH '03)*, San Diego, July 2003.
- [18] B Chazelle and H Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. In *29th Annual Symposium on Foundations of Computer Science*, pages 590–600, White Plains, October 1988.
- [19] Milton Chen. Leveraging the asymmetric sensitivity of eye contact for videoconference. In *Proceedings of the Conference on Human Factors in Computing Systems (SIGCHI '02)*, Minneapolis, April 2002.
- [20] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, pages 714–720, Hilton Head Island, June 2000.
- [21] G K M Cheung, S Baker, and T Kanade. Visual hull alignment and refinement across time: a 3D reconstruction algorithm combining shape-from-silhouette with stereo. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, Madison, June 2003.
- [22] C H Chien and J K Aggarwal. Volume/surface octrees for the representation of three-dimensional objects. *Computer Vision, Graphics, and Image Processing*, 36(1), November 1986.
- [23] Changsuk Cho and Haruyuki Minamitani. Obtaining 3-d shape from silhouette informations interpolated by photometric stereo. In *Workshop on*

- Machine Vision Applications (MVA '94)*, pages 147–150, Kanagawa, December 1994. Citeseer.
- [24] Slo-Li Chu and Chih-Chieh Hsiao. OpenCL: Make Ubiquitous Supercomputing Possible. In *IEEE International Conference on High Performance Computing and Communications (HPCC 2010)*, pages 556–561, Melbourne, September 2010. IEEE.
- [25] Antonio Criminisi, Jamie Shotton, Andrew Blake, and Philip H S Torr. Gaze manipulation for one-to-one teleconferencing. In *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, pages 191–198, Nice, October 2003.
- [26] Carolina Cruz-Neira. *Virtual reality based on multiple projection screens: the cave and its applications to computational science and engineering*. PhD thesis, University of Illinois, Chicago, September 1995.
- [27] B Cyganek and J P Siebert. *An introduction to 3D computer vision techniques and algorithms*. Wiley, 2011.
- [28] E R Davies. *Machine vision: theory, algorithms, practicalities*. Elsevier, 2004.
- [29] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmans. Delaunay Triangulations. In *Computational Geometry: Algorithms and Applications*, pages 191–218. Springer-Verlag, March 2008.
- [30] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH '96)*, New Orleans, August 1996.
- [31] B Delaunay. Sur la sphere vide. *Bulletin of Academy of Sciences of the USSR*, 6:793–800, 1934.

- [32] T Duckworth and D.J Roberts. Accelerated polyhedral visual hulls using OpenCL. In *IEEE Virtual Reality Conference (VR '11)*, pages 203–204, Singapore, March 2011.
- [33] Tobias Duckworth and David J Roberts. Camera image synchronisation in multiple camera real-time 3D reconstruction of moving humans. In *IEEE/ACM 15th International Symposium on Distributed Simulation and Real Time Applications (DSRT '11)*, pages 138–144, Salford, September 2011. IEEE Computer Society.
- [34] Tobias Duckworth and David J Roberts. Parallel processing for real-time 3D reconstruction from video streams. *Journal of Real-Time Image Processing*, pages 1–19, 2012.
- [35] Tobias Duckworth and David J Roberts. 3DRecon, a utility for 3D reconstruction from video. *Joint Virtual Conference of ICAT - EGVE - EuroVR (JVRC '12)*, October 2012.
- [36] CH Esteban and F Schmitt. Multi-stereo 3d object reconstruction. *International Symposium on 3D Processing, Visualization, and Transmission (3DPVT '02)*, pages 159–166, June 2002.
- [37] OD Faugeras, M Hebert, P Mussi, and JD Boissonnat. Polyhedral approximation of 3-D objects without holes. *Computer Vision, Graphics, and Image Processing*, 25(2):169–183, 1984.
- [38] Olivier Faugeras, Quang-Tuan Luong, and T Papadopoulou. *The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications*. MIT Press, March 2001.
- [39] J Franco and E Boyer. Exact polyhedral visual hulls. In *British Machine Vision Conference (BMVC '03)*, pages 329–338, Norwich, September 2003.
- [40] J Franco and E Boyer. Efficient Polyhedral Modeling from Silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3): 414–427, March 2009.

- [41] J Franco, C Menier, E Boyer, and B Raffin. A Distributed Approach for Real Time 3D Modeling. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW '04)*, pages 31–31, Washington, June 2004.
- [42] H Fuchs, N Kelshikar, J Mulligan, and K Daniilidis. 3D Tele-Collaboration over internet 2. In *International Workshop on Immersive Telepresence (ITP '02)*, Juan Les Pin, December 2002.
- [43] Y Furukawa and J Ponce. Accurate, Dense, and Robust Multi-View Stereopsis. In *Proceedings of the IEEE Conference on Computer Graphics and Pattern Recognition (CVPR '07)*, Minneapolis, June 2007.
- [44] Y Furukawa and J Ponce. Accurate camera calibration from multi-view stereo and bundle adjustment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, Anchorage, June 2008.
- [45] Paolo Simone Gasparello, Giuseppe Marino, Filippo Bannò, Franco Tecchia, and Massimo Bergamasco. Real-Time Network Streaming of Dynamic 3D Content with In-frame and Inter-frame Compression. In *IEEE/ACM 15th International Symposium on Distributed Simulation and Real Time Applications (DSRT '11)*, pages 81–87, Salford, September 2011. IEEE.
- [46] Jim Gemmell, Kentaro Toyama, C Lawrence Zitnick, Thomas Kang, and Steven Seitz. Gaze Awareness for Video-Conferencing: A Software Approach. *IEEE Multimedia*, 7(4), October 2000.
- [47] Antonin Guttman. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the ACM Conference on Management of Data (SIGMOD '84)*, pages 47–57, Boston, June 1984.
- [48] O Hall-Holt and S Rusinkiewicz. Stripe boundary codes for real-time structured-light range scanning of moving objects. In *Proceedings of the 8th International Conference on Computer Vision (ICCV '01)*, pages 359–366, Vancouver, July 2001. IEEE Comput. Soc.

- [49] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, September 2003.
- [50] J Herraiz, S Espana, S Garcia, R Cabido, A Montemayor, M Desco, J Vaquero, and J Udias. GPU acceleration of a fully 3D Iterative Reconstruction Software for PET using CUDA. *IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC '09)*, pages 4064–4067, October 2009.
- [51] Mark D Hill and Michael R Marty. Amdahl's Law in the Multicore Era. *IEEE Computer*, 41(7):33–38, July 2008.
- [52] A Hilton, D Beresford, T Gentils, R Smith, W Sun, and J Illingworth. Whole-body modelling of people from multiview images to populate virtual worlds. *The Visual Computer*, 16(7):411–436, November 2000.
- [53] G Hu, A K Jain, and G Stockman. Shape from light stripe texture. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '86)*, pages 412–414, Miami, June 1986.
- [54] PH Huang and SH Lai. Silhouette-based camera calibration from sparse views under circular motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, Anchorage, 2008.
- [55] Hui. Camera Calibration from Images of Spheres. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):499–502, 2007.
- [56] Hiroshi Ishii and Minoru Kobayashi. ClearBoard: a seamless medium for shared drawing and conversation with eye contact. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (SIGCHI '92)*, pages 525–532, Monterey, June 1992.
- [57] CL Jackins and SL Tanimoto. Oct-trees and their use in representing three-dimensional objects. *Computer Graphics and Image Processing*, 14(3): 249–270, 1980.

- [58] Andrew Jones, Magnus Lang, Graham Fyffe, Xueming Yu, Jay Busch, Ian McDowall, Mark Bolas, and Paul Debevec. Achieving eye contact in a one-to-many 3D video teleconferencing system. *ACM Transactions on Graphics*, 28(3), August 2009.
- [59] Roy S Kalawsky. *Science of Virtual Reality and Virtual Environments*. Addison Wesley Longman Publishing Co., Inc, April 2004.
- [60] Kibum Kim, John Bolton, Audrey Girouard, Jeremy Cooperstock, and Roel Vertegaal. TeleHuman: effects of 3d perspective on gaze and pose estimation with a life-size cylindrical telepresence pod. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (SIGCHI '12)*, Austin, May 2012.
- [61] D Knoblauch and F Kuester. VirtualizeMe: interactive model reconstruction from stereo video streams. In *Proceedings of the ACM symposium on Virtual Reality Software and Technology (VRST '08)*, pages 193–196, Bordeaux, October 2008.
- [62] G Kurillo, R Bajcsy, K Nahrsted, and O Kreylos. Immersive 3D Environment for Remote Collaboration and Training of Physical Activities. In *IEEE Virtual Reality Conference (VR '08)*, pages 269–270, Reno, March 2008.
- [63] Claudia Kuster, Tiberiu Popa, Jean-Charles Bazin, Craig Gotsman, and Markus Gross. Gaze correction for home video conferencing. *Transactions on Graphics (TOG)*, 31(6), November 2012.
- [64] KN Kutulakos and SM Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.
- [65] A Ladikos, S Benhimane, and N Navab. Efficient visual hull computation for real-time 3D reconstruction using CUDA. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '08)*, Anchorage, June 2008.

- [66] Denis Laurendeau, Nathalie Bertrand, and Régis Houde. The mapping of texture on VR polygonal models. In *Proceedings of the 2nd international conference on 3-D digital imaging and modeling (3DIM '99)*, Ottawa, October 1999. IEEE Computer Society.
- [67] A Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.
- [68] S Lazebnik, E Boyer, and J Ponce. On computing exact visual hulls of solids bounded by smooth surfaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, Kauai, December 2001.
- [69] S Lazebnik, Y Furukawa, and J Ponce. Projective visual hulls. *International Journal of Computer Vision*, 74(2):137–165, 2007.
- [70] HJ Lee and Z Chen. Determination of 3D human body postures from a single view. *Computer Vision, Graphics, and Image Processing*, 30(2):148–168, 1985.
- [71] Peter Lincoln, Greg Welch, Andrew Nashel, Andrei State, Adrian Ilie, and Henry Fuchs. Animatronic shader lamps avatars. *Virtual Reality*, 15(2-3):225–238, October 2010.
- [72] Shubao Liu, Kongbin Kang, J Tarel, and D Cooper. Distributed volumetric scene geometry reconstruction with a network of distributed smart cameras. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pages 2334–2341, June 2009.
- [73] X Liu, H Yao, and W Gao. Shape from silhouette outlines using an adaptive dandelion model. *Computer Vision and Image Understanding*, 105(2):121–130, 2007.
- [74] WE Lorensen and HE Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th annual conference on*

- Computer Graphics and Interactive Techniques (SIGGRAPH '87)*, pages 163–169, Anaheim, July 1987.
- [75] C N Macrae, B M Hood, A B Milne, A C Rowe, and M F Mason. Are You Looking at Me? Eye Gaze and Person Perception. *Psychological Science*, 13(5):460–464, September 2002.
- [76] Andrew Maimone and Henry Fuchs. Encumbrance-free telepresence system with real-time 3D capture and display using commodity depth cameras. In *10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR '11)*, pages 137–146, Basel, October 2011. IEEE Computer Society.
- [77] WN Martin and JK Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:150–158, 1983.
- [78] T Matsuyama, X Wu, T Takai, and T Wada. Real-time dynamic 3-D object shape reconstruction and high-fidelity texture mapping for 3-D video. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(3):357–369, 2004.
- [79] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J Gortler, and Leonard McMillan. Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2000)*, New Orleans, July 2000.
- [80] Joel Mitchelson and Adrian Hilton. Wand-based multiple camera studio calibration. Technical report, 2003.
- [81] Theo Moons, Luc Van Gool, and Maarten Vergauwen. *3d Reconstruction from Multiple Images*. Part 1: Principles. Now Publishers Inc, November 2009.
- [82] Carl Moore, Toby Duckworth, Rob Aspin, and David Roberts. Synchronization of Images from Multiple Cameras to Reconstruct a Moving Human. In *IEEE/ACM 14th International Symposium on Distributed Simula-*

- tion and Real Time Applications (DSRT '10)*, pages 53–60, Fairfax, October 2010. IEEE.
- [83] Carl Moore, Tobias Duckworth, and David J Roberts. Investigating the Suitability of a Software Capture Trigger in a 3D Reconstruction System for Telepresence. *IEEE/ACM 15th International Symposium on Distributed Simulation and Real Time Applications (DSRT '11)*, pages 134–137, September 2011.
- [84] Lothar Mühlbach, Martin Böcker, and Angela Prussog. Telepresence in Videocommunications: A Study on Stereoscopy and Individual Eye Contact. *Human Factors*, 37(2):290–305, June 1995.
- [85] Jane Mulligan, Volkan Isler, and Kostas Daniilidis. Trinocular Stereo: A Real-Time Algorithm and its Evaluation. *International Journal of Computer Vision*, 47(1-3):51–61, April 2002.
- [86] Jane Jane Mulligan and K Kaniilidis. Trinocular stereo for non-parallel configurations. In *15th International Conference on Pattern Recognition (ICPR 2000)*, pages 567–570, Barcelona, September 2000.
- [87] H H Nagel. Spatio-temporal modeling based on image sequences . *International Symposium on Image Processing and its Applications (ISSPA '84)*, August 1984.
- [88] Christian Nitschke. *3D Reconstruction. Real-time Volumetric Scene Reconstruction from Multiple Views*. VDM-Verlag Muller, April 2007.
- [89] Yuichi Ohta, Masaki Watanabe, and Katsuo Ikeda. *Improving Depth Map by Right-angled Trinocular Stereo*. PhD thesis, University of Tsukuba, 1986.
- [90] Ken-Ichi Okada, Fumihiko Maeda, Yusuke Ichikawaa, and Yutaka Matsushita. Multiparty videoconferencing at virtual social distance: MAJIC design. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work (CSCW '94)*, pages 385–393, Chapel Hill, October 1994.

- [91] M Okutomi and T Kanade. A Multiple-Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, April 1993.
- [92] Benjamin Petit, Jean-Denis Lesage, Jean-Sébastien Franco, Edmond Boyer, and Bruno Raffin. Grimage: 3D modeling for remote collaboration and telepresence. In *ACM Symposium on Virtual Reality Software and Technology (VRST '08)*, pages 299–300, Bordeaux, October 2008. ACM.
- [93] Benjamin Petit, Jean-Denis Lesage, Clément Menier, Jérémie Allard, Jean-Sébastien Franco, Bruno Raffin, Edmond Boyer, and François Faure. Multicamera Real-Time 3D Modeling for Telepresence and Remote Collaboration. *International Journal of Digital Multimedia Broadcasting*, 2010(2): 150–162, 2010.
- [94] M Pollefeys, S Sinha, L Guan, and J Franco. Multiview Calibration Synchronization and Dynamic Scene Reconstruction. In *Multi-Camera Networks: Principles and applications*, pages 29–72. Elsevier, January 2009.
- [95] S Pollefeys. Synchronization and Calibration of Camera Networks from Silhouettes. In *IEEE International Conference on Pattern Recognition (ICPR '04)*, Cambridge, 2004.
- [96] M Potmesil. Generating octree models of 3D objects from their silhouettes in a sequence of images. *Computer Vision, Graphics, and Image Processing*, 40(1):1–29, 1987.
- [97] C Poullis, Suya You, and U Neumann. Generating High-Resolution Textures for 3D Virtual Environments using View-Independent Texture Mapping. In *IEEE International Conference on Multimedia and Expo (ICME '07)*, pages 1295–1298, Beijing, July 2007.
- [98] P Ramanathan, E Steinbach, and B Girod. Silhouette-based multiple-view camera calibration. *Conference on Vision, Modeling, and Visualization*, November 2000.

- [99] V Ramesh, Robert L Glass, and Iris Vessey. Research in computer science: an empirical study. *Journal of Systems and Software*, 70(1-2):165–176, February 2004.
- [100] Ari Rappoport and Steven Spitz. Interactive Boolean operations for conceptual design of 3-D solids. In *Proceedings of the 24th annual conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*, Los Angeles, August 1997.
- [101] Fabio Remondino and Sabry El-Hakim. Image-based 3D Modelling: A Review. *The Photogrammetric Record*, 21(115):269–291, September 2006.
- [102] 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 *IEEE Virtual Reality Conference (VR '09)*, pages 135–142, Lafayette, March 2009. IEEE.
- [103] David Roberts, Toby Duckworth, Carl Moore, Robin Wolff, and John O'Hare. Comparing the end to end latency of an immersive collaborative environment and a video conference. *IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (DSRT '09)*, pages 89–94, October 2009.
- [104] David J Roberts, John Rae, Tobias W Duckworth, Carl M Moore, and Rob Aspin. Estimating the Gaze of a Virtuality Human. *Visualization and Computer Graphics, IEEE Transactions on*, 19(4):681–690, February 2013.
- [105] Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-time 3D model acquisition. In *Proceedings of the 29th annual conference on Computer Graphics and Interactive Techniques (SIGGRAPH '02)*, pages 438–446, San Antonio, July 2002.
- [106] Oliver Schreer, Ingo Feldmann, Nicole Atzpadin, Peter Eisert, Peter Kuff, and Harm J W Belt. 3DPresence -A System Concept for

- Multi-User and Multi-Party Immersive 3D Videoconferencing. In *Visual Media Production (CVMP 2008), 5th European Conference on*, London, November 2008.
- [107] Abigail J Sellen. Remote conversations: the effects of mediating talk with technology. *Human-Computer Interaction*, 10(4):401–444, December 1995.
- [108] Timos K Sellis, Nick Roussopoulos, and Christos Faloutsos. The R+ Tree: A Dynamic Index for Multi-Dimensional Objects. In *Proceedings of the 13th International Conference on Very Large Data Bases (VLDB '87)*, Brighton, September 1987. Morgan Kaufmann Publishers Inc.
- [109] R Shen, I Cheng, and A Basu. Multi-Camera Calibration Using a Globe. In *8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS '08)*, Marseille, September 2008.
- [110] I Shlyakhter, M Rozenoer, J Dorsey, and S Teller. Reconstruction of plausible 3D tree models from instrumented photographs. *IEEE Computer Graphics and Applications*, 21(3):53–61, May 2001.
- [111] Bo Shu, Xianjie Qiu, and Zhaoqi Wang. Hardware-based camera calibration and 3D modelling under circular motion. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '08)*, June 2008.
- [112] Zhang Shujun, Wang Cong, Shao Xuqiang, and Wu Wei. DreamWorld: CUDA-accelerated real-time 3D modeling system. In *IEEE International Conference On Virtual Environments, Human Computer Interfaces and Measurement Systems (VECIMS '09)*, pages 168–173, Hong Kong, May 2009.
- [113] J Starck. *Human Modelling from Multiple Views*. PhD thesis, University of Surrey, University of Surry, June 2003.

- [114] Jonathan Starck, Gordon Collins, Raymond Smith, Adrian Hilton, and John Illingworth. Animated statues. *Machine Vision and Applications*, 14(4): 248–259, September 2003.
- [115] G C Stockman, S W Chen, G Hu, and N Shrikhande. Sensing and recognition of rigid objects using structured light. *IEEE Control Systems Magazine*, 8(3):14–22, June 1988.
- [116] S Sullivan and J Ponce. Automatic Model Construction, Pose Estimation, and Object Recognition from Photographs Using Triangular Splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1091–1096, January 1998.
- [117] R Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Understanding*, 58(1):23–32, September 1993.
- [118] Peter Thoman, Klaus Kofler, Heiko Studt, John Thomson, and Thomas Fahringer. Automatic OpenCL device characterization: guiding optimized kernel design. In *International conference on Parallel processing (ICPP '11)*, Taipei, September 2011. Springer-Verlag.
- [119] H Towles, SU Kum, T Sparks, S Sinha, S Larsen, and N Beddes. Transport and rendering challenges of multi-stream, 3d tele-immersion data. In *NSF Lake Tahoe Workshop on Collaborative Virtual Reality and Visualization (CVRV'03)*, Lake Tahoe, October 2003.
- [120] R Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, August 1987.
- [121] Roel Vertegaal. The GAZE groupware system: mediating joint attention in multiparty communication and collaboration. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '99)*, Pittsburgh, May 1999.
- [122] John Vince. *Virtual reality systems*. ACM Press/Addison-Wesley Publishing Co, February 1995.

- [123] T Wada, X Wu, S Tokai, and T Matsuyama. Homography based parallel volume intersection: Toward real-time volume reconstruction using active cameras. In *Proc of Computer Architectures for Machine Perception (CAMP 2000)*, pages 331–339, Padova, September 2000.
- [124] P M Will and K S Pennington. Grid coding: a preprocessing technique for robot and machine vision. In *Proceedings of the 2nd international joint conference on Artificial intelligence (IJCAI'71)*, pages 66–70, London, September 1971. Morgan Kaufmann Publishers Inc.
- [125] K Wong and R Cipolla. Epipolar geometry from profiles under circular motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, February 2001.
- [126] Xiaojun Wu, O Takizawa, and T Matsuyama. Parallel Pipeline Volume Intersection for Real-Time 3D Shape Reconstruction on a PC Cluster. In *IEEE International Conference On Computer Vision Systems (ICVS '06)*, New York, January 2006.
- [127] Allan Young. Remembering the Evolutionary Freud. *Science in Context*, 19(01):175–189, March 2006.
- [128] S Zhang and ST Yau. Three-dimensional shape measurement using a structured light system with dual cameras. *Optical Engineering*, 47(1), January 2008.
- [129] Song Zhang and Peisen Huang. High-Resolution, Real-time 3D Shape Acquisition. In *Computer Vision and Pattern Recognition Workshop, 2004. (CVPRW '04)*, pages 1–10, Washington, July 2004. State University of New York.
- [130] Zhao and Taubin. Real-time stereo on GPGPU using progressive multi-resolution adaptive windows. *Image and Vision Computing*, 29(6):420–432, April 2011.

- [131] Zhengyou. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):892–899, July 2004.

Appendix A

Equations

A.0.2 Pinhole camera model

The pinhole camera model relates points in 3D space to points on the camera image plane through the equation:

$$x = PX \tag{A.1}$$

where x is the 2D image coordinates of the point, P is the 3x4 projection matrix, X is the coordinates of the point in 3D space.

The 3x4 projection matrix can be written as:

$$P = C[R|T] \tag{A.2}$$

C is the 3x3 camera calibration matrix, where R is a 3x3 rotation matrix, T is the translation vector between the origin of world coordinates and the camera origin.

The 3x3 camera calibration matrix C is defined as: $C = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$

where α_u and α_v are the scaling in the image x and y directions respectively, (u_0, v_0) is the principle point of the camera image

A.0.3 Formation of a unit vector from an image point

A unit vector v of the projection of a point x on the camera image plane can be obtained as follows:

$$\bar{v} = P_{3 \times 3}^{-1} x \quad (\text{A.3})$$

where x is the image position in homogenous coordinates $x = [uv1]^T$, $P_{3 \times 3}$ is the left-most 3x3 sub matrix of the 3x4 projection matrix.

A.0.4 Projection at infinity of a unit vector on a camera image plane

The projection p_∞ of a unit vector onto a camera image plane (maximum epipolar extent) is calculated by:

$$p_\infty = P_{3 \times 3} \bar{v} \quad (\text{A.4})$$

where $P_{3 \times 3}$ is the left-most 3x3 sub matrix of the camera 3x4 projection matrix, \bar{v} is the unit vector

A.0.5 Projection of the minimum epipolar extent

To determine the minimum extent of an epipolar line between two camera images, the following is used:

$$p_0 = P_{3 \times 3}(C_{des} - C) \quad (\text{A.5})$$

where $P_{3 \times 3}$ is the left-most 3x3 sub matrix of the camera 3x4 projection matrix, C_{des} is the origin in world coordinates of the source camera, C is the origin in world coordinates of the destination camera.

A.0.6 Projection of a 2D epipolar line intersection onto a 3D line

To determine the distance along the camera's principle ray of a 2D epipolar line intersection:

$$t = |\bar{v} \times P_{3 \times 3}^{-1} x| \quad (\text{A.6})$$

where \bar{v} is the unit vector of the camera from which the principle ray is formed, $P_{3 \times 3}$ is the left-most 3x3 sub matrix of the 3x4 projection matrix from camera in which the epipolar line position is found, x is the homogenous coordinates of the epipolar line in the camera image plane.