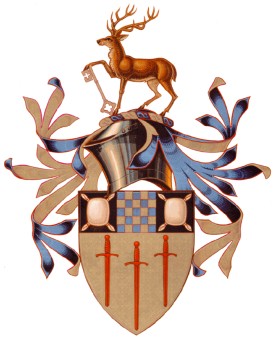
High Quality Novel View Rendering from Multiple Cameras

Gregor Miller

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey



Centre for Vision, Speech and Signal Processing
School of Electronics and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, UK

December 2007

Abstract

The research presented in this thesis is targeted towards obtaining high quality novel views of a dynamic scene using video from multiple wide-baseline views, with free-viewpoint video as the main application goal. The research has led to several novel contributions to the 3D reconstruction computer vision literature.

The first novel contribution of this work is the exact view-dependent visual hull, a method to efficiently reconstruct a three dimensional representation of the scene with respect to a given viewpoint. This approach includes two novel contributions which allow the reconstruction to be performed in the image domain. The first is the Visual Hull Visible Intersection Theorem, an efficient way to identify points on the visual hull surface from the input images. The second is the use of the cross ratio to globally order intersections from individual images, avoiding the need for explicit 3D reconstruction of every point. This not only increases the efficiency of the reconstruction, it also produces an exact representation of the visual hull by maintaining pixel accuracy in the original images.

A method for for producing high quality novel views through efficient local surface refinement is introduced. This reduces artefacts such as ghosting from incorrect correspondence between views when using the visual hull. A representation for rendering the refined surfaces in real-time with a user-controllable viewpoint is introduced.

An alternative method for producing high quality novel views from wide-baseline cameras using a global optimisation to refine the entire surface is presented. The goal of this is to produce a continuous surface which removes depth artefacts and represents the overall shape of the scene. The optimisation is constrained by surface contours called rims extracted from the visual hull, to avoid over-refinement of the surface.

The final novel contribution of this thesis is the safe hull, the first visual hull based reconstruction method which guarantees production of a surface without phantom volumes (an artefact of visual hull reconstruction, due to multiple objects in a scene). The safe hull identifies volumes inside the visual hull which only contain foreground i.e. the object to be reconstructed. This approach uses a novel geometric constraint, utilising information gained from the exact view-dependent visual hull, unlike other solutions which are either heuristic or require additional cameras.

Acknowledgements

I would like to thank Adrian Hilton, firstly for the opportunity to pursue this research, and also for his support and encouragement over the duration of my PhD. Thanks also goes to Jonathan Starck for his help and advice, and involving me in different research projects. Heartfelt thanks go to my family who have been an amazing source of support for me.

List of Publications

1. J. Starck, G. Miller and A. Hilton, Video-Based Character Animation, Symposium on Computer Animation 2005.

2. G. Miller, A. Hilton and J. Starck, Interactive Free-Viewpoint Video, Proceedings of Conference on Visual Media Production 2005.

3. O. Grau, A. Hilton, J. Kilner, G. Miller, T. Sargeant and J. Starck, A Free-Viewpoint Video System for Visualisation of Sport Scenes, International Broadcasting Conference 2006.

4. O. Grau, A. Hilton, J. Kilner, G. Miller, T. Sargeant and J. Starck, A Free-Viewpoint Video System for Visualisation of Sport Scenes, From IT to HD 2006.

5. O. Grau, A. Hilton, J. Kilner, G. Miller, T. Sargeant and J. Starck, A Free-Viewpoint Video System for Visualisation of Sport Scenes, SMPTE Motion Imaging Journal 2007.

6. J. Starck, G. Miller and A. Hilton, Volumetric Stereo with Silhouette and Feature Constraints, British Machine Vision Conference 2006.

7. G. Miller and A. Hilton, Exact View-Dependent Visual Hulls, International Conference on Pattern Recognition 2006.

8. G. Miller, J. Starck and A. Hilton, Projective Surface Refinement for Free-Viewpoint Video, Proceedings of Conference on Visual Media Production 2006.

9. G. Miller and A. Hilton, Safe Hulls, Proceedings of Conference on Visual Media Production 2007.

Contents

List of Figures

Chapter 1

Introduction

In classical theology God has traditionally been described as omniscient (all knowing) and omnipotent (all powerful). He was also said to be all seeing. While the former abilities are still beyond us, twenty first century computer vision techniques could well bring the latter within the reach of mere mortals.

The research presented in this thesis is targeted towards obtaining high quality novel views of a dynamic scene using video from multiple wide-baseline views, with free-viewpoint video as the main application goal. Rendering of real events from novel views is of interest for broadcast and film production, video games and visual communication. Ultimately the objective is to allow user interactive control of the viewpoint while producing images with a visual quality comparable to captured video.

Multiple camera capture systems have been widely developed to allow capture of real events both in the studio and in outdoor environments such as a sports arena. Studio captures allow high quality special effects such as freezing time or camera motions that would be physically impossible. For outdoor scenes an example application is to provide a virtual camera positioned where the director would like a view; sports broadcasts are often limited by what the stadium offers.

Generally the use of many cameras in a multiple view video setup increases the quality of view synthesis, but these systems are costly and difficult to set up and maintain. The research presented here is targeted towards using a minimal number of widely spaced cameras (greater than 20° between views), while still producing high quality novel views.

A review of previous work in free-viewpoint video and surface reconstruction from multiple cameras is presented in Chapter 2.

The first novel contribution of this work is the *exact view-dependent visual hull*, a method to efficiently reconstruct a three dimensional representation of the scene with respect to a given viewpoint. This approach includes two novel contributions which allow the reconstruction to be performed in the image domain. The first is the Visual Hull Visible Intersection Theorem, an efficient way to identify points on the visual hull surface from the input images. The second is the use of the cross ratio to globally order intersections from individual images, avoiding the need for explicit 3D reconstruction of every point. This not only increases the efficiency of the reconstruction, it also produces an exact representation of the visual hull by maintaining pixel accuracy in the original images. Details of this method can be found in Chapter 3. The research on the exact view dependent visual hull was presented at the International Conference on Pattern Recognition in 2006[58].

The goal of the research presented here is to produce high quality novel views. The first method to accomplish this is presented in Chapter 4, where a novel method for surface refinement is introduced. This is required because the novel view rendering using the visual hull surface produces artefacts such as ghosting from incorrect correspondence between views. A representation for rendering the refined surfaces in real-time with a user-controllable viewpoint is also described. The view-dependent visual hull is used as an initialisation, which allows a stereo matching algorithm to be used across wide baseline views. For points

on the reconstructed surface which are not colour consistent between views, the surface is refined using a stereo correspondence technique to ensure the colour matches. This work was presented at the Conference on Visual Media Production in 2005[60].

Chapter 5 presents another method for producing high quality novel views from wide-baseline cameras. This approach uses a global optimisation to refine the entire surface, and not just locally as in the previous approach. The goal of this is to produce a continuous surface which removes depth artefacts and represents the overall shape of the scene. The optimisation is constrained by surface contours called *rims* extracted from the visual hull, to avoid over-refinement of the surface. The research on global refinement was presented at the Conference on Visual Media Production in 2006[61]. Parallel research on this topic using a volumetric visual hull and global surface representation was presented at the British Machine Vision Conference in 2006[77].

Finally, the last novel contribution of this thesis is the *safe hull*, the first visual hull based reconstruction method which guarantees production of a surface without phantom volumes (an artefact of visual hull reconstruction, due to multiple objects in a scene). The safe hull identifies volumes inside the visual hull which only contain foreground i.e. the object to be reconstructed. This approach uses a novel geometric constraint, utilising information gained from the exact view-dependent visual hull, unlike other solutions which are either heuristic or require additional cameras. The safe hull reconstruction method is described in detail in Chapter 6, and was presented at the Conference on Visual Media Production in 2007[59].

The evaluation of the research was approached using two methods. For the surface reconstruction algorithms (view-dependent visual hull and safe hulls) the computed surfaces were compared to a synthetic model ground truth data set. The exact view-dependent visual hull was found to produce a higher quality surface

when compared to a volumetric visual hull algorithm, and in general produced a good approximation of the scene surface. The safe hull improved further on this by removing visual hull surface artefacts due to occlusion. The algorithms designed to improve the quality of synthesised views were evaluated using the missing view test: from a capture setup, one camera is removed from processing and used as the target viewpoint for novel view synthesis. The synthesised image is then compared to the original captured image to evaluate the quality, both qualitatively and quantitavely. The local refinement algorithm presented in Chapter 4 produces the highest quality output for the synthesised view, while the algorithm from Chapter 5 produces the most consistent refined surface (the surface produced has a lower variation in surface normal). Chapter 7 discusses the conclusions of research presented in this thesis and provides an outlook on future work.

Other research carried out in association with work presented in this thesis resulted in additional publications. Work on surface reconstruction and representation for character animation was presented at the Symposium on Computer Animation in 2005[76]. The research presented throughout this thesis also contributed to the iview project, a collaboration involving the University of Surrey, BBC Research & Development and Snell & Wilcox, working on free-viewpoint video for sports broadcasts[36, 37, 38].

Chapter 2

Literature Review

This chapter presents a review of past research into novel view synthesis, free-viewpoint video and surface reconstruction techniques.

2.1 Free-viewpoint video

High quality view synthesis of real events via multiple view video has been a long term goal in media production and visual communication. Novel view rendering is useful for special effects, unusual perspectives and for scenes where camera placement is limited (e.g. a football stadium or a concert). The aim is to produce virtual view video with a comparable quality to captured video. Free-viewpoint video systems have been developed to capture real events in studio and outdoor settings. The challenge is to produce good quality views from a limited number of cameras.

The *Virtualized Reality*™ system[43] reconstructs dynamic scenes using images captured from a 51 camera hemispherical dome. Narrow baseline stereo is used between views to produce depth maps which are subsequently fused into a single

3D surface. This process relies on stereo matching producing accurate geometry, which is not the case in areas of uniform or regular appearance. Grau designed a real-time studio system based on low-resolution volumetric visual hull[35]. Wuermlin et al. used a variant of image-based visual hulls for free-viewpoint video using point samples instead of mesh with texture, and applied splatting techniques for rendering novel views[89]. Vedula et al. introduced scene flow, based on volumetric visual and photo hull to produce free-viewpoint video using a temporally consistent surface[82]. These approaches reconstruct an approximate geometry which limits the visual quality of novel views due to incorrect correspondence between captured images[11].

2.2 Visual hull

Free-viewpoint video research widely uses the *visual hull* to synthesise novel viewpoints, either directly or as an approximation to the surface for refinement. Given N views, the set of captured images $\mathcal{I} = \{\mathcal{I}_n : n = 1, \ldots, N\}$ is converted into a set of silhouette images $\mathcal{S} = \{\mathcal{S}_n : n = 1, \ldots, N\}$ via foreground segmentation. The *silhouette cone* for the n^{th} view is produced by casting rays from the camera centre $\boldsymbol{c}_n$ through the occupied pixels in the silhouette S_n. The visual hull is the three dimensional shape formed by the intersection of all views' silhouette cones[47]. It is used in applications as diverse as crowd surveillance, 3D modelling of objects and medical imaging.

Various algorithms for constructing the visual hull have been presented, the most common of which is the volumetric approach. A volumetric grid where each element is tested against $\mathcal{S}$ is a simple and robust way to generate an approximate surface[71]. Real-time systems for generating visual hull surface using volumetric analysis has been demonstrated, either using multiple systems[15] or using low resolution volume grids[35]. Szeliski proposed a method of real-time volumetric

visual hull generation for rotating objects using octrees to increase the efficiency of construction [78].

There are many other methods for constructing the visual hull. Niem introduced a line-based representation for visual hull (similar to volumetric)[64]. Franco et al. [30] presented a technique to recover the exact representation of the visual hull corresponding to a polyhedral approximation of the silhouette contour. Brand et al. [10] describe a method of applying differential geometry to obtain a close estimate to the exact visual hull surface from silhouette contours. Li et al. developed a number of graphics hardware based techniques for constructing the visual hull in real-time[52, 51]. Lazebnik et al. worked on a method of visual hull construction which characterises the surface as a generalised polyhedron and uses projective differential geometry to perform the reconstruction[49].

Matusik et al. introduced the image-based visual hull, a method to construct the visual hull in real-time with respect to a specific viewpoint[56]. This approach uses various approximations which affect the quality of the resulting view. The research in Chapter 3 was inspired by this paper, and presents a technique with no intermediate approximations from the input silhouettes to the output surface.

Due to the limited accuracy of visual hull reconstruction and correspondence between views these approaches result in visual artefacts such as ghosting and blur. Loss of visual quality compared to captured video limits their application for visual content production. Constructing an exact representation of the visual hull surface and using this as an initialisation for a refinement algorithm improves the quality of the novel rendered views.

2.2.1 Rim extraction

The *bounding edge representation*[17] of visual hull exploits the unique property of the set of pixels $\mathcal{B}_n$ on the boundary of S_n: the ray cast from $\boldsymbol{c}_n$ through $p \in \mathcal{B}_n$

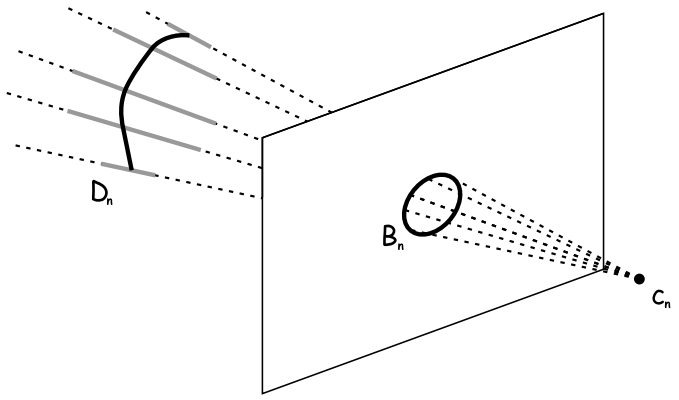


Figure 2.1: The dotted lines represent rays from the camera centre $\boldsymbol{c}_n$ through pixels p on the boundary of the silhouette, $\mathcal{B}_n$. The grey lines represent the intervals $\mathcal{D}_n$ on these rays where they intersect the visual hull. The black line through the intervals is the rim, representing the intersection of the real surface with the visual hull intervals.

touches the surface of the scene object tangentially. This is shown in Figure 2.1 where the rays from $\boldsymbol{c}_n$ through the boundary pixels $\mathcal{B}_n$ are intersected with the visual hull to give the intervals on the rays $\mathcal{D}_n$. Each interval has a scene object point which is evaluated using colour consistency from neighbouring cameras. A point cloud is produced for every frame in a sequence of multiple view video and used to align subsequent frames to the first (effectively adding cameras to the scene).

The smooth curve through the points on $\mathcal{D}_n$ is called the *rim* of the visual hull, shown as a black line through the grey intervals in Figure 2.1. The rim may not be smooth using the bounding edge representation; locating the correct or continuous point on an interval fails when the surface appearance is uniform or regular. Points on adjacent intervals will not necessarily match up correctly. Rim extraction has proved more popular in recent work, since it provides an additional constraint on scene surfaces[48, 70, 29]

2.3 Space carving

Voxel colouring has a similar foundation to the volumetric reconstruction approach, and extends it to use colour information in the images rather than silhouettes to reconstruct the scene[67, 22]. It is assumed that a surface point projects to a similar colour on all the images in which it appears, because if this was not the case then surface points would not be comparable across images. Voxel occupancy is tested by comparing the colour of the pixel onto which the voxel projects on all views, assuming it is visible. If the voxel passes a colour similarity test it is *photo consistent* and stored as part of the model[46]. Those which are inconsistent with the input images are marked as transparent, and any previously occluded voxels visible through the newly transparent elements are tested. On termination of the process, the remaining voxels describe the *photo hull* which is, like the volumetric approach, a superset of the actual object, but also a subset of the visual hull i.e. $object \subseteq photo\,hull \subseteq visual\,hull$. Photo consistency produces a more precise approximation to the object than voxel carving, at the expense of sensitivity to appearance and more complicated, time consuming computation.

There has also been interest in real-time view-synthesis for video conferencing using photo hull or stereo to correct viewpoint distortions [19, 1, 23], and also developing real-time graphics hardware based implementations of photo hull[91].

2.4 Stereo

The problem of shape reconstruction from pairs of images is known as *stereo vision*, and is one of the oldest in computer vision[45]. The challenge is solving the *correspondence problem*, finding areas of each image which correspond to the same point in the scene[26]. Once such a correspondence has been found the point can be triangulated to determine its coordinates in three dimensions. The

result is a depth map computed from all the correspondences found between the images.

The search for correspondences can be simplified by employing geometric constraints and making assumptions about the scene[54, 2]. The *epipolar constraint* guarantees that (with calibration information) a point on one image will lie on the *epipolar line* of that point on the other image. Given an opaque object, the *uniqueness constraint* states that a point on one image has a unique match in the other. In practice there may be more than one match (from objects of a single colour, for example), so this constraint alone is not enough to guarantee a correspondence, but it can help verify a match found by other means. When an object of similar colour is being reconstructed from stereo, it can be useful to employ the *continuity constraint*, which assumes the surface of the object is smooth. Erroneous matches which produce depths inconsistent with previous values can therefore be removed. Finally, the *ordering constraint* confines matches between images to be in the same order on each image (except for areas containing occlusions or different objects) so correspondences for a single surface can be verified.

The following provides a brief overview of the main techniques in stereo vision.

Dense (also known as *area based*) matching solves the correspondence problem by finding many matches between images to produce a densely populated depth map. There are two general methods for dense stereo matching, differing over the subject of comparison. The first finds correspondences for every pixel, normally using cross-correlation (comparing a window surrounding the pixel with windows in the other image and using a similarity function to decide if it matches) [65, 25, 63, 20]. This technique is sensitive to noisy images and lighting discrepancies between views, and performs badly for regions of similar intensity. It is also slow due to the number of comparisons it makes per pair of images (one per pixel). The second method for dense stereo takes advantage of *features* in the images, areas

distinct from the surrounding region such as lines or corners[94, 6]. The image must contain many features to produce a dense depth map. An experimental analysis of dense stereo techniques is conducted in [79] and [66].

An efficient variant of dense stereo was recently employed for view synthesis in a teleconferencing application to facilitate eye contact during communication[20]. Two cameras were positioned either side of the participant and the direction of gaze established. A viewpoint in the opposite direction was synthesised to produce an image of the participant staring directly at the screen.

Feature matching is similar to the second of the two methods mentioned above, except it extracts more distinct features for comparison with the other image[69, 53, 31]. These features are less common in images and so this approach results in a sparse depth map. However, feature matching is more robust than dense stereo because features are less sensitive to noise and colour differences, and more efficient due to less correspondences for which to search. The disadvantage is that it provides a much less detailed depth map than dense stereo.

The stereo techniques described above are generally used for static scenes or frame by frame analysis of moving scenes. *Dynamic* stereo uses motion cues in the images to aid in depth map construction, either from a dynamic scene or from a moving camera. Motion can be determined using optical flow between frames and combined with stereo to obtain a relative depth map[39]. Another approach recovers the camera motion between viewpoints and incrementally refines a depth map after every frame[81]. Visual navigation is the main application of dynamic stereo, employed by robots to identify and avoid obstacles.

Active methods transmit energy to aid in depth estimation as opposed to other stereo techniques which are passive. Many techniques project structured light, such as a bar code, onto an object and infer shape from the observed deformations of the light[4]. Another example of active stereo projects infrared dots onto a face, and the dots are captured by six infrared cameras. Stereo applied to the dots is

used to reconstruct the object in 3D. Three colour cameras take images at the same time to provide colour information for the final model, producing a very realistic face model[92].

Spatial (dense or feature based), dynamic and active stereo methods were recently unified into a general framework called *spacetime stereo*. Images are considered over space and time to produce more reliable information, in applications such as retrieving geometry for static scenes with varying illumination[24]. Another approach temporally sheared small windows on the image to construct a model for a moving object in the scene[93]. These techniques improved the quality of the model and the robustness of the depth calculation, but would require a significant number of cameras to cover a studio scene. They are also computationally intensive and therefore unsuitable for online applications.

Recent approaches to novel view synthesis of dynamic scenes have used image-based rendering approaches with reconstruction of geometry only as an intermediate proxy for correspondence and rendering [75, 95]. Zitnick et al. [95] simultaneously estimate foreground/background segmentation and stereo correspondence. This system achieves highly realistic view synthesis but is restricted to a narrow baseline camera configuration (8 cameras over 30°). Starck et al. [75] introduced a view-dependent optimisation for high quality rendering from wide-baseline views (7 cameras over 110°). This approach uses an initial coarse approximation of the scene geometry based on the visual hull. The initial coarse approximation is iteratively optimised for stereo correspondence to render novel viewpoints. These approaches achieve a visual quality comparable to the captured video but do not allow rendering of novel viewpoints at interactive rates.

Woodford et al. apply a multiple view stereo technique using a graph cut optimisation to produce high quality transitions between observed images of complicated scenes[88]. Goesele et al. [32] developed a technique for depth map construction using stereo correspondence and fusion using a volumetric grid. This

work produces accurately reconstructed surfaces when given a large number of input images (each point must be viewed by at least three cameras and have a high correlation score). These approaches either require narrow-baseline views or many input images, making the approach prohibitive for dynamic scenes with wide-baseline views.

2.4.1 Graph cuts

A *flow network* $G = (V, E)$ is a graph with vertices V and edges E, where each edge $(u, v) \in E$, $u, v \in V$ has a capacity $c(u, v)$[18]. G has a source $s \in V$ and a sink $t \in V$ defining the direction of flow. A *graph cut* (S, T) of G partitions V into S and $T = V - S$ such that $s \in S$ and $t \in T$. The capacity of a cut is $c(S, T) = \sum_{u \in S, v \in T} c(u, v)$. Finding a flow in G with the maximum value from s to t is known as the maximum flow problem, which, by the *max-flow min-cut theorem*, is equivalent to finding the minimum capacity cut of G.

Graph cuts on flow networks have become a popular way to solve optimisation problems in computer vision. Recent evaluation of multiple view surface reconstruction[68] show techniques based on graph cuts produce the most accurate results. This paper presents methods to recover the rims and refined surface of the object via graph cuts. The optimisation uses good scores as constraints across regions of similar scores to compensate for unreliable areas.

Previous work has shown how surface reconstruction can be accomplished using graph cuts. Snow et al. demonstrated the use of graph cuts for constructing a volumetric visual hull[73]. Boykov optimised a stereo reconstruction for a virtual view to produce a depth map, but without restricting the search space[9] (the work in this thesis uses the visual hull to restrict the search space and increase the reliability of stereo correspondence across wide-baseline views). Multi-view stereo with graph cuts has become a popular method, however visual hull and

silhouette constraints are not generally taken into account[44, 83]. Campbell et al. applied a multi-view stereo algorithm optimised by graph cuts to automatically perform segmentation and reconstruction[12], assuming the object was centred in the original images and had a similar colour profile throughout. These approaches tend to use large numbers of input views of static scenes, and are not suitable for wide-baseline observation of a dynamic scene. Sinha et al. employed volumetric visual hull and silhouette constraints in a single graph cut optimisation, but only for genus zero objects without self-occlusion[70]. The research presented in Chapter 5 uses exact visual hull and silhouette constraints to construct a surface of an arbitrary scene via graph cut optimisations.

2.5 Model-based view synthesis

Model-based free-viewpoint video has been popular due to the quality of the model output, when observing known objects. Carranza et al. introduce a modelling system where a deformable human body model is adjusted to fit silhouettes segmented from the input images[13]. Starck and Hilton fit a human model to a refined volumetric visual hull reconstruction of the person which is subsequently used for rendering[74]. Ivekovic and Trucco use stereo disparity space to fit a human model using evolutionary pose estimation to improve the quality of view synthesis[42].

These methods can produce high quality results, but only for known scenes. They also require much more detailed models to increase the detail and quality of the novel rendered views. The techniques presented in this thesis perform arbitrary scene reconstruction which gives more flexibility for scene capture.

2.6 Temporal Modelling

A recent advance in shape reconstruction has been the incorporation of temporal information. The earliest methods used a single calibrated camera taking pictures of an object on a turntable at regular intervals[64]. The rotation of the object between images is known, so they are treated as additional views and used to construct the visual hull. Generally, if the motion of the object between frames is known then every image after the first frame acts as a new camera which will further refine the model. More sophisticated techniques involving turntables were developed which did not require calibrated cameras or known rotations to construct the model[28, 86]. Temporal modelling recently advanced to multiple camera setups to reduce visual artefacts in the synthesised view[82]. The shape of the scene is determined at every frame using a volumetric method, and scene motion is determined between frames from the original images (*scene flow*). Novel views are synthesised for a particular instant by blending the information from the frames before, during and after the current frame.

The boundary representation for the visual hull described above was also used to help exploit temporal information[17]. It was designed as a means to find correspondences between frames to discover the object's rigid motion, and then to provide a dense point cloud which can be triangulated to provide the final mesh.

The technique uses the fact that every ray cast out through a pixel on the boundary of the silhouette touches the surface at least once. Under the assumptions used for photo hulls, a colour consistency check is used to find the point on the ray which touches the object. The camera from which the ray originated cannot reliably retrieve colour for the point on the surface, therefore two other cameras from which it is visible are used to find the most colour consistent pixel on the ray. The 3D coordinate of this point is found by triangulating the pixels in the

original images. Performing this for every pixel on the silhouette boundary from every camera provides a dense and coloured point cloud.

The process is repeated for the images from the next instant in time. To obtain the motion parameters of the object from the previous frame (f_{t-1}) to the current frame (f_t), the colour information of the points from f_t are aligned with the images from f_{t-1}, and vice versa. The motion information allows the points from f_t to be moved to the same position as the points in f_{t-1}, increasing the detail of the surface. This approach was extended to refining models of humans by treating each part of the body as a single rigid object[16]. The extraction of boundary points uses photo consistency only, and the points are individually selected. The reliability of the boundary point selection could be improved by using a stereo correspondence and optimising the rims along the continuous edges of the surface, as demonstrated by this research in Chapter 5.

Goldluecke and Magnor presented a method for representing a scene as a single surface in space-time, and is optimised using photo consistency across the entire sequence. This approach is computationally intensive[33].

2.7 Image-based rendering

From a set of viewpoints it is not necessary to reconstruct the shape of objects in a scene to generate novel views. There are a variety of techniques which resample colour from input photographs, and are generally far more realistic than a reconstruction based approach.

The following methods represent a scene as a large collection of images. New views are generated by interpolating between these images, or by using them to map light rays travelling towards the scene. This can produce very realistic results, although to capture the state of light in a scene many images are required

(at least hundreds), which is impractical for large scenes. This general approach is currently only suitable for static scenes, and once produced there is no way to manipulate the data to synthesise new scenes. However, viewing is independent of the contents of the scene, so no matter how complex it is the rendering time is always the same.

2.7.1 Mosaics and environment maps

Mosaics are cylindrical or spherical images that allow view generation using a small number of input photographs. Initial techniques used a calibrated camera on a motorised tripod, capturing images at regular intervals as it rotated round 360° [57]. The images were subsequently merged (*stitched*) to form cylindrical images, which are easy to manipulate and store, unlike spherical images. The cylindrical images allowed horizontal and limited vertical navigation, including pan, rotation and zoom capabilities. A famous application of this technology is Apple Computer's Quicktime VR system[14], which is widely used on the Internet and in computer games to provide realistic looking virtual reality. Techniques were later developed which used a hand held digital camera to generate a spherical mosaic, without requiring calibration information for the camera[80]. Navigation of a virtual world is achieved by connecting distinct points in the scene (each having a mosaic) by video sequences, but altering the camera position in space is not possible. A relatively large number of images are required for a high quality mosaic, therefore the capture of dynamic scenes is also impractical.

An *environment map* is effectively a texture which contains information on a real scene, and is projected onto a three dimensional object as a reflection to give the illusion that the object is in the scene. An image mosaic can be used as an environment map of a real scene, and can help virtual objects appear to be part of the scene.

2.7.2 Texture mapping

Although it cannot be used to synthesise new views, *texture mapping* is one form of image based rendering. Given a three dimensional model, images are mapped onto it to provide colour and environment information[5]. The process of texture mapping refers to the function which maps textures (colour information, such as clothing for a person) onto three dimensional objects[41]. The complexity of this operation is low but yields high quality results which would be hard to achieve without modelling every detail of the object. In the case of view synthesis, areas of the original images are mapped onto the reconstructed model to produce a heightened sense of realism.

2.7.3 Light fields

Light field rendering creates new views from arbitrary camera positions by combining and resampling the available images[50]. The light field represents a static scene's light flow, assuming fixed illumination. The light field is defined as the radiance at a point in a given direction, and it is created using *light slabs*. This technique is unsuitable for dynamic scenes, as hundreds or thousands of images are required for a full light field at each frame. It also requires that viewpoints not be inside the convex hull of the target object.

The *lumigraph*, developed concurrently with light field rendering, captures the surrounding colour data by recording the properties of light inside the environment[34]. A virtual cube is set up surrounding the object and the three sets of opposing sides on the cube are used in the same way as light slabs from light field rendering. Realistic images of the object can then be constructed from the lumigraph function. Similar to light field rendering, this also suffers from viewpoint limitations. The lumigraph uses an approximation surface (volumetric visual hull) to

act as a proxy whenever possible to improve the synthesised view[34], and was later extended to work without the explicit reconstruction[11].

Surface light fields overcome the viewpoint constraint by assigning a colour value to every ray leaving every point on a surface. The field is constructed from hundreds of images of an object, and used to construct realistic images from arbitrary viewpoints (including inside the object's convex hull). Since the surface point light value is recorded, the surface light field preserves surface texture, specularity and global effects such as inter-reflection and shadowing[87]. While the overall number of images required is much less than light fields, it is still sufficiently high to make the capture of dynamic scenes impractical.

2.8 Image-based surface reconstruction

The approaches described above are generally reconstruction or image based, but methods exist which are hybrids of both. Matusik et al presented *image-based visual hulls*, a technique which constructed a view-dependent visual hull from fixed calibrated cameras, mostly in the image domain[56]. Real-time frame rates were achieved by converting the silhouettes into polygons and dividing them into small sections, thereby reducing the search space during visual hull construction. Visibility of a point on the surface with respect to the cameras was determined, and of those cameras the closest to the virtual camera was used to colour the point. The advantage of this view synthesis approach was its ability to accurately capture the shape of multiple objects in dynamic scenes, although the quality of the models suffers from visual artefacts due to visual hull and inexact computation.

Image based visual hulls efficiently generate a close approximation to the exact visual hull, producing more accurate geometry than other visual hull approaches such as voxel carving with similar computation time. The quality of boundary visual hulls is higher than this approach because every pixel on the silhouettes

is used for model generation and colour is accurately determined for each point, but they are not as efficent as image-based visual hulls and therefore not suitable for online applications.

The accuracy of the model geometry and the colour information for image-based visual hulls was improved using photo consistency to create the *image based photo hull*[72]. Initially the visual hull was created from the silhouettes, and scene geometry was refined based on colour information from the input images. The visual quality improved but the approach uses only pixel-wise colour refinement, and greater improvement could be achieved by using a stereo correspondence algorithm. The view-dependent system was used in an immersive teleconferencing application called *Coliseum*, which synthesised views of participants with the current user's position as the desired viewpoint[3]. Participants appeared in a virtual scene, communicating visually in addition to verbally. The algorithms developed for the image based visual hull view-dependent method were adapted to work in a view-independent context to produce complete models[55] and also in graphics hardware to render at above frame-rate[52].

Fitzgibbon et al. [27] presented an image-based rendering approach which attempts to reconstruct colour instead of depth for a given pixel in a virtual view, which can avoid the artefacts associated with depth reconstruction via colour comparison in regions of similar colour. From the initial generated view a second operation is performed to identify pixels inconsistent to their surroundings, using the original images to constrain the possible outcomes. The images produced are of a high quality, however the technique requires a large number of input images and suffers from artefacts where regions of a synthesised image are unusual (such as object corners or hair).

Cross et al. [21] developed a system to reconstruct geometry of static objects with smooth surfaces using a moving video camera's images as input. This system calibrated the cameras as part of the process of reconstruction. This technique

was only applied to static objects and requires many images to construct the 3D textured model.

The technique presented in Chapter 3 has a similar basis to image based visual hulls in its approach, but improves the quality and accuracy of the model by using every pixel on the boundary of the silhouettes (instead of a polygon representation) and introducing several novel techniques for visual hull construction.

2.9 Conclusion

This section has presented the various methods used to synthesise novel viewpoints of a scene. Shape reconstruction based approaches are best suited to dynamic scenes and arbitrary viewpoints, but suffer from lower quality and visual inconsistencies across time. Image-based approaches are highly realistic yet do not lend themselves to dynamic scenes, and viewpoint locations are often constrained.

The approach taken in this research is a hybrid approach between shape reconstruction and image-based rendering. By using the original images in the final result and minimising the number of resampling steps in the synthesis process the quality of the novel rendered views should increase.

Chapter 3

Exact View-Dependent Visual Hulls

"O wad some power the giftie gie us to see oursels as ithers see us!"

Robert Burns, *To A Louse*

This chapter introduces a novel method for visual hull construction which produces samples on the visual hull surface from a specific viewpoint using images from multiple views. Efficient construction is achieved by performing computation in the image domain. This enables efficient computation of the exact visual hull surface visible from a virtual viewpoint from a set of silhouette images of a scene.

The objective of this research is to design an efficient visual hull algorithm to produce high quality surface rendering comparable to the quality of the original images. The algorithm described here exploits projective geometry to compute the surface in the image domain, which has several advantages:

- Computation in 2D is more efficient

- The scale of the scene is not important
- Processing is highly parallelisable

The novel contributions of this chapter are as follows:

1. Exact algorithm : there are no approximations (samples are on the visual hull surface, as accurately as matting, calibration and image resolution allow)
2. Efficient point selection for view-dependent visual hull
3. Efficient ordering process using projective invariants to evaluate surface sampling
4. An algorithm which computes the visibility to the same resolution as the surface

The novel contributions of items 1 - 3 were published in *Exact View-Dependent Visual Hulls*, International Conference on Pattern Recognition, 2006 [58].

3.1 Visual Hull

The technique presented here, and those in subsequent chapters, make a number of assumptions about the acquisition system used:

1. The region of interest (foreground) can be separated from everything else (background).
2. The cameras used to capture images are calibrated so that internal parameters are known for each camera. The position and orientation of each camera is also known.

3. For dynamic scenes, the time at which all cameras capture an image is synchronised to within a delay ϵ such that $\epsilon \ll s$, where s is the exposure time.

Given N calibrated views, the set of captured images $\mathcal{I} = \{\mathcal{I}_n : n = 1, \ldots, N\}$ is processed to produce the set of silhouettes $\mathcal{S} = \{\mathcal{S}_n : n = 1, \ldots, N\}$ via foreground extraction. $\mathcal{S}_n$ is the set of all pixels in the foreground region of $\mathcal{I}_n$. The *silhouette cone* for the n^{th} view is produced by back-projecting rays from the camera centre $\boldsymbol{c}_n$ through the foreground pixels in the silhouette $\mathcal{S}_n$. The visual hull is the three dimensional shape formed by the intersection of all views' silhouette cones[47].

3.2 Overview

The exact view-dependent visual hull (VDVH) is an algorithm for finding the intersection of the silhouette cones in the image domain and returning a depth map of the visible visual hull surface with respect to a virtual view. The silhouettes are processed to produce ordered sets of pixels on the boundary of the foreground and these sets are subdivided into indexable lists for efficient access. This subdivision is performed by constructing a set of bins in the image, based on angle to the epipole (shown in Figure 3.4), and populating the bins with pointers to the boundary point where the boundary begins. A ray is cast for each pixel in the virtual view and intersected with the silhouette cones from all views. This is performed in the image plane by projecting the ray onto the image to produce an epipolar line, and finding the intersection of the epipolar line with the boundary pixels from the silhouette. The relevant sections of the boundary are retrieved from the relevant bin and intersected with the epipolar line. The cross ratio is used to define a consistent ordering of intersections on the ray across images, and

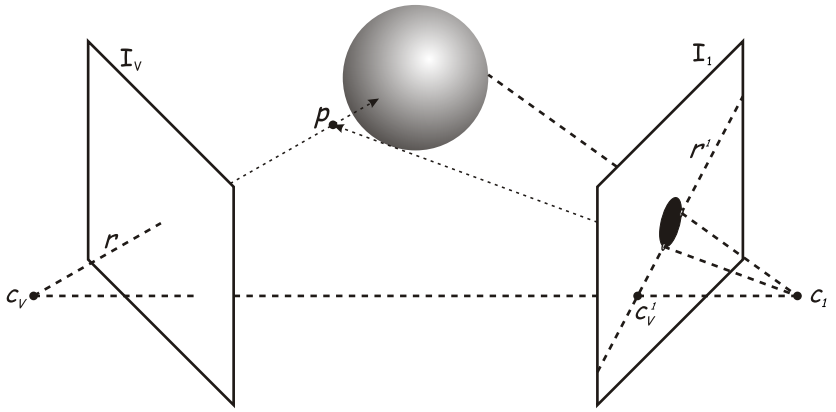


Figure 3.1: The ray $\boldsymbol{r}$ is projected onto $\mathcal{I}_1$ to give $\boldsymbol{r}^1$, the epipolar line. Rays are cast from $\boldsymbol{c}_1$ through intersections between $\boldsymbol{r}^1$ and the silhouette boundary. These rays are triangulated with $\boldsymbol{r}$ to find the points on the visual hull.

an efficient counting process is used on the sorted intersections to select the one which corresponds to the visual hull surface.

3.3 Single View Visual Hull Intersection

This section will explain the fundamental idea underlying visual hull construction, and the representation of the silhouettes which improves the efficiency of the process. The VDVH construction process is first presented for a single view to simplify the explanation, and subsequently extended to an arbitrary number of views.

Consider the case of a single image $\mathcal{I}_1$ and corresponding silhouette $\mathcal{S}_1$ whose occupied pixels represent the foreground in the scene. Let $\mathcal{I}_v$ be the virtual image for which the VDVH is to be constructed, $\boldsymbol{c}_v$ be the virtual camera centre for $\mathcal{I}_v$, and $\mathcal{R}_v = \{\boldsymbol{r} = P_v^{-1}\boldsymbol{u}, \boldsymbol{u} \in \mathcal{I}_v\}$ be the set of rays projected from $\boldsymbol{c}_v$ through the pixel centres in $\mathcal{I}_v$. Then the visual hull for $\mathcal{I}_v$ results from the intersection of

$\mathcal{R}_v$ with the silhouette cone from $\boldsymbol{c}_1$ through $\mathcal{S}_1$. Equivalently this is defined by the two view projective geometry illustrated in Figure 3.1, where the intersection can be performed in the image plane.

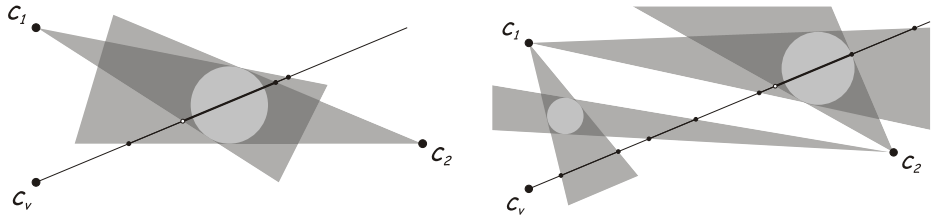
The intersection is performed in the image plane by projecting $\boldsymbol{r} \in \mathcal{R}_v$ onto $\mathcal{I}_1$ to produce $\boldsymbol{r}^1$, a two dimensional line in the image plane of $\mathcal{I}_1$ passing through the epipole $\boldsymbol{c}_v^1 = P_1 \boldsymbol{c}_v$, shown in Figure 3.1. The intersection of $\boldsymbol{r}^1$ with the contour of $\mathcal{S}_1$ produces the set of points $U_1 = \{\mu_k^1 \in \mathbb{R}^2 : k = 1, \ldots, K\}$ ordered along $\boldsymbol{r}^1$ starting from $\boldsymbol{c}_v^1$, where K is the number of intersections. The three dimensional points on the visual hull surface can be recovered by finding the intersection of the rays cast from $\boldsymbol{c}_1$ through μ_k^1 with $\boldsymbol{r}, \forall k \in K$.

For the special case in the image domain where the epipole is inside the silhouette, the first silhouette intersection on the epipolar line after the epipole is removed. This follows from the assumption that all intersections must be in front of the camera (in other words objects are not behind the camera and the camera is not inside an object).

For the VDVH the important point to identify is the one which is visible in the current view. The intersections where the epipolar line enters the silhouette correspond to possible visible surface points. More formally, for a point to be visible the following condition must be satisfied:

Observation 3.1. For a silhouette intersection $\mu_k^1 \in U^1$ to correspond to an intersection of ray $\boldsymbol{r}$ with a visible part of the visual hull surface the intersection number k along the epipolar line $\boldsymbol{r}^1$ must be odd.

Proof. Visible surface must have its surface normal pointing towards the viewing camera, and therefore the intersection corresponding to visible surface must be when the epipolar line enters the silhouette. Since there must be an even number of intersections on the line (the ray must enter and leave the surface) and



(a) Single object (b) Two objects

Figure 3.2: Cross-section of the silhouette intersections along a virtual camera ray with centre of projection $\boldsymbol{c}_v$ with silhouette images for two cameras with centres of projection $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$. The first visible intersection point on the visual hull surface marked as an **o** on both images.

numbering the intersections beginning at 1, the visible intersections correspond to their number being odd. □

This condition guarantees that the intersection point μ_k is visible (the surface normal points towards the camera viewpoint).

For the single view case the first intersection of the virtual camera ray $\boldsymbol{r}$ with the visual hull surface is given by the point $\boldsymbol{p}$ on $\boldsymbol{r}$ corresponding to the first intersection μ_1 of the epipolar line $\boldsymbol{r}^1$ with the silhouette boundary. The point can be represented by a depth d from the camera centre along the ray such that $\boldsymbol{p} = \boldsymbol{c}_v + d\boldsymbol{r}$. Given a point μ on the epipolar line $\boldsymbol{r}^1$ there is a corresponding point $\boldsymbol{p}(\mu)$ on the ray $\boldsymbol{r}$ with depth $d(\mu)$ which is computed by finding the intersection of the ray through μ from $\boldsymbol{c}_1$ with $\boldsymbol{r}$.

The exact VDVH for a single view is given by the first silhouette intersection on the epipolar line of every ray through the virtual image $\mathcal{I}_v$, which can be represented as a depth map D_v. The depth map is an image made up of depth elements, *dexels*, each of which holds a single depth to the visual hull surface from the camera centre. Dexels whose rays do not intersect the surface are set to zero.

3.4 Multiple View Intersection Selection

Following the case of a single view, the case with an arbitrary number of views is now considered. Given a set of images $\mathcal{I}$ and silhouettes $\mathcal{S}$ (as previously defined), each is treated individually in exactly the same way as for the single view case. Each image has an ordered set of silhouette intersections associated with the ray $\boldsymbol{r}$ through the virtual view $\mathcal{I}_v$. $\boldsymbol{r}$ is projected onto $\mathcal{I}_n$ to give the epipolar line $\boldsymbol{r}^n$ and intersected with $\mathcal{S}_n$ to give $U_n = \{\mu_k^n : k = 1, \ldots, K_n\}$ for the n^{th} view.

Establishing the point which corresponds to the first point of intersection between $\boldsymbol{r}$ and the visual hull surface is more complex than for the single view case. Other visual hull techniques use an explicit interval intersection on $\boldsymbol{r}$ to find the sections occupied by the visual hull[56, 72]. All intersections of the projection of $\boldsymbol{r}$ with the silhouettes must be combined into a single ordered set U. The point on $\boldsymbol{r}$ is found for each intersection and a distance metric from $\boldsymbol{c}_v$ to this point is used to insert it in the correct position in U. This process can be done more efficiently in the image domain, as shown in the following section.

The silhouette intersection which corresponds to the first intersection of $\boldsymbol{r}$ with the visual hull surface is given by the following theorem:

Theorem 3.1. (Visual Hull Visible Intersection Theorem) *The silhouette intersection $\mu \in U$ corresponding to the first intersection of ray $\boldsymbol{r}$ with the visual hull surface is the first silhouette intersection which satisfies the condition that for each of the views there is an odd number of silhouette intersections on the projection of ray $\boldsymbol{r}$ from the virtual camera centre $\boldsymbol{c}_v$ up to and including the point $\boldsymbol{p}(\mu)$.*

Proof. If there is an even number of intersections for any view n on the line segment between $\boldsymbol{c}_v$ and $\boldsymbol{p}(\mu)$ then for the n^{th} view the projection of $\boldsymbol{p}(\mu)$ is observed as outside the silhouette corresponding to empty space. Consequently

if the projection of $\boldsymbol{p}(\mu)$ is not inside or on the silhouette for all views then it does not correspond to a point on the visual hull. Therefore the *visual hull visible intersection condition* (3.1) must be satisfied in all views for μ to be on the visual hull. This requires an odd number of silhouette intersections along the corresponding epipolar line r^n for all views. □

This can be seen intuitively from the previous observation that whenever a projected ray enters a silhouette, the number of this intersection must be odd. For an intersection to correspond to the visual hull surface, the ray $\boldsymbol{r}$ must have entered every silhouette and not exited, and therefore every view must have an odd number of intersections.

This gives a depth for the first intersection of the ray $\boldsymbol{r}$ with the visual hull surface. Figure 3.2 illustrates the silhouette intersections for a virtual camera ray with two silhouette images with multiple objects. The first visible intersection of the ray with the visual hull surface is the first point which is inside the silhouette for both camera views. This is given by an odd number of silhouette intersections for each camera view as stated in the *Visual Hull Visible Intersection Theorem.*

3.5 Ordering by Projective Invariant

The theorem introduced in the previous section states that for a set of images the exact intersection of a virtual camera ray $\boldsymbol{r}$ with the visual hull can be determined from the ordering of silhouette intersections for each view. In this section it is demonstrated how projective invariants can be used to evaluate the relative ordering of silhouette intersections for different views without explicit computation of the three dimensional points $\boldsymbol{p}(\mu), \mu \in U, \forall \mu$ along the ray. This allows computationally efficient evaluation of the exact intersection of each ray with the silhouette boundaries.

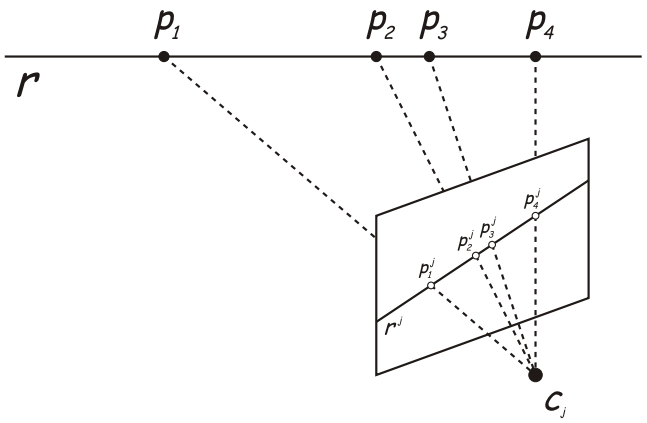


Figure 3.3: The cross ratio of $\boldsymbol{p}_{1-4}$ on $\boldsymbol{r}$ is equal to the cross ratio of $\boldsymbol{p}^j_{1-4} = P_j \boldsymbol{p}_{1-4}$ on $\boldsymbol{r}^j$ in the j^{th} view.

The *cross ratio* of four collinear points, $\boldsymbol{p}_{1-4}$, is the only invariant in projective geometry [40], and is defined by:

$$\chi(\boldsymbol{p}_{1-4}) = \frac{|\overrightarrow{\boldsymbol{p}_1\boldsymbol{p}_2}||\overrightarrow{\boldsymbol{p}_3\boldsymbol{p}_4}|}{|\overrightarrow{\boldsymbol{p}_1\boldsymbol{p}_3}||\overrightarrow{\boldsymbol{p}_2\boldsymbol{p}_4}|} \tag{3.1}$$

where

$$\overrightarrow{\boldsymbol{p}_k\boldsymbol{p}_l} = \boldsymbol{p}_l - \boldsymbol{p}_k$$

This leads to the key observation which allows silhouette intersection ordering to be processed in the image domain:

Observation 3.2. The cross ratio is constant across projection for the same set of points: given the collinear points $\boldsymbol{p}_{1-4} \in \mathbb{R}^3$ and their projections in $\mathcal{I}_n$, $\boldsymbol{p}^n_{1-4} \in \mathbb{R}^2$, then $\chi(\boldsymbol{p}_{1-4}) = \chi(\boldsymbol{p}^n_{1-4})$

This property is illustrated in Figure 3.3, and can be exploited to order silhouette intersections along the virtual camera ray $\boldsymbol{r}$ by comparison of the cross ratio along the epipolar lines for different views.

To evaluate the cross ratio for a point on an epipolar line, three points are generated on $\boldsymbol{r}$ and projected onto all images. For example:

$$\begin{aligned} \boldsymbol{p}_1 &= \boldsymbol{c}_v - 2\boldsymbol{r'} \\ \boldsymbol{p}_2 &= \boldsymbol{c}_v - \boldsymbol{r'} \\ \boldsymbol{p}_3 &= \boldsymbol{c}_v \end{aligned} \tag{3.2}$$

where $\boldsymbol{r'} = \frac{\boldsymbol{r}}{|\boldsymbol{r}|}$ is the ray unit vector.

These common points are projected onto view n to obtain three points on the epipolar line, r^n: $\boldsymbol{p}_1^n$, $\boldsymbol{p}_2^n$ and $\boldsymbol{p}_3^n = c_v^n$. The cross ratio χ_k of the projected points with a silhouette intersection point $p_4^n = \mu_k^n$ is calculated from Equation 3.1, and used to sort the points implicitly by increasing distance from the camera centre.

Ordering of silhouette intersections U for multiple views along the virtual camera ray $\boldsymbol{r}$, using the cross ratio χ_k, is used to identify the silhouette intersection μ_k^n which corresponds to the first visible intersection with the visual hull surface. The corresponding point on the visual hull surface $\boldsymbol{p}(\mu_k^n)$ is reconstructed as the distance $d(\mu_k^n)$ along the ray from $\boldsymbol{c}_v$. $\boldsymbol{p}(\mu_k^n)$ is the exact intersection of the ray $\boldsymbol{r}$ with the visual hull surface, such that $\boldsymbol{p}(\mu_k^n) = \boldsymbol{c}_v + d(\mu_k^n)\boldsymbol{r}$. Repeating this process for virtual rays corresponding to each pixel in the virtual image, I_v, the exact view-dependent visual hull is obtained.

3.5.1 Efficiency Comparison

For every epipolar line which intersects the silhouette, the two distances $|\overrightarrow{\boldsymbol{p}_1^n\boldsymbol{p}_2^n}|$ and $|\overrightarrow{\boldsymbol{p}_1^n\boldsymbol{p}_3^n}|$ are precomputed and stored. Each intersection on the ray then requires two 2D distance computations ($|\overrightarrow{\boldsymbol{p}_3^n\boldsymbol{p}_4^n}|$ and $|\overrightarrow{\boldsymbol{p}_2^n\boldsymbol{p}_4^n}|$), then two multiplies and a divide to compute the cross ratio.

After the intersection corresponding to the visual hull surface has been selected, the distance from $\boldsymbol{c}_v$ to the point $\boldsymbol{p}_4$ must be computed. This is done by rearrang-

ing Equation 3.1 to find the distance from $\boldsymbol{p}_3$ to $\boldsymbol{p}_4$. Let $u = |\overrightarrow{\boldsymbol{p}_1^n \boldsymbol{p}_2^n}|$, $v = |\overrightarrow{\boldsymbol{p}_2^n \boldsymbol{p}_3^n}|$ and $w = |\overrightarrow{\boldsymbol{p}_3^n \boldsymbol{p}_4^n}|$, then Equation 3.1 becomes:

$$\chi = \frac{uw}{(u+v)(v+w)} \tag{3.3}$$

The points $\boldsymbol{p}_{1-3}$ in 3.2 were chosen to be unit distance apart, so substituting into Equation 3.3:

$$\chi = \frac{w}{2+2w} \tag{3.4}$$

and rearranging:

$$\begin{aligned} \frac{1}{\chi} &= \frac{2+2w}{w}, \qquad \chi, w \neq 0 \\ &= \frac{2}{w} + 2 \\ \Rightarrow \qquad w &= \frac{1}{\frac{1}{2\chi} - 1} \\ &= \frac{\chi}{\frac{1}{2} - \chi} \end{aligned} \tag{3.5}$$

Therefore given the cross ratio for a silhouette intersection the distance to the point can be computed using one subtract and one divide.

The cost per intersection for computing the distance via triangulation is one 3×3 matrix multiplication to find the ray through the silhouette intersection, a triangulation of the two rays and then computing $|\overrightarrow{\boldsymbol{p}_3 \boldsymbol{p}_4}|$.

The cost using the cross ratio method requires two 4×3 matrix multiplications to find $\boldsymbol{p}_{1,2}^n$ ($\boldsymbol{p}_3^n$ is the epipole, which is computed once for each image) and two 2D distance operations to find $|\overrightarrow{\boldsymbol{p}_1^n \boldsymbol{p}_2^n}|$ and $|\overrightarrow{\boldsymbol{p}_1^n \boldsymbol{p}_3^n}|$ per epipolar line. Then the cost per intersection is two 2D distance operations, two multiplies and a divide. The cost of computing the 3D distance for the selected point is one subtract and one divide using Equation 3.5.

If an epipolar line intersects the silhouette boundary, there must be a minimum of two intersections. For triangulation, the cost of two intersections is two 3×3

matrix multiplications, two line triangulations and two 3D distance operations. For the cross ratio, the cost is two 4×3 matrix multiplications, six 2D distance operations, four multiplies, three divides and one subtract. Since triangulation requires a number of operations (such as dot and vector products), the method of computation presented here is more efficient.

For every additional pair of intersections on an epipolar line the efficiency increases in comparison to triangulation since only an incremental computation is required for each additional intersection. Visual hull construction of a virtual view for the capture in Figure 3.7 had an average of 2.8 intersections per epipolar line (discounting those that do not intersect the silhouette), with a resulting 9.7% decrease in construction time, and so the ordering by a projective invariant using 2D computation provides a more efficient approach.

3.6 Efficient Implementation

The silhouettes are pre-processed to increase the efficiency of contour-line intersection by representing them as a set of contours (ordered lists of pixels on the boundary of $\mathcal{S}_n$), and splitting these contours into smaller, indexed, sections. This allows the efficient use of every pixel on the boundary, unlike other approaches which approximate the silhouette with piecewise linear segments[56, 30], and an exact sampling of the visual hull surface is produced.

Finding the intersections between the boundary of a silhouette and a line has a complexity of $O(n)$, assuming an image with $O(n^2)$ pixels and a silhouette boundary proportional to the size of the image perimeter. Processing all pixels in the virtual image on every original image would be computationally very costly ($O(sn^3)$, where s is the number of images). This section demonstrates that the efficiency of the process is improved by inserting the boundary points into an ordered list, subdividing this into smaller indexed lists, reducing the complexity

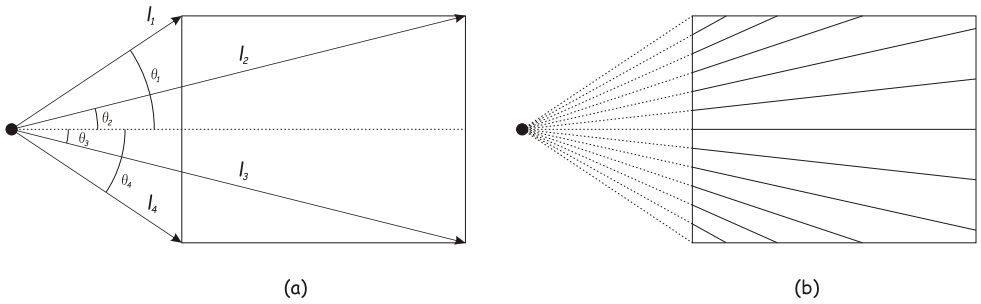


Figure 3.4: The largest difference in angle between l_{1-4} is required to construct the bins. In this case, l_1 and l_4 in (a) have the largest difference, and the bins are constructed between them, resulting in the structure in (b).

of cone intersections to $O(sn^2i)$ (i is the average number of intersections of a ray with the silhouette cone, $i \ll n$)[56]. The remainder of this section describes one way of achieving this, by dividing the silhouette image into a number of equal size bins. Note that in the following description all angles $\theta \in (-\pi, \pi]$, $\theta = 0$ represents the positive x-axis and $\theta > 0$ lies in positive y-space.

The bins use the epipole of the virtual camera as a base, and are indexed by angle using the x-axis as $\theta = 0$. The corners of the image are connected to the epipole via four lines, l_{1-4}, shown in Figure 3.4a. The lines that deviate most from the horizontal line through the image are set as top and bottom of the set of bins (l_1 and l_4 in the figure). The difference in angle from one bin edge to the next, θ_{incr}, is the angle between these two lines ($\theta_{diff} = \theta_1 - \theta_4$) divided by the total number of bins. This number is set arbitrarily, but could be linked to image resolution. Starting from l_1, the bins' edges are constructed by creating a line (of the form $ax + by + c = 0$) from the epipole every θ_{incr} radians, until l_4 is reached. An illustration of the final data structure is shown in Figure 3.4b. The correct bin for an epipolar line is the result of $\frac{\theta - \theta_4}{\theta_{diff}} N$, where θ is the angle between the epipolar line and the x-axis and N is the number of bins.

In order to efficiently find points where the silhouette boundary meets the epipolar

line, the bin edges must be intersected with the boundary. At the visual hull construction stage this will provide the pixels from which to start during silhoutte cone intersection. The intersections can be located by traversing the boundary pixel list, creating an epipolar line by connecting the current pixel to the epipole, calculating which bin the current pixel should be in and comparing this to the previous pixel. When a change in bin occurs, a pointer to the pixel before the change occurred is saved as a starting point for intersection testing.

During visual hull construction the intersection test is performed using the epipolar line produced by the projection of a ray through the virtual image. Its corresponding bin is accessed and the line is tested against all the boundary sections in the bin. The test is performed by iterating along boundary segments and checking which side of the line the current pixel lies (for a pixel with coordinates (u, v), the sign of $au + bv + c$, where a, b and c are the parameters of the line, indicates which side it is on). When the sign changes the pixel closest to the epipolar line is chosen. This could be improved by constructing a line using the current and previous pixels and finding the intersection of this line with the epipolar line to produce the silhouette intersection.

3.7 Visual Hull

The previous sections presented a novel method of producing the exact VDVH, utilising the cross ratio for efficient ordering of intersections, and the *Visual Hull Visible Intersection Theorem* for selecting the correct intersection. This section extends the approach to efficiently produce the full exact visual hull from a set of silhouette images. Following the same approach as before, the visual hull is constructed with respect to an arbitrarily chosen viewpoint. This allows us to use the same efficient framework as for VDVH construction. The VDVH is now extended to represent the full visual hull with respect to a specific viewpoint.

3.7.1 Extending intersection selection

Construction of the full visual hull follows the same steps as VDVH construction up to the intersection selection. At this point we have a set of ordered silhouette intersections $\{\boldsymbol{U}^n\}_{n=1}^{\boldsymbol{N}}$ from images $\mathcal{I}$ corresponding to points on the virtual camera ray $\boldsymbol{r}$. Observation 3.1 and Theorem 3.1 have shown that an intersection corresponding to a visible surface point on the visual hull can be identified by counting the number of silhouette intersections for each view. This intersection is the point at which $\boldsymbol{r}$ has entered all silhouettes. The following theorem extends Theorem 3.1 to find all points on $\boldsymbol{r}$ corresponding to intersections with the visual hull surface, not just the visible surface.

Theorem 3.2. (Visual Hull Intersection Theorem)

(i) Theorem 3.1 provides a method of identifying the first visible intersection of the ray $\boldsymbol{r}$ with the visual hull surface. The condition in Theorem 3.1 can be applied to select any front-facing intersection $\mu \in \{\boldsymbol{U}^n\}_{n=1}^{\boldsymbol{N}}$ (a visual hull surface point whose normal points towards the virtual view).

(ii) The silhouette intersection immediately after μ corresponds to the next visual hull surface point on $\boldsymbol{r}$. This surface point has a normal which points away from the virtual view.

Proof.

(i) Since the front-facing points correspond to when the ray $\boldsymbol{r}$ enters all silhouettes, the proof from Theorem 3.1 can be applied to provide the result.

(ii) The intersection immediately after μ corresponds to the ray leaving a silhouette on one of the images. Therefore that intersection was the last point inside all silhouettes and so belongs to the visual hull. □

The full visual hull requires an alternative representation to the view-dependent depth map. The extended representation is a depth map with multiple layers whose elements, *depthels*, contain an ordered set of real values representing depth from the camera centre. Depthels which do not represent surface are empty. The entries of a depthel are the result of the intersection of the ray through its pixel in the depth image with the constructed surface (in this case the visual hull), and each depthel is independent of its neighbours. This new representation will be referred to as a multi-layer depth map in the remainder of the thesis.

This representation of the full visual hull has not been designed for use as a final model but rather as a basis for further work, therefore a method of producing a triangulated mesh has not been formulated. The information contained within the full visual hull allows us to compute visibility on the visual hull surface which allows more accurate colouring or refinement than a technique which did not take visibility into account. This representation is also the basis for further surface reconstruction work, described in Chapter 6.

3.8 Visibility

Visibility information allows surfaces to be textured or refined more accurately, leading to fewer artefacts. For a given surface point $\boldsymbol{x}$, camera $\boldsymbol{c}_n$ and corresponding image $\mathcal{I}_n$, if $\boldsymbol{x}$ is not visible to $\boldsymbol{c}_n$ then the colour information in $\mathcal{I}_n$ should not be used for texture or refinement operations. Previous methods[56, 72] for finding visibility of an image-based visual hull use inaccurate computation, whereas the approach presented here constructs per-view exact visibility maps of the visual hull surface. Each visibility map is represented as a multi-layer depth map with respect to the original view, to show how much of the surface is visible from that view.

The visibility for a surface point corresponding to a pixel in the image is computed

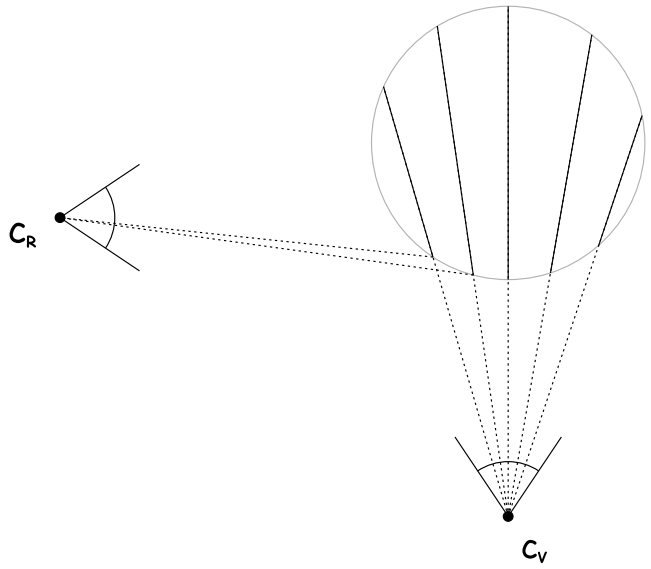


Figure 3.5: Cross section view of the iterative visibility approach. Computation starts from the left (using the location of the real camera), and proceeds to the right. The left-most interval is projected onto the next interval using the real camera as a reference, and subtracted from the second interval. If any of the interval remains (as it does in this case) then this surface point is visible to the real camera. The process is repeated for each interval.

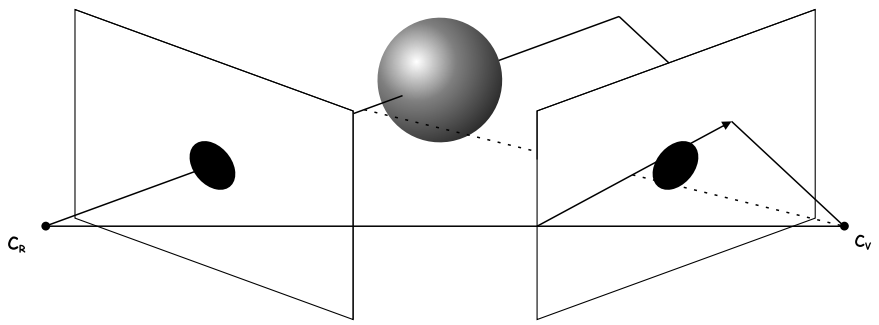


Figure 3.6: The visibility plane constructed for a particular epipolar line in the virtual view when computing the visibility of the surface with respect to a real view. The dotted line represents the ray from the virtual camera where the visibility computation starts. The ray is swept along the plane, using intervals inside the surface to update the visibility information for each virtual view pixel. The result is a depth map which identifies the regions of the surface, which has been constructed with respect to the virtual camera, visible to the real camera.

by constructing a per-ray occlusion map. Consider the case of a single pixel $\boldsymbol{p}$ in the virtual image $\mathcal{I}_v$, with corresponding ray $\boldsymbol{r}$ through $\boldsymbol{p}$ from the virtual camera centre $\boldsymbol{c}_v$. Let $\mathcal{X}$ be the set of intervals on $\boldsymbol{r}$ representing the segments where $\boldsymbol{r}$ is inside the visual hull surface (in the example in Figure 3.6 there would only be one interval). Then the visibility algorithm will find the visible portions of $\mathcal{X}$ with respect to another camera $\boldsymbol{c}_n$, in other words those not occluded by other surfaces.

The visibility for $\mathcal{X}$ can be computed in three dimensions by first constructing a plane which contains $\boldsymbol{r}$ and $\boldsymbol{c}_n$, finding all surface regions intersected by this plane between $\boldsymbol{r}$ and $\boldsymbol{c}_n$, and projecting these onto $\boldsymbol{r}$ from $\boldsymbol{c}_n$ to create a new set of intervals $\mathcal{X}'$. The visibility for the surface at pixel $\boldsymbol{p}$ is:

$$vis(\boldsymbol{p}) = \mathcal{X} - (\mathcal{X} \cap \mathcal{X}') \tag{3.6}$$

This equation represents the occluding intervals prior to the current interval being

subtracted and the remainder is the visible portion of the surface. Figure 3.5 shows the rays projected from the virtual camera, and the intervals inside the surface. To compute the visibility the algorithm iterates across the plane (cross-section shown) from the left-most interval (in the general case processing starts at the epipole of the real camera in the virtual image). The visibility of the second interval is the result of projecting the first interval onto the second using $\boldsymbol{c}_r$ as a reference. In this case the second interval is visible to the real camera, but the third would not be.

Visibility for $\mathcal{X}$ is computed using multi-layer depth maps and projective geometry. The plane is implicitly constructed by creating a line from the epipole $\boldsymbol{c}_n^v$ to $\boldsymbol{p}$ and iterating along this line and updating $\mathcal{X}'$. The epipolar line is shown in the virtual view (right) in Figure 3.6 and the dotted line shows the current ray being processed. At every occupied pixel on the line, points on the ray through this pixel are computed using the depths in the depth map, and the projection of these points onto $\boldsymbol{r}$ from $\boldsymbol{c}_n$ are calculated. The union of this new set of intervals with the previous set produces the new occlusion map on $\boldsymbol{r}$. The process is continued up until the pixel before $\boldsymbol{p}$ on the line. Then the exact visibility map for this ray is computed using the equation above. By advancing along the epipolar line from the epipole towards the surface, any occlusions of the surface will be encountered because this represents the viewing angle of the other camera in the virtual camera's image plane.

The exact visibility map is an advantage over previous techniques such as that used in image-based visual hull[56], which used an interval splatting method for visibility, or the plane-sweep approach[46] which has a regular sampling of the scene (coinciding with voxel space).

Assuming the depth map has $O(n^2)$ occupied pixels after reconstruction, and each pixel is checked separately for visibility, the cost of the algorithm is $O(svn^2)$, where v is the cost of visibility computation per pixel.

This approach is limited to visibility of the object seen by the current view. If any other objects exist in the scene not visible to the current camera, they are not taken into account. It also requires a method of traversing a line in an image, from epipole to current pixel. This line can either be made conservative (check every pixel the theoretical line touches) or approximate (using an algorithm such as Bresenham).

3.9 Reference View-Dependent Visual Hull

There is a special case for construction of the VDVH with respect to a real view. This can be useful for generating a depth map for an existing image, establishing an approximate depth for every foreground pixel in the image.

The real view $\mathcal{I}_R$ is set as the virtual view, and the real view's camera and silhouette are removed from the visual hull construction process. This is referred to as *reference view VDVH* throughout this work. Instead the search space of the virtual image for visual hull surface intersection is reduced by using the silhouette for this view as a mask. Only occupied pixels in the silhouette have rays cast from the camera centre. A comparison of computation time taken for reference VDVH and virtual VDVH is presented in Section 3.11.

Due to calibration and matting errors, all occupied pixels in the real view's silhouette will not necessarily have a surface depth. With perfect calibration and matting all silhouette pixels would correspond to points on the visual hull surface.

3.10 Surface Construction

This section describes how to construct a triangulated mesh of a multi-layer depth map, using information from the layers to identify depth discontinuities.

The vertices of the mesh are constructed by projecting a ray out through each pixel and finding the point on the ray that lies at the first depth stored in the depthel for that pixel. Since a depth map is an image, a triangle strip can be constructed along two rows of pixels (with edges removed for pixels that do not contain surface information) and repeated vertically for the entire image. However, depth discontinuities must be identified so that triangles are not created over occlusions.

Each triangle is constructed using the pixels in the image as a basis and the vertices are constructed as the first depth of the depthels at these pixels. The first intervals of these depthels are extracted and tested to see if they overlap: if they do the triangle is accepted as part of the same surface; if they do not overlap then a new vertex is created to avoid holes in the mesh. If all three intervals exclusively do not overlap each other, the triangle is rejected, however this is a rare case. The general case is where two of the intervals overlap and the third does not. In this case a new vertex is created at the same depth as the first two but on the same ray as the third, to maintain the continuity of the mesh.

The vertices of the mesh are given texture coordinates corresponding to the image coordinates in the depth maps. This method of mesh construction would not be possible without reconstruction of the full visual hull. When using a depth map composed of dexels a thresholding technique would be required to identify discontinuities.

3.11 Results

This section presents results and evaluation of the view-dependent visual hull technique for surface reconstruction from multiple views. Three different acquisition systems were used for testing:

Setup 1 Ten equally spaced cameras in an approximate circle of radius $4m$, baseline 36°, each capturing at 25Hz SD resolution (720×576) progressive scan. The original images from a capture from this setup can be seen in Figure 3.7.

Setup 2 Eight cameras in total, seven in an arc of 120° pointed towards the subject approximately $4m$ away. The eighth camera supplies a view from above. This setup uses the same cameras as Setup 1. The original images from a capture from this setup can be seen in Figure 3.12.

Setup 3 This is a synthetic setup comprising ten virtual cameras in a ring around the model. The images were rendered at SD resolution. The silhouettes from this setup can be seen in Figure 3.18.

For setups 1 and 2 intrinsic camera parameters were estimated in both cases using the public domain calibration toolbox [7]. Camera calibration gives a maximum reprojection error of 1.6 pixels (0.6rms) averaged across the cameras which is equivalent to a reconstruction error in the order of 10mm at the centre of the volume. The calibration for setup 3 was defined manually. All tests were performed on an AMD 3100+ Sempron with 2GB RAM and results rendered using OpenGL on an nVidia 6600 graphics card.

3.11.1 Surface Construction

The original images for a frame of a dynamic capture with the corresponding silhouettes are shown in Figures 3.7 and 3.8. The silhouettes are retrieved via a combination of background subtraction and chroma key techniques. The rendered surfaces shown in Figure 3.9 are VDVH reconstructions performed with respect to the real viewpoints, which gives an image and depth representation for every view. The surfaces in Figure 3.10 are VDVH reconstructions performed with respect

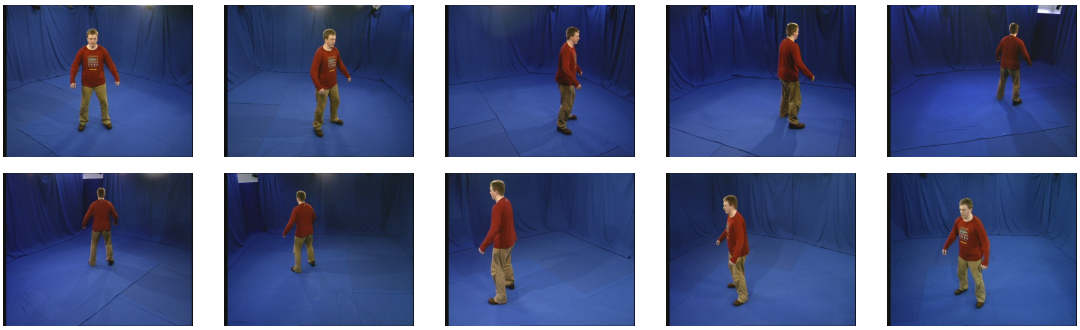


Figure 3.7: The original images from a single frame of a studio capture against a blue screen, and the corresponding silhouettes for extracted using background subtraction and chroma keying

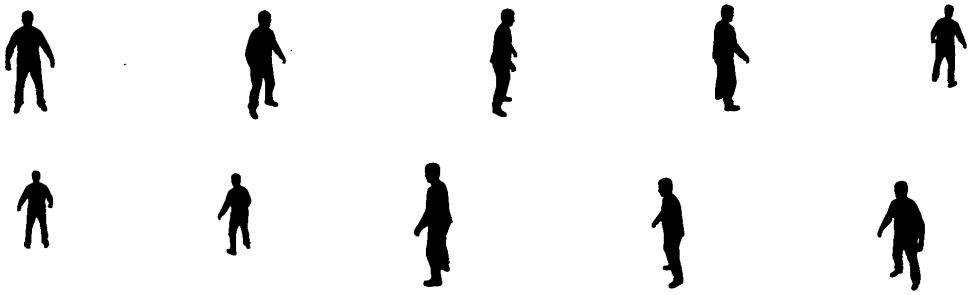


Figure 3.8: The corresponding silhouettes for Figure 3.7 for extracted using background subtraction and chroma keying

to virtual views; each virtual view is at the midpoint between two adjacent real views.

The rendered meshes are not smoothed and represent the exact visual hull surface given the input images and the calibration. The surfaces have regions which appear to be discontinuous; these are due to the discretisation of the images, and the change in direction of the contour of the silhouettes. The surface protrudes slightly in the chest region due to the arms being slightly in front of the body, therefore the silhouettes do not provide a clear view of the front of the body. A camera directly above the subject would help in this case, although if the subject's head were leaning forward this would also obscure the front. The following two chapters present methods to refine the surface to reduce the appearance of artefacts such as this.

The visual hull is affected by the quality of the silhouettes (i.e. the matting) and the calibration. If the calibration for a single camera is incorrect, parts of the surface are mistakenly removed and can cause the visual hull to be smaller than the original object. Even for accurately calibrated scenes used in these captures, the cumulative error of all cameras can cause slight reductions in the size of the visual hull when compared to the original silhouettes. The same error is caused by poor image segmentation, since if part of the real surface is not represented in the silhouette it will be removed from the reconstruction.

The table in Figure 3.11 shows how long surface construction takes for each view. The time taken for reconstruction with respect to a real viewpoint is much less than for a virtual viewpoint because the silhouette for the view acts as a mask of where to search for surface. Construction of a virtual view requires checking every pixel in the virtual image. It also shows the 9.7% increase in efficiency by using the cross ratio instead of triangulation for VDVH reconstruction.

The images shown in Figure 3.12 are from a single frame of a capture using Setup 2. The reconstruction of the VDVH with respect to the real views is shown in

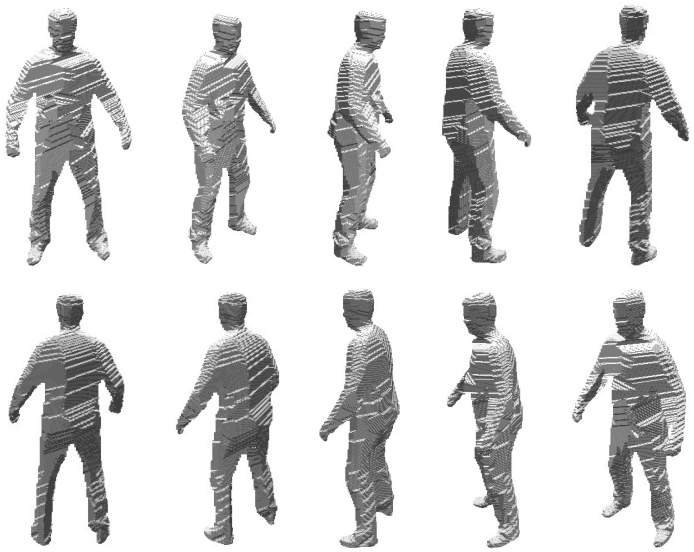


Figure 3.9: The VDVH reconstruction with respect to each original view (cropped here to show more detail), rendered as a flat shaded mesh. This representation produces a depth per pixel for the original image.

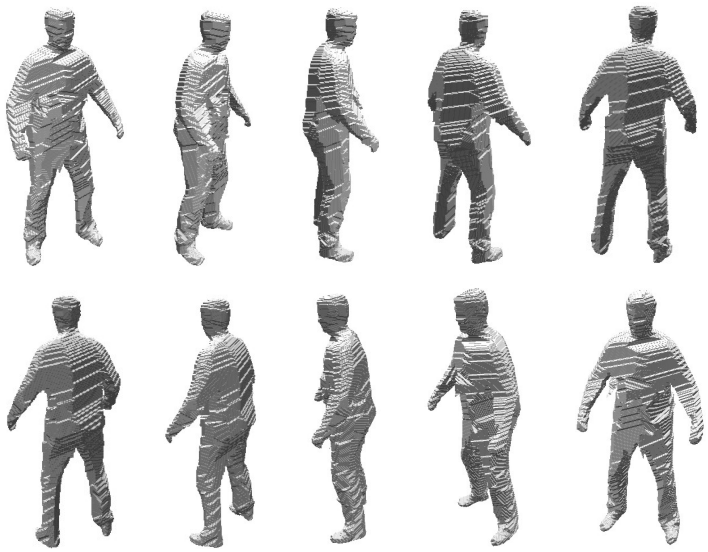


Figure 3.10: The VDVH reconstruction with respect to virtual views, each view at the midpoint between two real views.

View	Reference VDVH	VDVH	VDVH with Triangulation	% reduction
1	2.203	13.64	15.75	13.4
2	2.469	17.625	19.5	9.62
3	2.141	19.204	20.953	8.35
4	2.578	18.109	19.937	9.17
5	1.844	15.156	16.875	10.19
6	1.766	16.688	18.204	8.33
7	1.828	16.656	18.328	9.12
8	2.984	17.734	19.469	8.91
9	2.500	20.093	22.61	11.13
10	3.047	19.228	21.219	9.38

Figure 3.11: The time taken (in seconds) to perform the VDVH reconstructions shown in Figures 3.9 and 3.10 using the images in Figure 3.7. The last column shows the figures for time taken when using triangulation and not the more efficient cross ratio method to order the intersections.

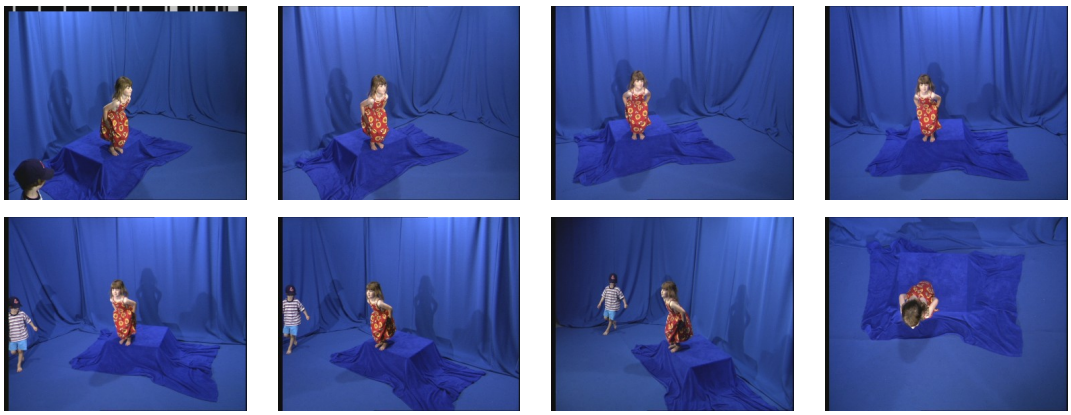


Figure 3.12: The original images from a single frame of a studio capture against a blue screen.

Figure 3.13. The overall surface shape reflects the subject, especially the head and the arms, however the torso is not properly represented. The visual hull is not capable of representing surface concavities, and as the original images show the dress around the torso is a concavity in the surface. Refinement techniques are presented in the subsequent chapters to reduce the artefacts associated with these surface regions.

The time taken to perform reconstructions for real and virtual views is shown in Figure 3.14. For this capture the reconstruction time with respect to real views is faster than for reconstruction of Figure 3.9 because the size of the subject is smaller in the image, therefore the search space is reduced. The time to reconstruct virtual views is slightly less due to this being an eight camera setup, while the other capture used ten cameras.

The tests were performed on a single computer with one CPU. For real-time implementation without sacrificing quality each camera could have its own computer for processing, and then a central computer to merge the views' information together. Alternatively an approach using programmable shaders in graphics hardware could be considered.

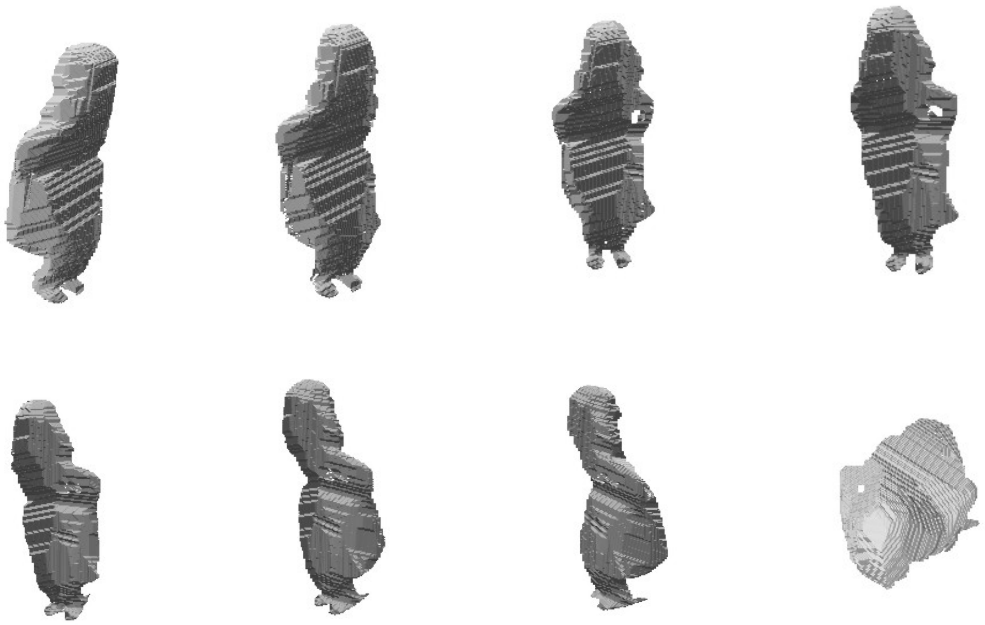


Figure 3.13: The VDVH reconstruction with respect to each original view, rendered as a shaded mesh.

View	Reference VDVH	VDVH
1	2.031	15.531
2	1.390	16.625
3	1.454	17.078
4	1.312	17.797
5	1.046	17.625
6	1.640	16.735
7	1.234	13.250
8	1.297	15.125

Figure 3.14: The time taken (in seconds) to perform the VDVH reconstruction shown in Figure 3.13 and also for a virtual view reconstruction, using the images in Figure 3.12.

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

Figure 3.15: Results of visibility computation on colouring of a virtual view VDVH reconstruction using the two adjacent real views. The texture of the right arm is incorrectly rendered onto the body in (a) and (c), but by using visibility information the colour of the surface is improved as shown in (b) and (d).

3.11.2 Visibility and Colouring

Visibility computation is important for virtual view reconstruction and colouring to reduce artefacts in the final result. Figure 3.15 shows results of rendering two virtual views with and without visibility information from Setup 1. The colour for a vertex is chosen by view-dependently rendering between the two closest views. The images in the figure without visibility information have an incorrectly textured body because the occlusion of the right arm has not been taken into account, whereas in the images with visibility the texture from the arm is not used on the body. Parts of the surface in the images which use visibility have not been coloured, because neither of the two adjacent real views observe this surface. In this case the colour from other cameras in the scene could be used to texture these regions.

3.11.3 Ground Truth Comparison

The goal of the surface reconstruction is to produce high quality novel views. Using setups 1 and 2, one of the real views is removed from processing and used to evaluate the result, by setting the real view as the target virtual view. Rendering the result to the virtual image and comparing it to the real captured image provides a measure of the quality of the view synthesis.

For setup 1, the first view is removed from the ring of ten cameras. The real cameras have a baseline of $36°$ and so the baseline of the target setup is $72°$. The colour for the synthesised view comes from the two adjacent views, blending using visibility information. In regions where neither camera has good visibility of the surface, the colour is blended from both to fill in the gaps (this could be improved by recovering colour from other cameras which have good visibility of these regions).

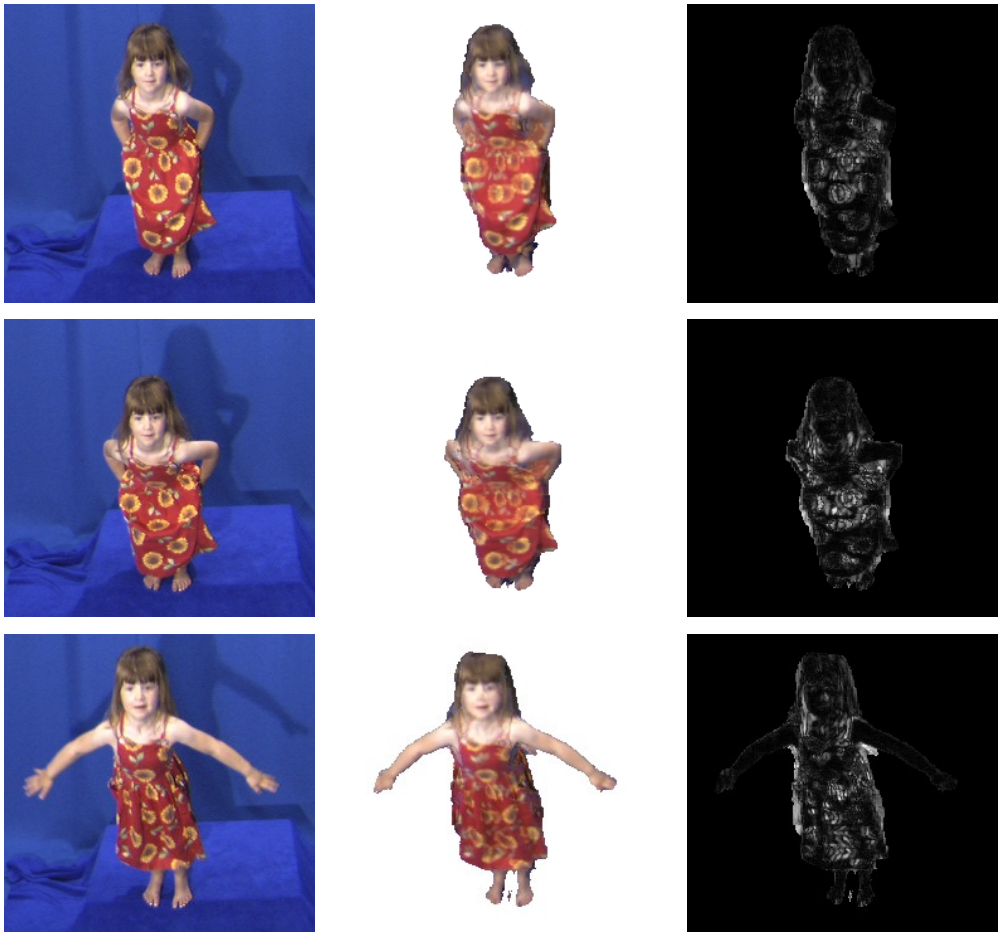
Figure 3.16 shows the results of the tests on setup 1. The errors in the surface and the colour are evident around the upper legs and under the arms, due to the lack of original views near the virtual view. The large baseline provides a difficult problem for the method, but the synthesised view still maintains colour close to the original view.

For setup 2, the central view of the arc of seven cameras (the fourth view in Figure 3.12) is removed for this comparison. The real cameras have a baseline of $20°$ between them, so the baseline between views three and five is $40°$. The VDVH is constructed using all seven remaining cameras, and the colour for the synthesised view comes from cameras three and five only.

Figure 3.17 shows a comparison between the captured image, the synthesised novel view via VDVH reconstruction and colouring using visibility, and the error image. The larger the error between the colour from the real and synthesised view, the higher the intensity of the pixel in the error image. The error is defined

(a) Original | (b) VDVH Reconstruction | (c) Error

Figure 3.16: The images above are three frames of a sequence from a camera which was removed from processing to be used as ground truth. (a) shows the original images, the synthesised view via VDVH is shown in (b), and the error intensity image is shown in (c).



(a) Original (b) VDVH Reconstruction (c) Error

Figure 3.17: The images above are three frames of a sequence from a camera which was removed from processing to be used as ground truth. (a) shows the original images, the synthesised view via VDVH is shown in (b), and the error intensity image is shown in (c).

as the distance between colours, using a Euclidean distance in RGB space.

The majority of the surface is sufficiently close to the original, although slightly blurred by the view-dependent rendering. Artefacts appear in regions where the surface does not represent the object correctly, mainly where concavities exist: the torso and the shoulders. The border of the surface also has a higher error and is often hard to colour correctly since this is the area of least sampling due to the use of depth images with respect to a particular view.

3.11.4 Ground Truth Surface Evaluation

This section evaluates the surface quality of the VDVH using a synthetic data set. A 3D model of a person created in a modelling program is used for the evaluation, since this provides an accurate depth per pixel which can be compared to the reconstructed depth per pixel.

The model used was taken from [85] and rendered in a custom OpenGL environment to ten views with known camera parameters and SD resolution. The rendered silhouettes are shown in Figure 3.18. The VDVH was constructed with respect to each real view, and a depth map produced for the real surface and the reconstructed surface. The real surface depth map is shown in the left column of Figures 3.19 and 3.20, the reconstructed VDVH depth map in the middle column, and the depth map for each view of a volumetric reconstruction is shown on the right.

The reconstructed surface produced closely represents the real surface. Errors occur in areas where the viewpoints produce ambiguous information, for example in views three and nine a protrusion from the chest is visible due to the arms coming slightly forward and occluding the chest from the side views. The visual hull also does not properly represent the upper leg regions, and produces artefacts where the arms join the torso.

A volumetric visual hull reconstruction was performed on the same data set to compare with the exact VDVH. Each voxel in the grid was approximately $5mm^3$ at the highest resolution (the reconstruction process uses an octree to increase efficiency). Since the data is synthetic and the model constructed in modelling software, the dimensions of the model are between 0 and 1. The approximate $5mm^3$ was established by assuming the character has a height of $1.8m$, and scaling the units appropriately. The volumetric visual hull is then smoothed and a mesh constructed using the marching cubes algorithm. A comparison of the volumetric and VDVH reconstructions is shown in Figure 3.22, using the ground truth model as a basis. The VDVH leaves detailed features intact, such as the holes in the hands in view 1, and the shape of the nose in view 3.

The error intensity images for the volumetric approach show a lighter shade over most of the reconstruction, indicating a larger error than the VDVH against the ground truth. The median errors of the VDVH reconstruction and the volumetric reconstruction are shown in Figure 3.21 (median used to avoid incorporating the errors due to artefacts which influence the average) and clearly show the benefit of using VDVH.

Aside from the artefacts associated with the visual hull, the reconstruction of the exact VDVH shows how close the visual hull surface is to the true surface. The VDVH approach can efficiently produce high quality visual hull depth maps from multiple views to the most accurate resolution possible given the resolution of the input images.

3.11.5 Computational Efficiency

Given a $n \times n$ virtual view, n^2 rays must be cast from the virtual camera centre and intersected with the silhouettes (also containing n^2 pixels). Assuming a silhouette boundary to have approximately $O(n)$ pixels, if every boundary was traversed

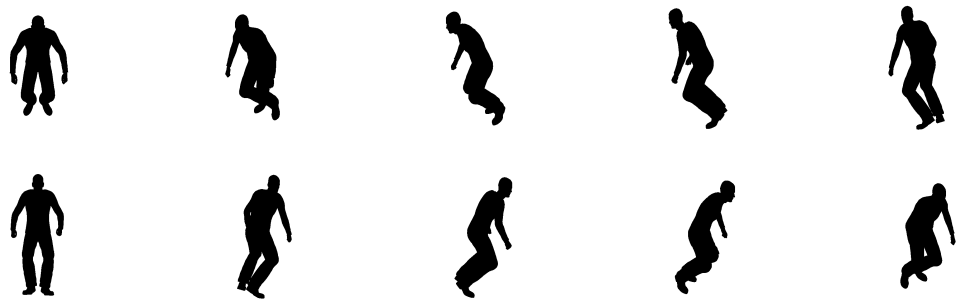


Figure 3.18: The synthetic silhouettes for a 3D model of a human, taken from 10 virtual cameras.

once to find all intersections with an epipolar line the running time would be $O(n^3)$, equivalent to a brute-force volumetric approach. Efficiency is improved by employing the bin representation of the silhouette boundary. This look-up table reduces the running time to $O(n^2 i)$ for a single camera, and $O(sn^2 i)$ where s is the number of cameras and i is the average number of silhouette intersections per epipolar line. The exact method presented here has a cost equivalent to that of the approximate solution presented in image-based visual hulls[56], and is more efficient than a brute-force volumetric approach with $O(sn^3)$ or an octree-based volumetric approach with complexity $O(sn^2 \lg n)$[78].

3.12 Conclusion

A novel algorithm for efficient computation of the exact View-Dependent Visual Hull has been presented which produces a sampled representation of the true visual hull surface. The cross ratio is used to order silhouette intersections in 2D and reduce the number of calculations required. A *Visual Hull Visible Intersection Theorem* is introduced to efficiently select the intersection corresponding to the Visual Hull surface. Advantages of the VDVH algorithm over previous visual hull methods are: (1) exact computation of intersection points on the visual

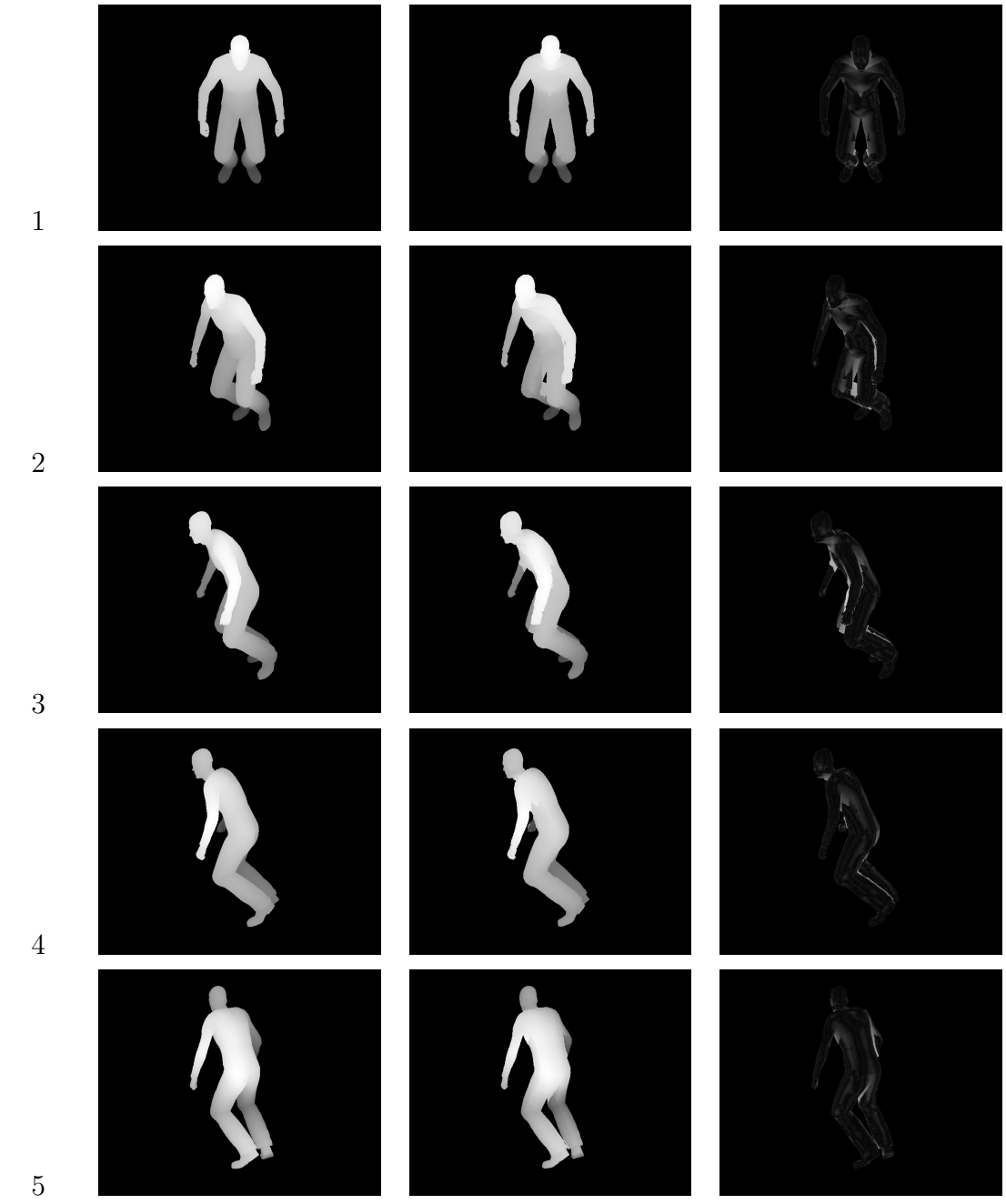


Figure 3.19: The depth image of the synthetic data, the depth image of the VDVH reconstruction, and the error intensity image of the two compared.

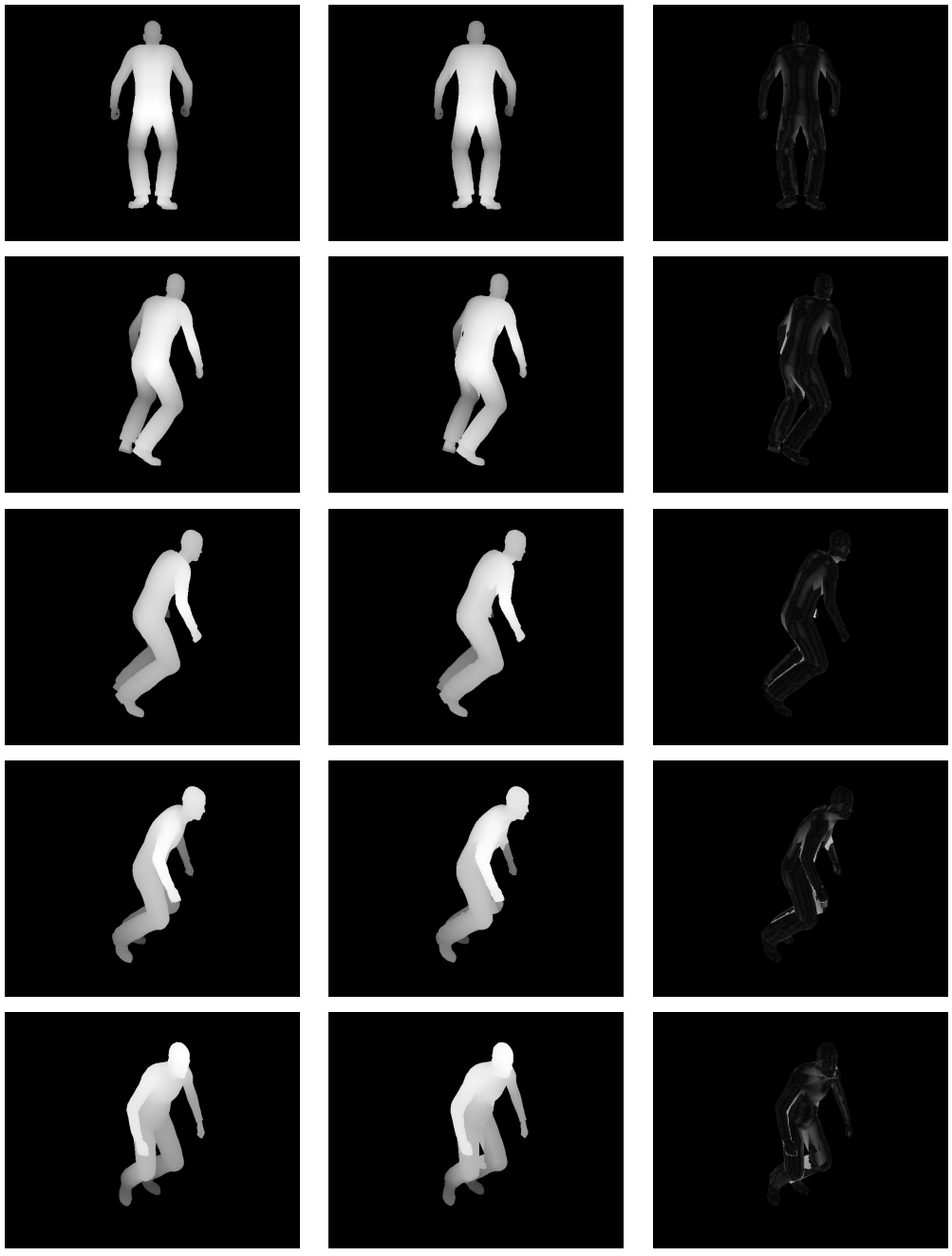


Figure 3.20: (continued from above) The depth image of the synthetic data, the depth image of the VDVH reconstruction, and the error intensity image of the two compared.

View	Volumetric	VDVH
1	9.19931	4.9909
2	10.9722	4.56936
3	10.0613	4.03987
4	7.01547	3.61171
5	7.25852	3.99748
6	9.28409	4.10974
7	10.7113	3.85874
8	8.80659	3.49118
9	9.24534	3.87662
10	9.38574	5.13064

Figure 3.21: Comparison of median errors per view between a volumetric visual hull reconstruction and VDVH. The table clearly demonstrates the improvement achieved using an exact sampling of the visual hull surface via VDVH. The figures displayed are approximately millimetres (converted from the units of the synthetic test).

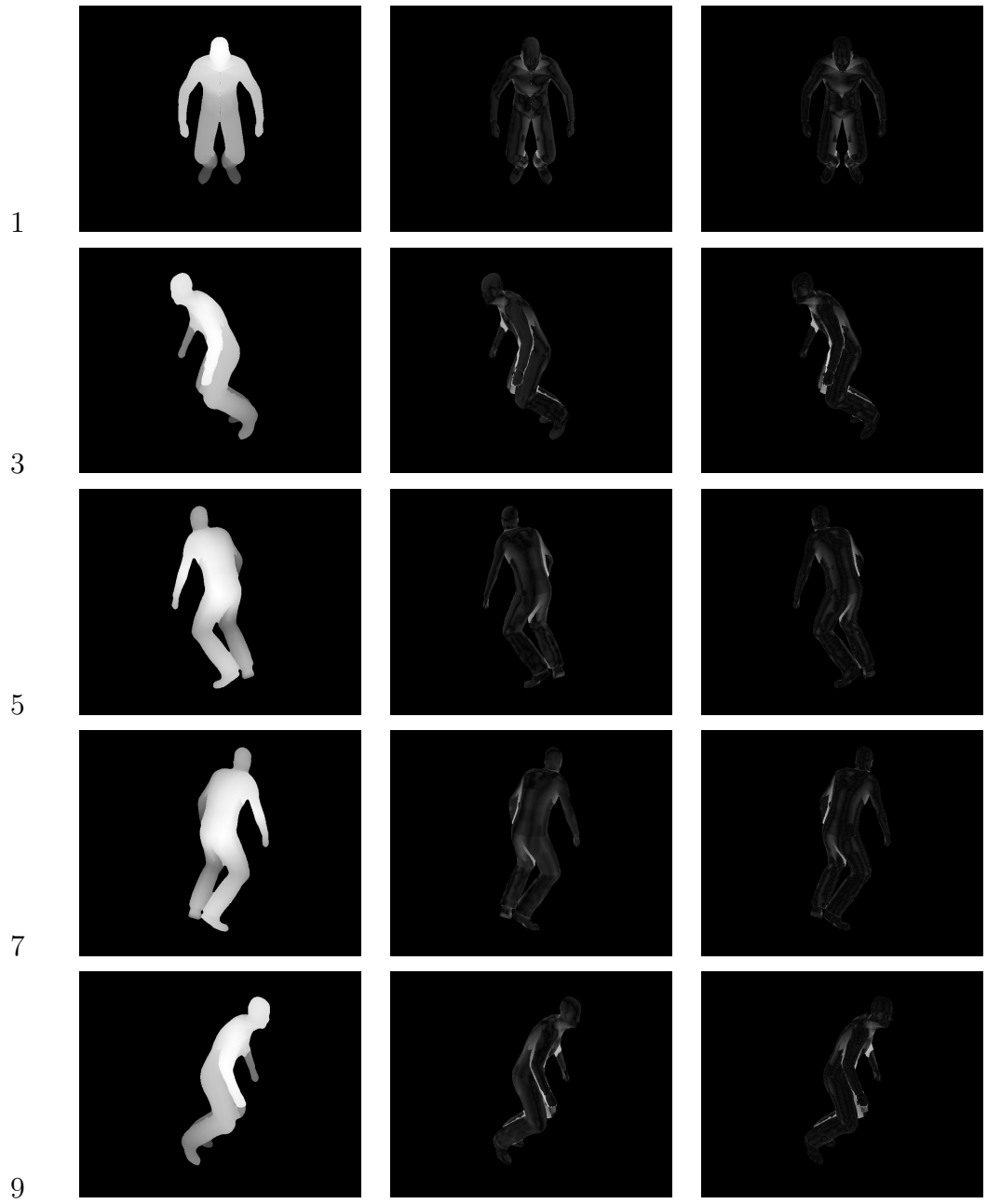


Figure 3.22: The depth image of the volumetric reconstruction, with the error of the volumetric reconstruction compared to the ground truth and the VDVH reconstruction compared to the ground truth (comparisons represented as error intensity images).

hull surface without requiring an intermediate approximation or quantisation step; and (2) efficient computation of intersections in 2D using the cross ratio. As with all visual hull methods, the algorithm is limited by the quality of the silhouettes, segmentation and calibration. Further work is required to optimise the segmentation of the input images.

Chapter 4

Efficient Local Refinement and Representation

The previous chapter described how to construct the visual hull from multiple views for use as a proxy surface in view synthesis. Using only the visual hull to render novel views leads to artefacts, for instance it is not capable of representing concavities in objects. This chapter presents an efficient technique for refinement of the visual hull to improve regions with concavities, and other parts where the surface does not lie close to the real object. Colour and intensity information are used from the original views to compute a refined proxy surface which allows us to blend from one view into another, between wide-baseline views.

A representation for multiple view video is also presented to allow high quality free-viewpoint rendering of video sequences with interactive control of the viewpoint in real-time. The refinement technique computes a surface using VDVH as an initialisation so that the view-dependently rendered surface is colour consistent between views. The refinement is efficiently processed using an image-based approach to obtain correspondence between views, which reduces the visual artefacts associated with visual and photo hull. The refinement itself is carried out

using a stereo matching technique based on texture intensity; correlation is pre-computed for computational efficiency and represented in a form which can be used for rendering novel views at interactive rates.

A novel representation for interactive free-viewpoint rendering from wide-baseline multiple view video capture is introduced in this section. The initial process is an offline construction and refinement of view-dependent visual hull (VDVH) surfaces. A multiple view video representation for online interactive rendering based on the refined surfaces is then presented.

The novel contributions of this chapter were published in *Interactive Free-Viewpoint Video*, Conference on Visual Media Production, 2005 [60].

4.1 Surface Estimation

Previous work has seen the visual hull surface widely used for rendering novel viewpoints from multiple view video capture. The visual hull is constructed from the set of captured images $\mathcal{I} = \{\mathcal{I}_n : n = 1, \ldots, N\}$ using the corresponding set of silhouettes $\mathcal{S} = \{\mathcal{S}_n : n = 1, \ldots, N\}$ produced via foreground extraction, where N is the number of calibrated views. The process and notation are both described in detail in Chapter 3.

The exact view-dependent visual hull (VDVH), as introduced previously, is extended to produce surfaces which are consistent between views. The inaccuracies in the visual hull produce erroneous alignment between views, therefore rendering novel views based on its geometry will result in visual artefacts (ghosting and blur). This limits the quality of virtual views and prohibits their use in broadcast production which requires a visual quality comparable to captured video.

(a) View-dependent visual hull (b) Overlapping area inside red line (c) Refined mesh representation

Figure 4.1: Stages in the refinement process at the mid-point between two cameras

4.2 VDVH Refinement

Refinement of the VDVH surface has been accomplished using two similar methods. The first computes a transition surface between every pair of adjacent views, and the second constructs a surface for every view, which allows a transition to every adjacent view.

4.2.1 Intermediate View Refinement

Given a novel viewpoint $\boldsymbol{c}_v$ between any two adjacent views $\boldsymbol{c}_j$ and $\boldsymbol{c}_k$, the depth map produced via VDVH with respect to $\boldsymbol{c}_v$ is an approximation of the scene which can be refined by applying a stereo matching algorithm. Direct computation of dense correspondence for wide-baseline views is an open problem in computer vision. Difficulties arise due to surface regions of uniform appearance, occlusion and camera calibration error. This work introduces an efficient image-based refinement of the VDVH using constrained stereo correspondence.

A proxy surface is constructed between the two adjacent views, coloured using each image and pixel-level refinement is applied where the colour is inconsistent. This process is demonstrated to achieve a surface approximation which allows novel viewpoint rendering with reduced visual artefacts from incorrect correspondence. An intermediate view is used to allow refinement between two wide-baseline views, and then to transition from one view to the other when rendering.

For every pair of physically adjacent cameras in the capture setup a visual hull surface is generated and refined. View-dependent refinement has been shown in previous work to improve rendering quality[75]. The reliability of correspondences is also improved in the presence of camera calibration error and changes in appearance with viewing direction.

The virtual viewpoint $\boldsymbol{c}_v$ is fixed at the midpoint between two adjacent cameras and a depth map is constructed for this view using VDVH. Coloured VDVHs are constructed for views j and k by constructing the VDVH and applying the colour at each pixel to its associated depth. These are projected onto $\boldsymbol{c}_v$'s image plane and the overlapping areas of the projections are compared. For every pixel in the overlapping region whose colour is inconsistent between views, the depth at that pixel is refined.

The system is initialised by constructing the VDVH for each real camera view $n \in [1, N]$ from the $(N-1)$ other views for all points inside the silhouette of the n^{th} view. A depth map M_n is produced via VDVH using the method defined in Section 3.10.

For each pair of adjacent cameras $\boldsymbol{c}_j$ and $\boldsymbol{c}_k$ with images $\mathcal{I}_j$ and $\mathcal{I}_k$, $j, k \in [1, N]$, the refined representation M_{jk} is obtained as follows:

> **Refinement:** Define the projection matrix P_{jk} of a virtual camera $\boldsymbol{c}_{jk}$ (positioned at the midpoint of the line connecting $\boldsymbol{c}_j$ and $\boldsymbol{c}_k$) by copying

the intrinsic parameters from a real camera and interpolating the extrinsic parameters of $\boldsymbol{c}_j$ and $\boldsymbol{c}_k$ (interpolation of rotation matrices is accomplished using quaternions). For this novel viewpoint:

(a) Evaluate the VDVH to produce a depth map, M_{jk}, for the virtual view $\boldsymbol{c}_{jk}$ from the N real camera views.

(b) Render the reconstructed surfaces M_j and M_k with colour onto $\boldsymbol{c}_{jk}$'s image plane to obtain images $\mathcal{I}^j_{jk}$ and $\mathcal{I}^k_{jk}$ containing only visible parts of the surface.

(c) For each pixel u in the reference image which has colour in both $\mathcal{I}^j_{jk}$ and $\mathcal{I}^k_{jk}$:

i. Test for colour consistency: $|\mathcal{I}^j_{jk}(u) - \mathcal{I}^k_{jk}(u)| < t_c$ where $I(u)$ is the RGB colour triplet for pixel u in image I and t_c is a threshold which determines how much refinement is required. The colour distance is defined as the difference between the two normalised RGB vectors (less variable to intensity variation).

ii. If pixel u is not colour consistent between images the depth map at u is refined using stereo matching. $M_{jk}(u)$ represents the distance $D_{jk}(u)$ along the virtual ray $\boldsymbol{r}$ from the camera centre $\boldsymbol{c}_{jk}$ to the visual hull intersection. Refinement starts at this depth and is constrained to lie inside the visual hull. An $m \times m$ window is used to evaluate the normalised cross-correlation between camera images $\mathcal{I}_j$ and $\mathcal{I}_k$ along the epipolar line for each view. The depth $d(u) = D_{jk}(u) + d'$ which gives the maximum correlation between views is taken as the refined depth estimate, or the original point is retained if no better match was found.

iii. The corresponding pixel in the depth map $D_{jk}(u)$ is updated with the refined depth estimate $d(u)$. The three-dimensional point at

this depth is computed and projected into $\mathcal{I}_j$ and $\mathcal{I}_k$ to retrieve the RGB values.

Output: M_{jk} contains depths from non-overlapping, overlapping and refined regions. The refined surface is textured by sampling from both images $\mathcal{I}_j$ and $\mathcal{I}_k$ which are blended based on visibility and the position of the required rendered view.

Stages of the refinement process are presented in Figure 4.1.

The algorithm constrains the refined surface for a camera pair to lie inside the visual hull. The refined mesh is evaluated offline for each pair of adjacent cameras. This provides the basis for online rendering of novel views with a higher visual quality than that obtained with the visual hull for wide-baseline views.

The border of the overlapping region is not refined since one of the cameras will have an unreliable view of the surface at these points. The colour threshold t_c is set to 0.05 for extensive surface refinement and 0.1 for conservative refinement. Throughout this work a 13×13 window is used in the stereo matching algorithm.

Occlusion in the target virtual view is not currently taken into account. For complex scenes there may be regions of the surface visible from the virtual view which are not visible in the adjacent reference views. The rendering step uses information from multiple views which may supply the missing information (at the cost of lower quality, since these regions may not be refined). In the results presented for free-viewpoint rendering of individual people this has not been found to produce visible artefacts. However, in more complex scenes with multiple people a reference representation with multiple depths per pixel may be required.

4.2.2 Reference View Refinement

The previous section presented a method for producing transitions between views. This section presents a more general solution to the problem by constructing a surface for each reference camera of the scene. Constructing a surface for every pair of views may require more surfaces than there are cameras, so constructing a surface for every camera is more efficient. The surface for a single camera then becomes the proxy for transitioning between multiple adjacent viewpoints. An additional benefit of this method is the use of multiple cameras to refine the surface.

The reference view refinement operation is very similar to the previous approach. Each real view has a depth map constructed with respect to itself using VDVH, producing a representation of approximate depth and known colour for each pixel. The colour consistency of each pixel at its approximate depth is tested against the adjacent cameras, and if inconsistent the depth is refined using stereo correspondence. Stereo refinement can be applied to a surface point using all cameras to which that point is visible, and the average of all views' best correspondences taken as the new depth.

Intermediate view refinement supplies a proxy surface for high quality transitions between views, but may require more surfaces than there are views to represent a general scene. The advantage of the reference view method is it requires the minimum number of surfaces to represent a scene. However, since the geometry is refined with respect to more than one camera it may not produce as high quality a transition to other views as the intermediate approach.

4.3 Representation for Interactive Free-Viewpoint Rendering

For free-viewpoint rendering the scene is represented by the R refined surfaces and view-dependent texture maps for all adjacent pairs of camera views. Rendering of novel views at interactive rates is achieved by rendering the set of R meshes in back-to-front order with back face culling enabled. The mesh generated from the camera furthest from the current viewpoint is rendered first, followed by the next furthest, and so on. The ordering is established using Euclidean distance between camera centres which is useful for setups where the cameras are all roughly the same distance from the subject. For a completely general scene using the angle between viewing direction would be more suitable.

The ordered rendering of the refined meshes guarantees that each pixel u of the final novel view image I_v is rendered from the closest refined view containing a colour for u. All refined meshes are rendered to ensure that any missing surfaces which may occur due to occlusion are included in the final rendering.

View-dependent rendering of each refined mesh is performed by blending the texture from the captured images I_j and I_k according to the angle between the camera and rendered view point. As in previous view-dependent rendering[75] this ensures a smooth transition between views using the estimated correspondence. At the location of the camera viewpoints the rendered image is almost identical to the captured image (the image is not absolutely identical due to resampling of the original images during rendering).

4.3.1 Computation and Representation Cost

Representation of the scene requires R meshes and associated textures to be stored for each frame of the multiple view video sequence. The rendering cost is

the total cost of rendering each of the individual meshes. If the camera image size is $P \times Q$ then each mesh has $O(PQ)$ vertices and the total cost of rendering is $O(RPQ)$. In the standard definition video used in this work $R = 8-10$, $P = 720$ and $Q = 576$ giving worst case representation and rendering cost of $6M$ textured triangles. In practice both the representation and rendering cost are an order of magnitude smaller as the foreground object only occupies a fraction (typically 25%) of the viewing area in any scene and approximately 50% of the triangles are back-facing for any given novel view. This gives representation cost at each frame of $1M$ triangles. This could be further reduced by pre-computing the overlap regions between view's meshes and rendering these once. Rendering can be achieved at interactive rates (greater than 25 frames per second) on consumer graphics hardware.

4.4 Results

This section presents results and comparative evaluation for interactive free-viewpoint rendering of people. Two different acquisition systems were used for testing:

Setup 1 Ten equally spaced cameras in an approximate circle of radius $4m$, baseline 36°, each capturing at 25Hz SD resolution (720×576) progressive scan. The original images from a capture from this setup can be seen in Figure 4.2.

Setup 2 Eight cameras in total, seven in an arc of 120° pointed towards the subject approximately $4m$ away. The eighth camera supplies a view from above. This setup uses the same cameras as Setup 1. The original images from a capture from this setup can be seen in Figure 3.12.

Tests were performed on an Intel Pentium IV 3.2GHz with 1GB RAM and results rendered using OpenGL on an nVidia 6600GT graphics card. This implementa-

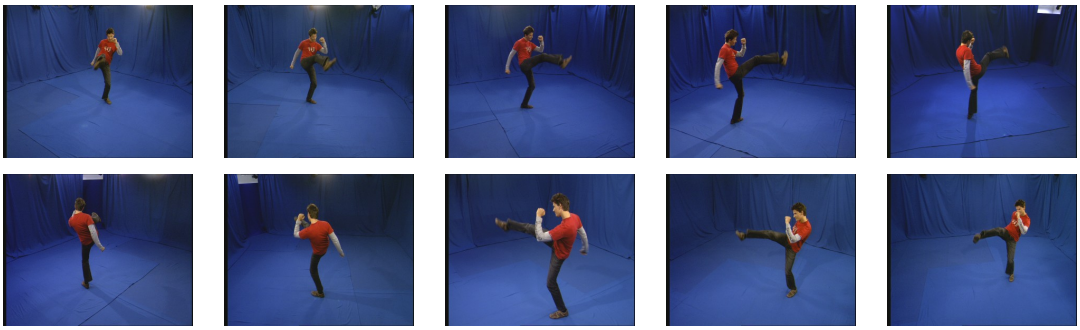


Figure 4.2: The original images from a single frame of a studio capture against a blue screen using Setup 1.

tion gives interactive rendering at 34 frames per second for novel viewpoints with the setup 1 and 43 frames per second for setup 2. Pre-computation for setup 2 takes approximately 3 minutes per frame.

4.4.1 Interactive Free-Viewpoint Video

Figure 4.3 shows a sequence of novel rendered views of a person at the midpoint between two real views using intermediate view refinement. The images in Figure 4.4 show novel rendered views of a person at the midpoint between each camera, using reference view refinement (the original views for this frame are shown in Figure 4.2). Results demonstrate the quality of rendered views which correctly reproduce detailed scene dynamics such as wrinkles in the clothing.

Figures 4.6 and 4.5 show interactive free-viewpoint video rendering of novel views for setup 1. The viewpoint is constrained to lie close to the original views (not exactly on the inter-camera paths) to ensure a smooth output. These images demonstrates that even through using a limited number of cameras high-quality novel view synthesis can be achieved for a complete circle surrounding the subject.

Figure 4.3: Video sequence from a virtual view at the midpoint of the line connecting two real views with a 36° baseline, generated using intermediate view refinement.

Figure 4.4: Reconstruction for a single frame shown from virtual views between every pair of views (with a 36° baseline), generated using reference view refinement.

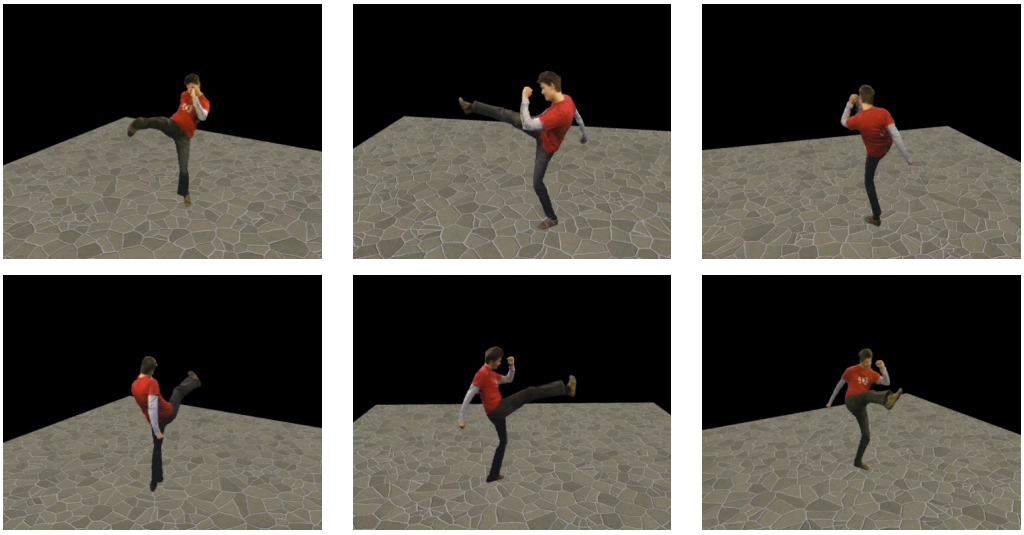


Figure 4.5: Screenshots from an interactive free-viewpoint video application: the images show the system running a bullet-time effect on a sequence captured using Setup 1.

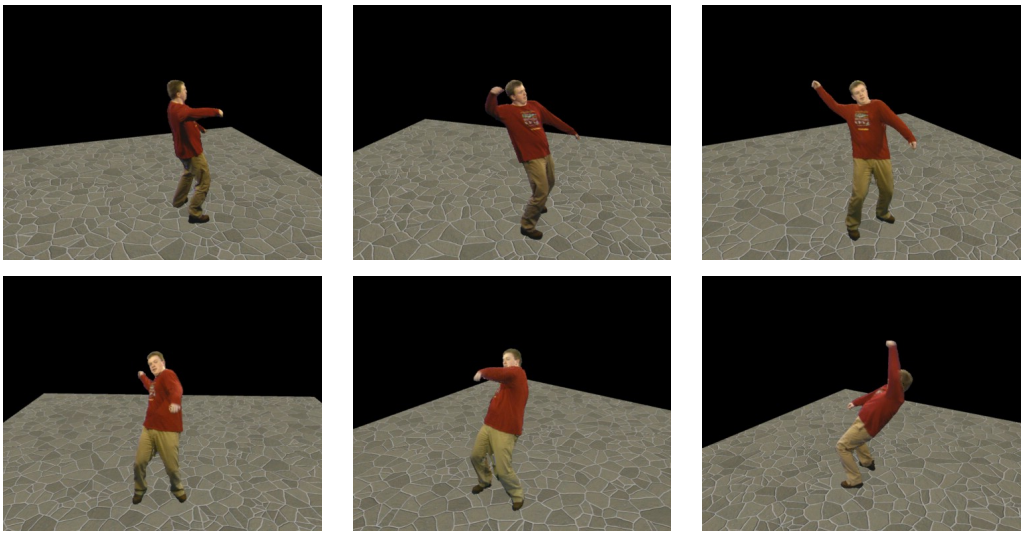


Figure 4.6: Screenshots from an interactive free-viewpoint video application: the images show 3D video on a sequence captured using Setup 1.

4.4.2 Comparative Evaluation

A comparative evaluation of free-viewpoint rendering quality from wide-baseline views has been performed comparing visual hull and photo hull with the representation based on stereo refinement introduced in this work. Figures 4.7 and 4.8 present comparative results for rendering of multiple video frames from a novel viewpoint for a sequence captured with setup 2. This comparison, and that of the close-up shown in Figure 4.9, demonstrates that visual artefacts present in the visual hull and photo hull rendering due to incorrect correspondence between views are not visible in the refined stereo surface. The rendering based on the refined representation reproduces hair and clothing movement. This representation eliminates visual artefacts such as ghosting due to incorrect correspondence which occur with previous visual and photo hull based free-viewpoint video techniques. The detailed pattern on the girl's dress is correctly reproduced demonstrating high quality rendering with interactive viewpoint control. Furthermore as the proposed representation and refinement is pre-computed rendering is performed at above video-rate on consumer graphics hardware.

4.4.3 Ground Truth Comparison

The missing view test setups, described in Section 3.11.3, are used to evaluate the quality of the novel rendered views using intermediate and reference view refinement.

The results of intermediate view refinement on the test set from setup 1 are shown in Figure 4.10. The quality of the synthesised novel view is not visually comparable to captured video; however, the baseline of 72° between these views provides a challenging task. The reference view refinement results are shown in Figure 4.11, and the results are similar to the intermediate view images. The reference view representation does not cover the complete surface, due to surface

(a) Visual hull (b) Photo hull (c) Stereo

Figure 4.7: Comparison of rendering using the visual hull, photo hull and the presented technique for intermediate view refinement, clearly showing the improvement of novel views, especially the sharpness in the torso regions.

(a) Visual hull (b) Photo hull (c) Stereo

Figure 4.8: Comparison of rendering using the visual hull, photo hull and the presented technique for intermediate view refinement, clearly showing the improvement in the novel views, especially the sharpness in the torso regions.

(a) Artefacts in visual hull

(b) Refinement via photo hull

(c) Proposed method

Figure 4.9: Close-ups of the stages of refinement showing reduction in artefacts using stereo refinement.

regions not visible to the cameras. The intermediate view representation fills the colour in with blending between the two cameras. In this case this has not led to artefacts, but may blend incorrect colours in the same way visual hull rendering does. A comparison of VDVH, intermediate view refinement and reference view refinement errors in the synthesised view is shown in Figure 4.12. The virtual view refinement produces the best quality image, but only by a few small details (such as around the collar). Due to the wide baseline used, the method was unable to accurately produce colour for the synthesised view.

For the first frame, the rms error of the VDVH is 0.098; the error for the intermediate view refinement is 0.097; the error for reference view refinement is 0.097. Quantitatively there is no improvement in image quality using the refinement, and so visual improvements are being offset by greater errors elsewhere.

The quality of the synthesised view for the test set from setup 2 is much higher than for the test from setup 1, and visibly improved from the quality produced via VDVH reconstruction. On both the intermediate view results (Figure 4.13) and the reference view results (Figure 4.14) the difference in the synthesised view and the ground truth image is reduced. The intermediate refinement performs better in this case as the error intensity images show (compared in Figure 4.15). The results indicate the error lies in the level of sharpness surrounding features in the scene, such as the flowers on the dress. These are slightly blurred in the reference view refinement, possibly due to the different orientation of the triangles in rendering (the sampling of the surface with respect to the reference views will produce longer triangles when rendering to a view directly between two real views). This effect also occurs for intermediate view refinement when rendering to virtual cameras positioned at the original viewpoints.

For the first frame, the rms error of the VDVH is 0.098; the error for the intermediate view refinement is 0.078; the error for reference view refinement is 0.085. Quantitatively the intermediate view refinement produces the best result, and

(a) Original (b) Refinement (c) Error

Figure 4.10: The images above are three frames of a sequence from a camera which was removed from processing to be used as ground truth. (a) shows the original images, the synthesised view via intermediate view refinement is shown in (b), and the error intensity image is shown in (c).

(a) Original (b) Refinement (c) Error

Figure 4.11: The images above are three frames of a sequence from a camera which was removed from processing to be used as ground truth. (a) shows the original images, the synthesised view via reference view refinement is shown in (b), and the error intensity image is shown in (c).

(a) VDVH Error (b) Intermediate Error (c) Reference Error

Figure 4.12: The images above are the error intensity images from Figures 3.16, 4.10 and 4.11. (a) shows the error with the ground truth of the synthesised view via VDVH, (b) shows the error with intermediate view refinement, and (c) shows the error with reference view refinement.

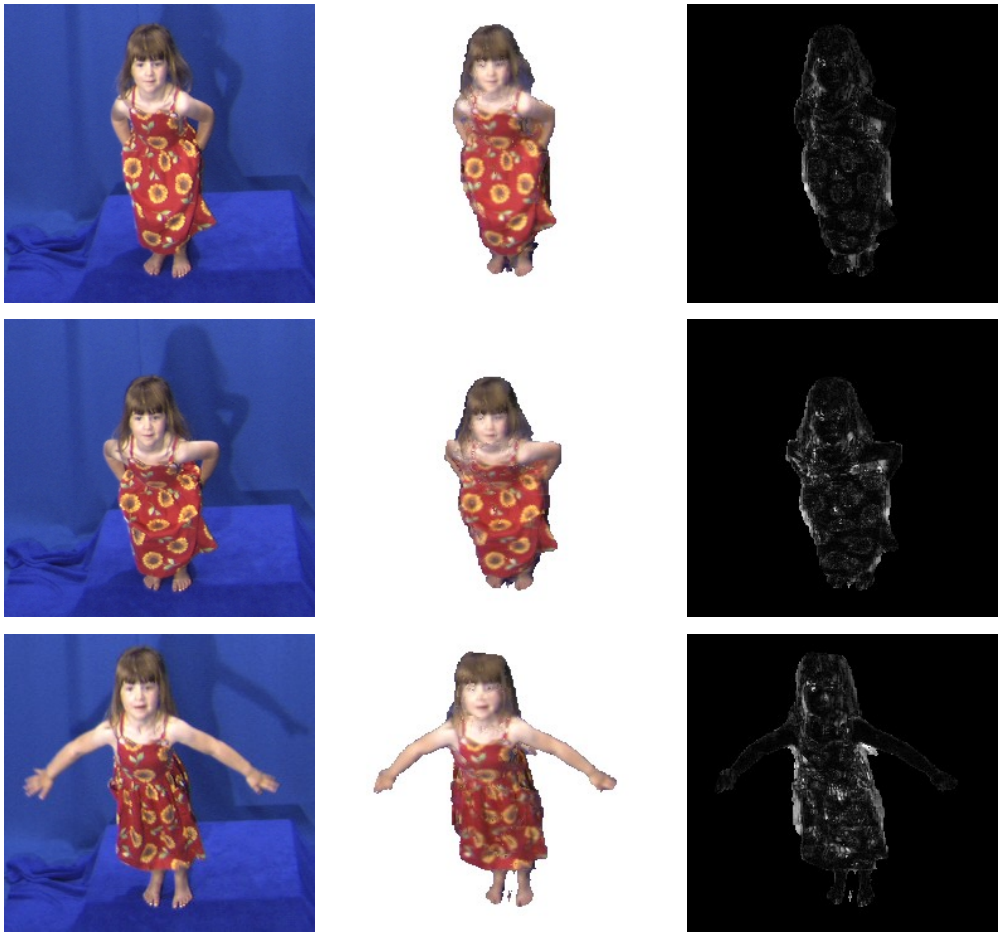
the reference view refinement provides an improvement over the visual hull.

4.5 Conclusions and Discussion

A representation for high-quality free-viewpoint rendering with interactive viewpoint control from multiple view wide-baseline video capture has been introduced. The representation is based on the pre-computation of stereo correspondence between adjacent wide-baseline views. Wide-baseline stereo correspondence is achieved by refinement of an initial scene approximation based on the view-dependent visual hull (VDVH). A novel algorithm for efficient VDVH computation has been presented which evaluates an exact sampling of the visual-hull surface for a given viewpoint. To estimate wide-baseline correspondence the VDVH for the mid-point between adjacent views is refined based on photo-consistency and stereo correlation. This produces a refined representation of the visible surface geometry and appearance with improved correspondence between views.

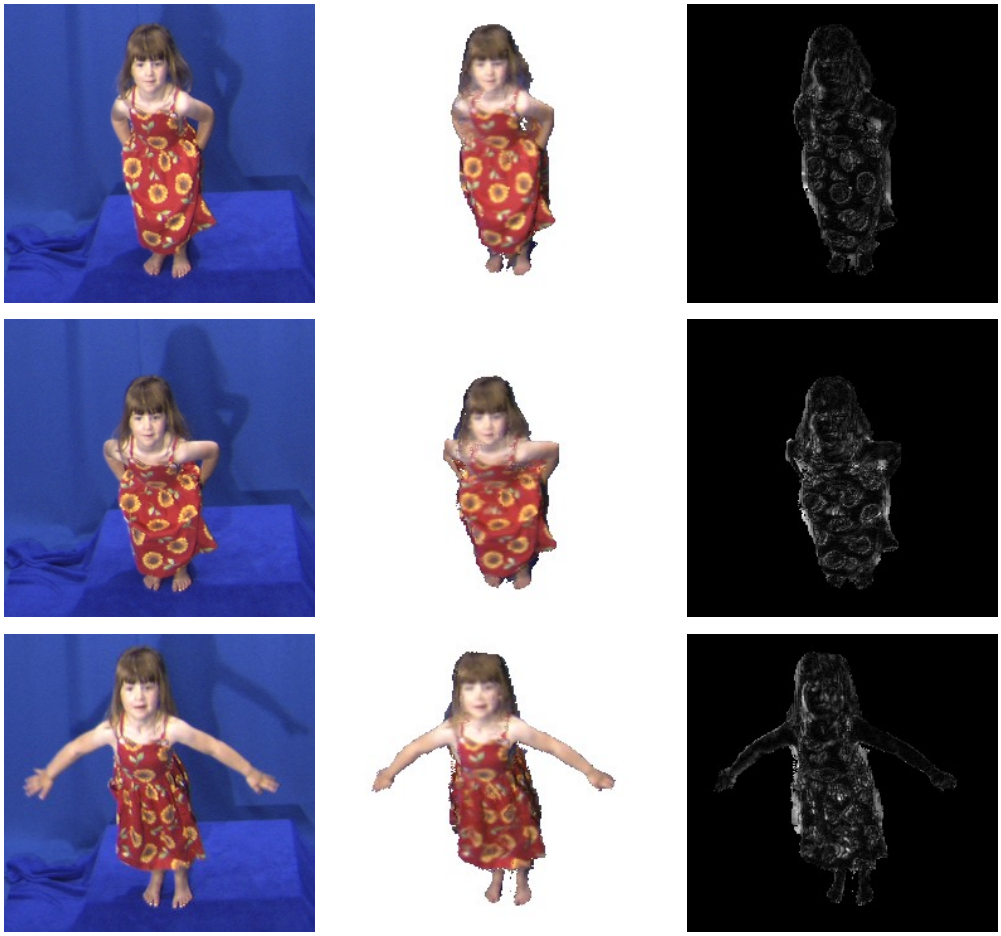
Interactive rendering of novel viewpoints is performed by back-to-front rendering or the refined representation starting from viewpoints furthest from the desired views and finishing with the closest viewpoint. Rendering is performed at video-rate (25Hz) on consumer graphics hardware allowing interactive viewpoint control. Results from 8 and 10 camera multi-view wide-baseline studio capture demonstrate high-quality rendering of people with reduced visual artefacts. Comparative evaluation with previous visual and photo hull approaches demonstrates that visual artefacts such as blur and ghosting are removed. The representation achieves high quality rendering with accurate reproduction of the detailed dynamics of hair and clothing.

The approach suffers from artefacts at the boundary of reconstructed surfaces due to errors in the silhouette segmentation. Further work is required to optimise the boundary segmentation together with the surface refinement.



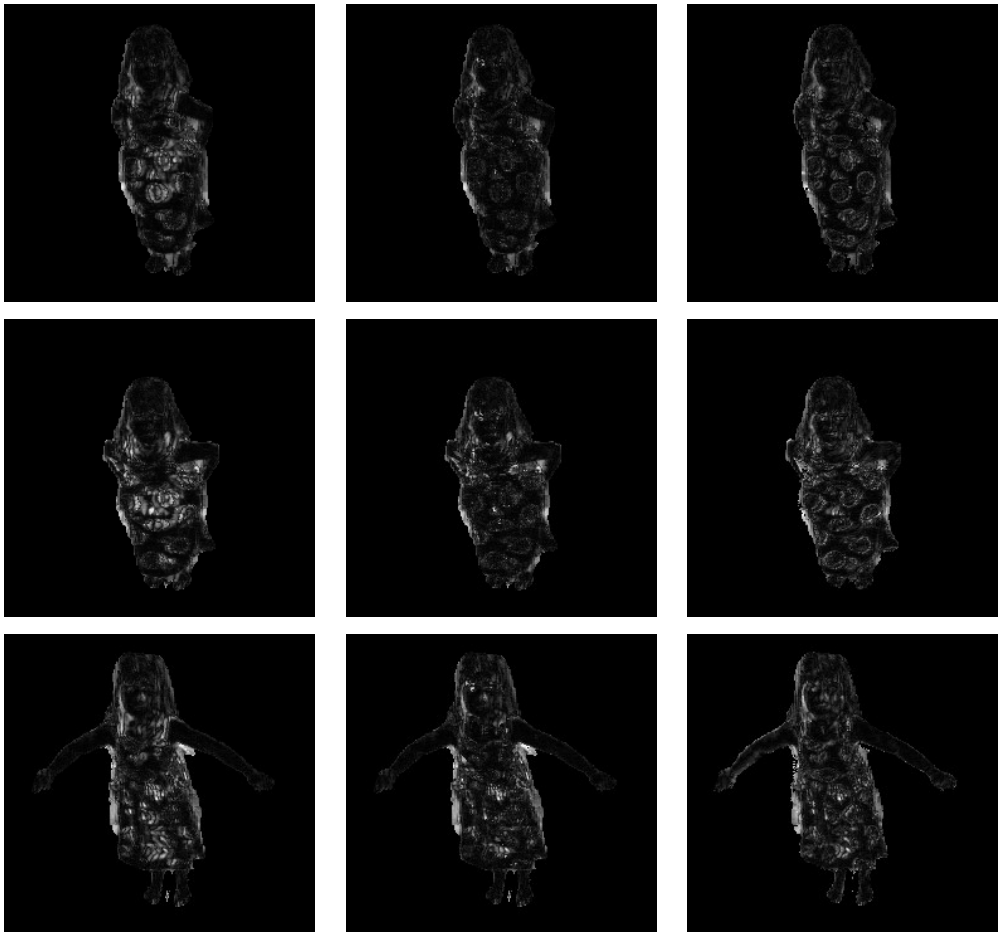
(a) Original (b) Refinement (c) Error

Figure 4.13: The images above are three frames of a sequence from a camera which was removed from processing to be used as ground truth. (a) shows the original images, the synthesised view via intermediate view refinement is shown in (b), and the error intensity image is shown in (c).



(a) Original (b) Refinement (c) Error

Figure 4.14: The images above are three frames of a sequence from a camera which was removed from processing to be used as ground truth. (a) shows the original images, the synthesised view via reference view refinement is shown in (b), and the error intensity image is shown in (c).



(a) VDVH Error (b) Intermediate Error (c) Reference Error

Figure 4.15: The images above are the error intensity images from Figures 3.17, 4.13 and 4.14. (a) shows the error with the ground truth of the synthesised view via VDVH, (b) shows the error with intermediate view refinement, and (c) shows the error with reference view refinement.

The intermediate view representation is limited because it assumes that the refined surface at the midpoint between views includes all overlapping visible surface regions for the adjacent views. This assumption is not guaranteed due to occlusion. This is an advantage of the reference view approach since it represents all surface visible to the original views. A method of incorporating both the intermediate and reference view refinement techniques into a single representation to combine the strengths of both will be investigated.

Chapter 5

Constrained Global Surface Optimisation

This chapter presents a novel method of surface refinement for free-viewpoint video. The previous chapter presented a local refinement method to preserve colours when transitioning between views. The approach presented in this chapter performs a global surface refinement for each view's visible surface and uses the previously described representation for rendering.

The global refinement uses both visual hull and silhouette contours to preserve information from the original images for refinement of view-dependent surfaces. Silhouette contours are represented in 3D as *rims*, and a novel technique is presented for extracting rims from the view-dependent visual hull (VDVH). Given the VDVH as an approximation, a new method for improving correspondence is presented where refinement is posed as a global surface optimisation problem in projective ray space. Rims provide local information which constrain the refined surface to lie on known strips of the true surface, and the global optimisation reduces artefacts such as depth discontinuities that can occur with local approaches. Real time rendering of novel views in a free-viewpoint video system is achieved

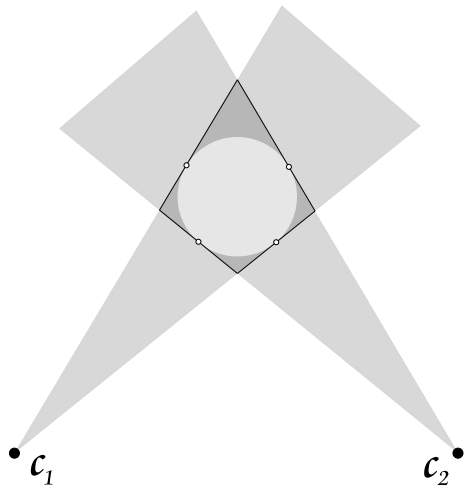


Figure 5.1: The circle represent the scene. The silhouette cones (light shade) are projected out from the cameras and form the visual hull where they intersect (darker region). The highlighted lines are boundary edges, and the points on them represent the rim points for those edges.

using the image+depth representation introduced in the previous chapter.

The novel contributions of this chapter were published in *Projective Surface Refinement for Free-Viewpoint Video*, Conference on Visual Media Production, 2006 [61]. Contributions from this chapter were also included in parallel research published in *Volumetric Stereo with Silhouette and Feature Constraints*, British Machine Vision Conference, 2006 [77].

5.1 Background Theory

This section briefly covers the background theory for visual hull rims and network flow; a review of relevant literature can be found in Chapter 2.

The visual hull is constructed from the set of captured images $\mathcal{I} = \{\mathcal{I}_n : n =$

$1, \ldots, N\}$ using the corresponding set of silhouettes $\mathcal{S} = \{\mathcal{S}_n : n = 1, \ldots, N\}$ produced via foreground extraction, where N is the number of calibrated views. The process and notation are both described in detail in Chapter 3.

Assuming perfect matting and calibration, the set of pixels $\mathcal{B}_n$ on the boundary of $\mathcal{S}_n$ have the unique property that the ray cast from $\boldsymbol{c}_n$ through $p \in \mathcal{B}_n$ touches the surface of the scene object. Given the visual hull constructed with respect to $\boldsymbol{c}_n$, the depthels for $\mathcal{B}_n$ are extracted to produce a set of intervals $\mathcal{D}_n$ (bounding edges) in projective ray space. The surface point touched by the ray can be evaluated using colour consistency of neighbouring cameras from which the ray is visible. The smooth curve through the points on $\mathcal{D}_n$ is called the *rim* of the visual hull. Various methods have been presented in the past to extract rims from a visual hull reconsruction [17, 70]. The research presented here advances this by providing an optimisation technique on rims that can be applied to arbitrary scene objects.

Section 5.2.2 describes how to retrieve the rims $\mathcal{R}_n$ for the n^{th} view using an optimisation on $\mathcal{D}_n$. The intervals in $\mathcal{D}_n$ are extracted from a multi-layer depth map M_n representing the exact visual hull (see Chapter 3), avoiding additional quantisation.

5.1.1 Network Flows and Graph Cuts

Graph cuts on flow networks have become a popular way to solve optimisation problems in computer vision because it finds a global optimimum solution. Recent evaluation of multiple view surface reconstruction[68] show techniques based on graph cuts produce the most accurate results. This chapter presents methods to recover the rims and refined surface of the object via graph cuts. The optimisation uses strong stereo correspondence to constrain the solution over regions of uniform appearance.

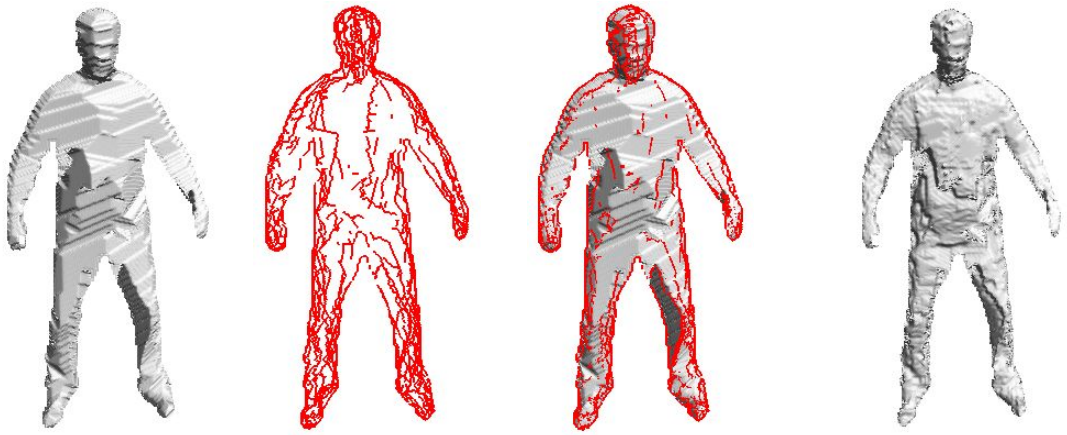
A *flow network* $G = (V, E)$ is a graph with vertices V and edges E, where each edge $(u, v) \in E$, $u, v \in V$ has a capacity $c(u, v)$[18]. G has a source $s \in V$ and a sink $t \in V$ defining the direction of flow. A *graph cut* (S, T) of G partitions V into S and $T = V - S$ such that $s \in S$ and $t \in T$. The capacity of a cut is $c(S, T) = \sum_{u \in S, v \in T} c(u, v)$. Finding a flow in G with the maximum value from s to t is known as the maximum flow problem, which, by the *max-flow min-cut theorem*, is equivalent to finding the minimum capacity cut of G.

Section 5.2.3 presents a novel method for the global optimisation of a depth map M_n for the n^{th} view using the set of rims $\mathcal{R} = \{\mathcal{R}_n : n = 1, \ldots, N\}$ to constrain the problem with local information.

5.2 Projective Surface Refinement

This section introduces a novel method for global refinement of the surface visible from a specific view by enforcing depth and silhouette contour constraints in projective ray space.

Global surface refinement techniques produce artefacts where no reliable information is present, for example in a surface region of uniform or regular appearance. This can lead to over- or under-refinement of the surface. Incorporating information from $\mathcal{S}$ (the silhouettes of the scene) additional constraints can be applied to the surface optimisation. The method presented here refines depth maps produced with respect to an existing viewpoint using view-dependent visual hull (VDVH). The rims are evaluated for each view's VDVH using a graph cut on $\mathcal{D}_n$. These are incorporated into a global optimisation of the visible surface formulated as a network flow problem. Vertices are positioned inside the visual hull in projective ray space, and given a score from stereo matching between adjacent views. The graph cut yields the refined surface which is converted into an image+depth representation for real-time rendering.



(a) VDVH (b) Rims (c) VDVH + Rims (d) Refined Depth Map

Figure 5.2: Stages of surface reconstruction for a specific viewpoint from the initial VDVH approximation to the globally optimised surface.

5.2.1 Initial Surface Approximation

The refinement technique relies upon an initial approximation to the surface for the following reasons: it directly supplies a narrow search space for refinement; a subset of the true surface can be recovered in the form of rims to constrain the optimisation; and it allows use of wide baseline cameras for stereo matching.

The initial surface approximation is the full visual hull generated by the VDVH algorithm. In the multi-layer depth map produced there are an even number of intersections for every depthel, the odd intersections are the ray entering the surface, and the even ones exiting it. The intersections are grouped into intervals representing the segments of the ray inside the visual hull surface. The first interval on each ray from the camera centre is the search space for refinement, since we're improving the visible surface only.

5.2.2 Rim Recovery

The set of rims $\mathcal{R}_n$ for the n^{th} view can be recovered by finding the points on the rays through pixels on the silhouette contour $\mathcal{B}_n$ which correspond to the true surface. On a depth map M_n produced using VDVH, the surface point lies on the interval corresponding to $M_n(u)$, $u \in \mathcal{B}_n$. For this work, only contour points with one interval in M_n are considered since those with multiple intervals may represent phantom volumes, an artefact of visual hull resulting from occlusion or multiple objects in a scene. (See the next chapter for an in-depth discussion of phantom volumes and a method for removing them.)

The rim for a single genus-zero object with no self-occlusion is a smooth continuous curve. This scene constraint has been invoked in other work[70], however the goal of this section is to find the rims on visual hulls representing people. The technique must therefore deal with non-genus-zero surfaces, and occlusion either from one object occluding itself or from the presence of multiple objects. Occlusions appear in the depth map as depth discontinuities.

As with any visual hull based technique, it is important to have good camera calibration and image matting. For a synthetic scene where calibration and matting are perfect the contour of the silhouette will directly correspond to the contour of the depth map silhouette (an image constructed from a depth map by setting pixels with depths as foreground and those without as background). In practice, calibration and matting both have some degree of error, so the silhouette used to construct the rims is taken from the depth map.

Before constructing the rims the contour of the silhouette must by analysed to detect occlusions. This process will produce a set of *pixel chains* $\mathcal{C} = \{\mathcal{C}_i : \mathcal{C}_i \subseteq \mathcal{B}_n, i = 1, \ldots, N_{\mathcal{C}}\}$ where $\mathcal{C}_i$ is an ordered set of pixels on $\mathcal{B}_n$ and $N_{\mathcal{C}}$ is the number of chains.

To produce the chains $\mathcal{B}_n$ is represented as an ordered set of pixels $\mathcal{O}_n$. $\mathcal{O}_n$ is

analysed to produce pixel chains: if the interval $M_n(p_t), p_t \in \mathcal{O}_n$ overlaps the interval $M_n(p_{t-1})$, p_t is added to the current pixel chain $\mathcal{C}_i$. Otherwise p_t marks a depth discontinuity (occlusion), so $\mathcal{C}_i$ is saved and $\mathcal{C}_{i+1}$ begins a new chain with p_t. For a scene with no occlusions, one pixel chain is produced.

One rim segment is produced for every chain $\mathcal{C}_i \in \mathcal{C}$. For every $p \in \mathcal{C}_i$, the interval $M_n(p)$ is sampled regularly and each sample is given a score based on a stereo comparison between two camera views with good visibility of the interval.

Previous methods found the point on the interval with the highest photo consistency score[17], but this approach leads to a discontinuous rim, because surfaces may have uniform appearance or repetitive patterns which give false positives. An optimisation problem is formulated for each pixel chain to obtain a smooth continuous curve for its rim segment. Each chain is set up as a flow network and the optimum path (the rim) through the intervals is found via a graph cut.

Each interval on the chain is sampled regularly, using the effective sampling resolution of the nearest camera at the current depth. The effective sampling resolution is half the distance between pixels for an image plane projected from the camera to the current depth at the original resolution. Every sample is given a score using normalised cross-correlation stereo matching between two adjacent cameras with the best visibility of the point. The score for each sample is mapped to the range $[0, 1]$. Visibility maps are constructed as described in Section 3.8. At a sample which is not visible to two adjacent views but is visible to at least two views, a photo consistency test is performed to attach a score to the sample. Regions of zero visibility (for example, under the arms) are given scores of 0.5 (the midpoint of the range of scores) so as not to bias the optimisation and allow interpolation over these regions.

Stereo windows in the original images are constructed using a base plane in 3D, set up tangentially to the surface to improve correlation scores. A square correlation window is used with dimensions set manually using the units of the calibration

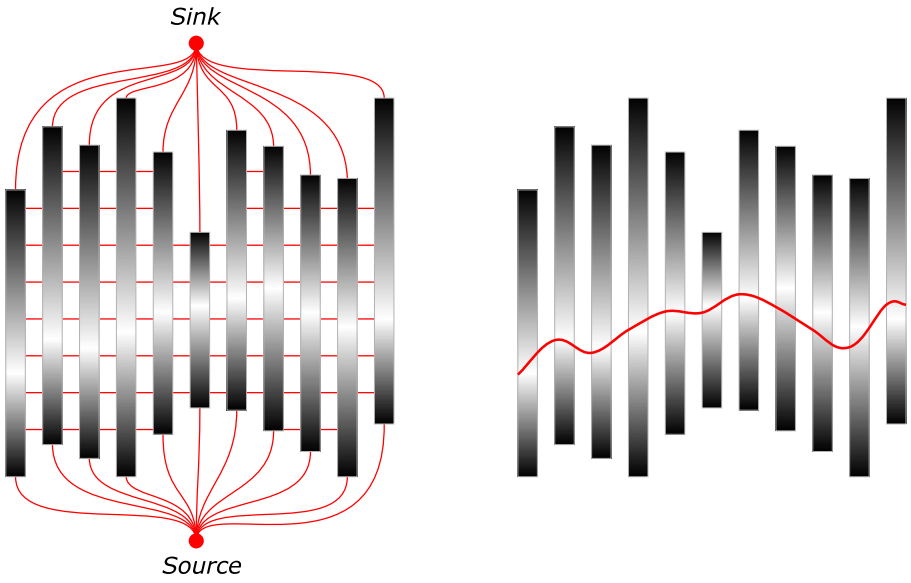


Figure 5.3: Diagram showing a graph cut on a chain: intervals are shown as columns in which depth increases vertically from the bottom. Good stereo scores are represented as white, and bad scores as black. The graph setup is shown on the left, with adjacent depths connected (and adjacent vertices on each interval are also connected, but not explicitly shown). The red line through the white region on the graph on the right is the cut, representing the rim.

(usually metres). The orientation of the window is established as follows: the derivative of the silhouette contour at the current pixel is found and rotated 90° to give a 2D perpendicular vector pointing out of the silhouette. The 3D normal is evaluated and used to construct the 3D window at the required point on the interval with the same normal as the surface point. The 3D window is projected onto each image to produce two images for comparison.

A flow network for each chain is constructed as a set of vertices $\mathcal{V}_{\mathcal{C}_i}$ based on the sample points, and a set of edges $\mathcal{E}_{\mathcal{C}_i}$ based on the scores. The first vertex of every interval is connected to the source $s \in \mathcal{V}_{\mathcal{C}_i}$ and the last to the sink $t \in \mathcal{V}_{\mathcal{C}_i}$, shown in Figure 5.3(left). A 4-connected neighbourhood is set up on the rest of the

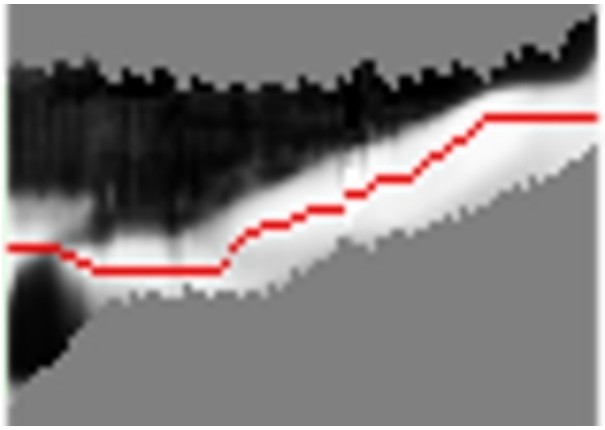


Figure 5.4: Diagram showing a graph cut on a chain: intervals are shown as columns in which depth increases vertically from the bottom. Good stereo scores are represented as white, and bad scores as black. The dark line through the white region is the graph cut, representing the rim.

graph. Adjacent vertices on an interval are connected by an edge, and vertices at equivalent depths between intervals are connected. The capacity of each edge is $c(u,v) = 1 - \frac{s(u)+s(v)}{2}$, $u, v \in \mathcal{V}_{\mathcal{C}_i}$, where $s(u)$ is the score at vertex u. Stereo scores are maximal, whereas for a flow network a good score should have a low capacity, so the average score is subtracted from 1.

The graph cut is applied to $\mathcal{C}_i$ to retrieve the rim segment's path through the interval, as shown in Figure 5.3(right). An example of a real graph cut on actual data is shown in Figure 5.4. This is mapped into 3D using the depths on the interval to recover the actual rim segment. This process is performed for every $\mathcal{C}_i \in \mathcal{C}$ to retrieve $\mathcal{R}_n$, the rims for view n. $\mathcal{R}$, the complete set of rims, is found by applying this process for every viewpoint, which is important for constraining the global optimisation (the rims in $\mathcal{R}_n$ do not constrain the interior surface of $\mathcal{M}_n$, whereas the rims from other views do).

5.2.3 Constrained Global Optimisation

The refined surface for rendering is produced by performing a global optimisation on the view-dependent surface (the depth map). Refining depth maps has been proposed before, but has either neglected silhouette constraints[9] or performed a local refinement which produces a discontinuous surface (such as the method described in the previous chapter). The novelty of this work is to first constrain the problem using VDVH to define the search range (allowing use of wide baseline views), and secondly to use rims to provide local information to achieve a higher quality surface reconstruction.

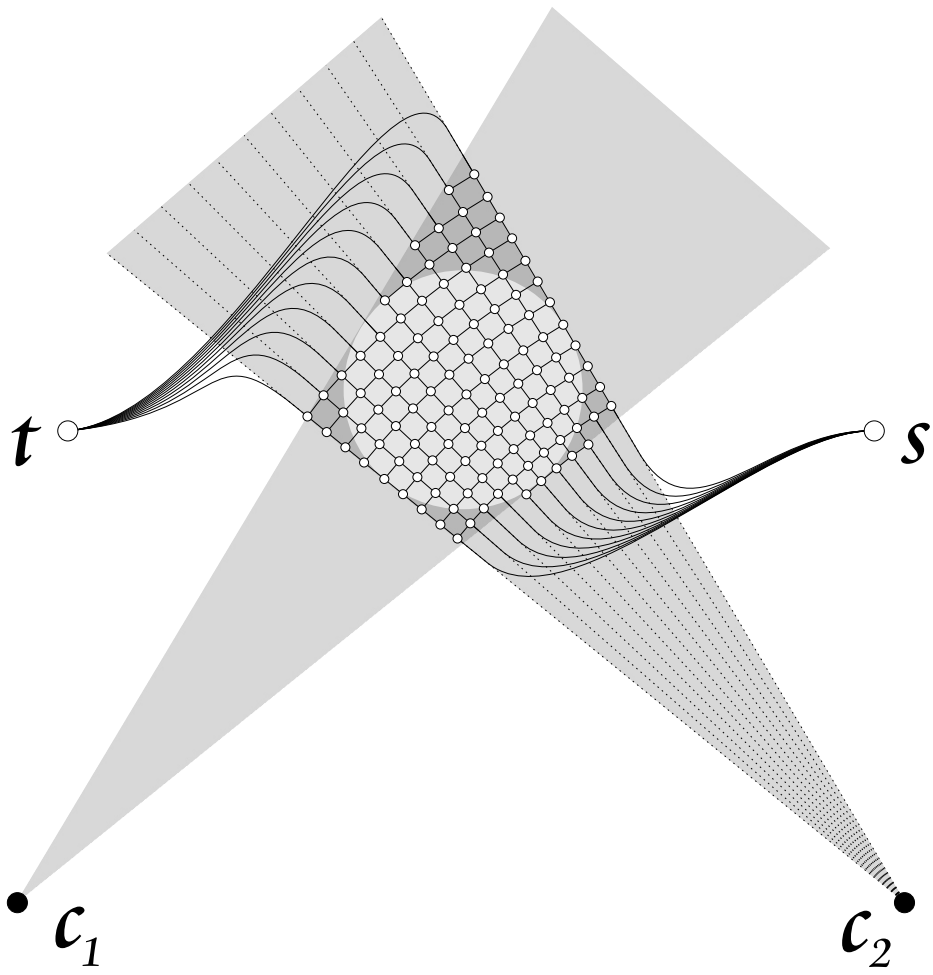
The technique for performing a global optimisation on a depth map produced using VDVH without enforcing contour constraints is defined first.

Global Optimisation of Depth Maps

Global optimisation is performed on the first layer of a multi-layer depth map, using the second layer (end of the first interval) to constrain the search space. More formally, let $\mathcal{P}_n = \{p \in M_n : p \textit{ is non-empty}\}$, then $\forall p \in \mathcal{P}_n$ the possible location of the surface is defined strictly by the first interval of $M_n(p)$. The set of intervals $\{M_n(p) : p \in \mathcal{P}_n\}$ exist in projective ray space: the intervals are defined on rays cast through $\mathcal{P}_n$ from the camera centre $\boldsymbol{c}_n$.

The intervals are sampled at regular depths to produce vertices on a 3D projective grid. Each vertex is given a score from the stereo comparison between view n and an adjacent viewpoint (chosen based on visibility). A normalised cross-correlation on a window around the pixel in $\mathcal{I}_n$ and the window around the projection of the vertex to the adjacent view is used to produce a correspondence score (mapped to the range $[0, 1]$).

The optimisation for the n^{th} view is formulated as a flow network $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$ with vertices $\mathcal{V}_n$ and edges $\mathcal{E}_n$, illustrated in Figure 5.5. The first vertex of every



Global optimisation

Figure 5.5: A cross-section example of a graph set up on the visual hull from Figure 5.1 in projective ray space with respect to $\boldsymbol{c}_2$. Vertices are marked as white circles, connected by edges marked in black. The first vertex of every interval is connected to the source s, and the last is connected to the sink t.

interval is connected to the source $s \in \mathcal{V}_n$ and the last to the sink $t \in \mathcal{V}_n$. A 6-connected neighbourhood of edges is set up for the internal vertices. Vertices at equal depth on horizontally and vertically adjacent intervals are connected by an edge, using the capacity function $c(u, v)$, $u, v \in \mathcal{V}_n$ from Section 5.2.2. Adjacent vertices on an interval are connected by an edge using $c(u, v)$ with a smoothing multiplier k. As the value of k increases, the resulting surface moves toward the best scores per interval with less influence from constraints. Correspondingly, as k decreases the surface is more constrained; at $k = 0$ the surface corresponds to the initial approximation.

The refined surface is produced by separating the graph into two regions using the max-flow min-cut algorithm. Only edges along the intervals are checked to see if they were part of the cut, and the vertices on the edges which were cut are extracted for the surface (the vertex further away from the camera is chosen).

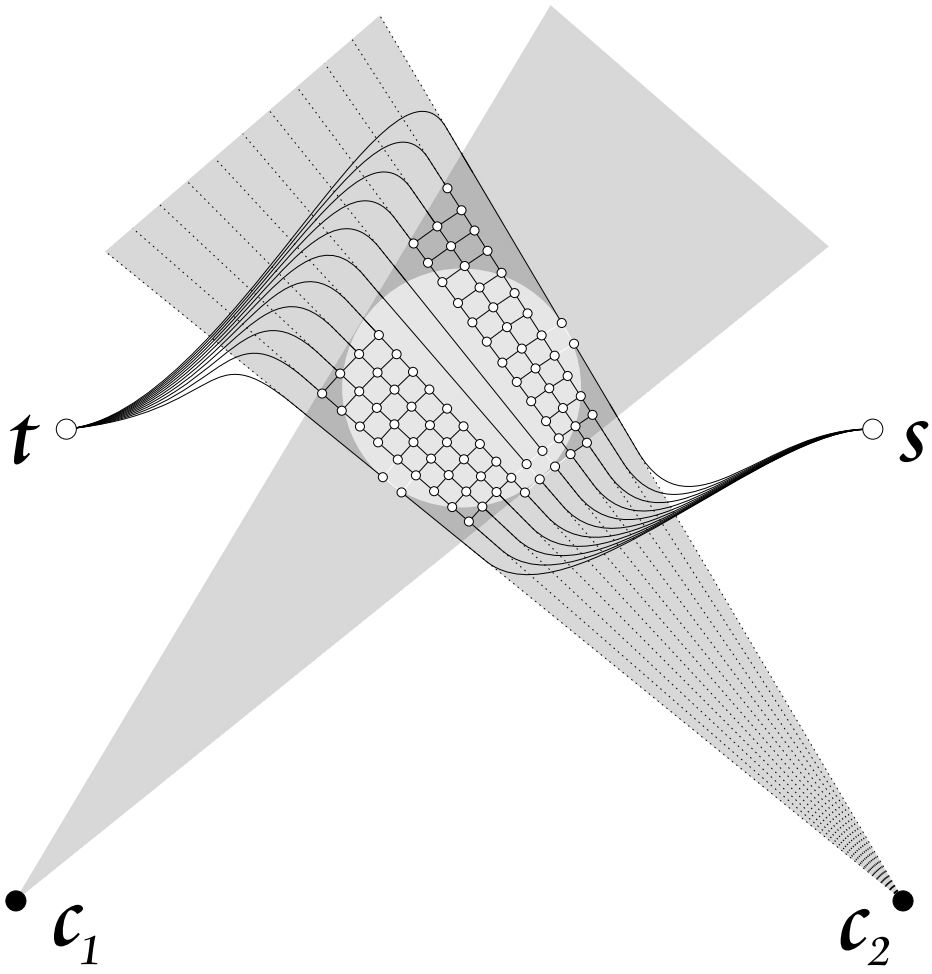
This method for global optimisation works very well in detailed regions of the surface, and performs a 'best guess' in regions of uniform appearance with similar scores. Unfortunately this can lead to incorrect surfaces due to the stereo scores over a volume having similar values (see Figure 5.7 in the results section).

Rim-Constrained Optimisation

The novel approach presented here incorporates the rims into the optimisation problem to provide local constraints, preserving the original information from the silhouette contours.

The rims are added to the flow network as it is set up, with one pre-computed step. A set of points $\mathcal{R}_n^v = \{\boldsymbol{p} \in R : \boldsymbol{p}$ *visible to view* $n, R \in \mathcal{R}_j, j = 1, \ldots, N\}$ is extracted from the set of rims if they are visible to the current view.

Every $\boldsymbol{p} \in \mathcal{R}_n^v$ is projected onto the image plane of the n^{th} view. Edges are not added to the graph between the four pixel centres surrounding it, or to the



Rim constrained optimisation

Figure 5.6: The same graph from Figure 5.5 with rim constraints included. Vertices are removed where the surface is known not to exist, and vertices where the surface is are connected by zero capacity edges (shown as white). The cut is expected to follow the shape of the underlying surface (the circle) more closely.

vertices on the intervals corresponding to the four pixels. Instead, for each of the four pixels an edge is added between depths at the depth of the rim with a capacity of zero; horizontal and vertical edges are added for the vertices at those depths to adjacent intervals and among the four, as shown in Figure 5.6. Allocating a capacity of zero to the edges corresponding to the rim's location guarantees that edge becomes part of the cut, and the rest of the cut is bound to this depth. The remaining graph structure spans regions of unknown surface, but will now be constrained to lie close to the rims and improve the reconstruction. The smoothing value k dictates how constrained the graph cut is by the rims: large values of k let the optimisation deviate from the rims if the scores allow.

The surface in Figure 5.7(d) is more accurately the shape of a shoulder due to adding rims to the global surface optimisation, compared to the surface without rims shown in Figure 5.7(c) which the global optimisation refined further than required due to lack of constraints.

5.3 Rendering

The refinement operation produces N image+depth sufaces per frame; identical topology is used to produce a mesh of each surface for free-viewpoint rendering. Novel views are synthesised in real-time by rendering the N meshes in back-to-front order.

View-dependent rendering of each mesh is performed by blending the texture from images $\mathcal{I}_m$ and $\mathcal{I}_n$ when transitioning between views m and n. The colour from each image is weighted according to the angle between the camera and the rendered viewpoint. This ensures a smooth transition between views using the estimated correspondence.

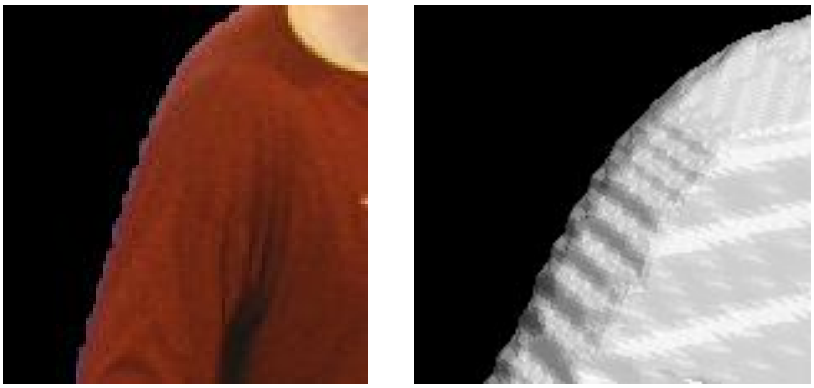
The use of multiple local representations over a single global representation gives

the best correspondence between adjacent views in the presence of camera calibration error and reconstruction ambiguity[75]. High quality rendering with accurate reproduction of surface detail is achieved using locally refined surfaces.

5.4 Results

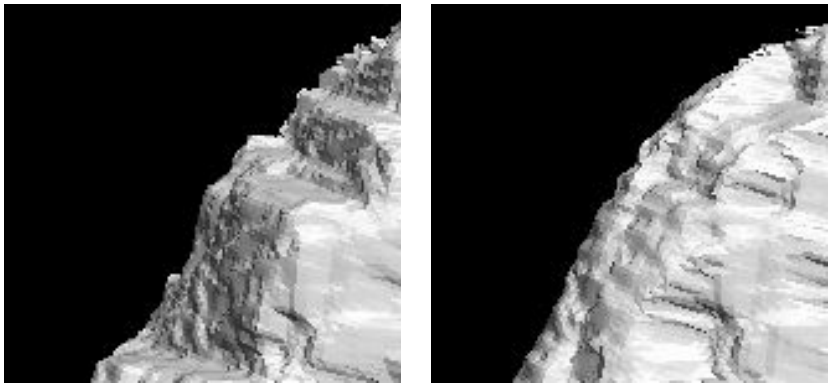
This section presents results and evaluation of projective surface refinement for free-viewpoint rendering. Multiple view video capture was performed in a studio with eight cameras equally spaced in a ring of radius $6m$ at a height of $2.5m$ looking towards the centre of the studio. Each camera pair had a baseline of $4.6m$ with a $45°$ angle between them, and the capture volume was approximately $8m^3$. A comparative evaluation of the proposed method was performed against results from previous work (Chapter 4). The studio setup for these results comprised eight cameras, seven in an arc spanning $110°$ of radius $4m$ with a baseline of $1.2m/18°$ and approximate capture volume of $2.5m^3$ (the eighth camera gave a view from above). Synchronised video sequences were captured at 25Hz PAL resolution (720×576) progressive scan with Sony DXC-9100P 3-CCD colour cameras. Intrinsic and extrinsic camera parameters were estimated using the public domain calibration toolbox [7].

The rendering software was implemented using OpenGL, and tests were performed on an AMD 3100+ Sempron with 1GB RAM and an nVidia 6600 graphics card. The eight camera scene was rendered interactively at 28 frames per second for novel viewpoints, though this could be much improved by using hardware based view-dependent rendering. Projective surface refinement takes approximately twenty minutes to refine eight depth maps for one frame.



(a) Original colour (from different viewpoint)

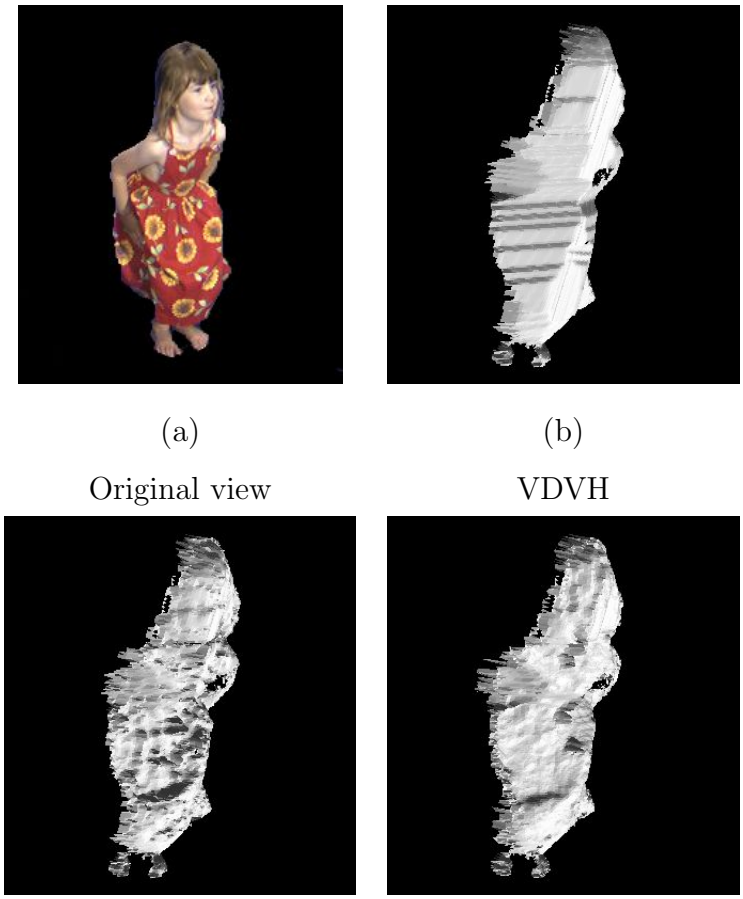
(b) VDVH



(c) Global optimisation

(d) Global optimisation constrained by rims

Figure 5.7: Comparison of visual hull, global refinement and refinement with rim constraints ((a) taken from a different angle to the surfaces, to provide a better view of the colour)



(a) Original view

(b) VDVH

(c) Local refinement

(d) Proposed method

Figure 5.8: The results of this method compared to a previous local refinement method. Image (c) shows the depth artefacts associated with local refinement, whereas the global refinement in (d) produces a smooth surface.

5.4.1 Comparative Evaluation

Figure 5.7 displays a comparison of view-dependent visual hull and optimisations with and without silhouette contour constraints. As can be seen from Figure 5.7(a) there is not much variation in surface appearance, and the optimisation without silhouette constraints over-refines the surface (Figure 5.7(c)). Figure 5.7(d) shows the result after adding rims to constrain the problem: the surface regains its original shape plus refinement.

The images in Figure 5.8 show the difference between the proposed method and work previously demonstrated (Chapter 4), using the eight camera studio setup. Figure 5.8(c) displays the result of a local refinement performed on inconsistent areas of the surface, to produce consistent colour when transitioning between views. Figure 5.8(d) shows the reconstruction proposed using the presented approach which eliminates the depth map spikes and resulting render artefacts. The high variation in surface normal in Figure 5.8(c) makes this surface unsuitable for relighting, unlike the method proposed in this chapter which produces a consistent surface with fewer depth artefacts.

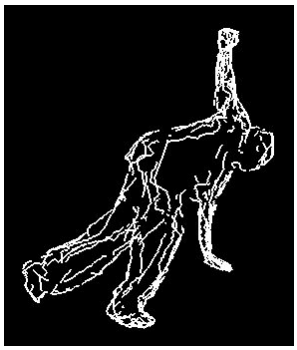
Results of the different stages of the method are shown in Figure 5.9. The refined mesh is a more accurate representation of the surface, as can be seen in the rendered shape. The VDVH in Figure 5.9(b) gives a coarse shape approximation, while the refined shape constrained by the rims in Figure 5.9(d) is a more accurate approximation of surface shape. The surface was slightly over-refined around the torso area (Figure 5.9) due to the lack of rims in that region to constrain the optimisation. Results of the graph cut can be improved by varying the smoothness multiplier or altering the size of the stereo window.

(a)

Original view

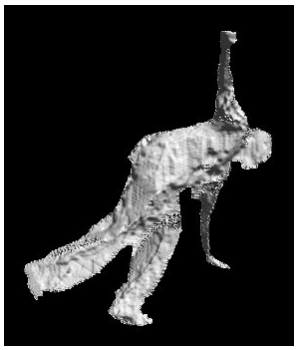
(b)

VDVH



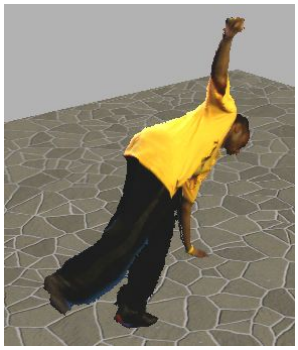
(c)

Visual hull rims



(d)

Refined



(e)

Virtual viewpoint

Figure 5.9: Different stages of the refinement: VDVH is constructed from all views (b), rims are recovered (c) and the VDVH depth map refined in projective ray space (d). A rendered virtual view is shown in (e).

5.4.2 Interactive Free-Viewpoint Video

Figures 5.14 and 5.15 show novel rendered views of a person using an eight camera studio setup from 45° views. The virtual viewpoints in Figure 5.14 are at the midpoint between two cameras, and show a static actor. The novel view in Figure 5.15 is fixed and the images show a dynamic sequence of the actor dancing. The rims for the visual hulls were recovered using an $8cm^2$ 3D stereo window, and stereo scores for the depth map optimisation used 9×9 windows on the original images. This window size was chosen instead of something larger due to the wide baseline of the cameras in the studio. The results images demonstrate the high quality of the rendered views, correctly reproducing details of the face and wrinkles in the clothing, from a limited set of cameras in a complete circle surrounding the scene.

5.4.3 Ground Truth Comparison

The missing view test setups, described in Section 3.11.3, are used to evaluate the quality of the novel rendered views using global constrained optimisation.

The synthesised views and associated errors for the test from setup 1 are shown in Figure 5.10. As for the previous tests, the quality of the novel view is not comparable to captured video due to the wide baseline. However the synthesised views using this technique generate a more consistent and detailed image, of a higher quality than the other methods applied to this test. The global optimisation creates a smoother surface which allows for novel views further from the original views. The refinement operation in the previous chapter produces higher quality images with a smaller baseline, as shown in Figure 5.12. The error intensity images of the three techniques are shown in Figure 5.11, where the global optimisation shows a slight improvement in colour over local refinement and visual hull.

For the first frame, the rms error of the VDVH is 0.098; the error for reference view refinement is 0.097, and the error for the current method is 0.099. Quantitatively there is no improvement via refinement, and so the slight visual quality improvement must be offset by larger errors in other regions.

The results from the missing view test for setup 2 are shown in Figures 5.12 and 5.13. The synthesised views produced are comparable to video quality. Compared to the results of local refinement, the features are less sharp (due to rendering via multi-textures and not via consistent colour, as the local refinement used), but the overall quality is comparable. The error intensity images show a definite improvement over local refinement for the third frame in the test set.

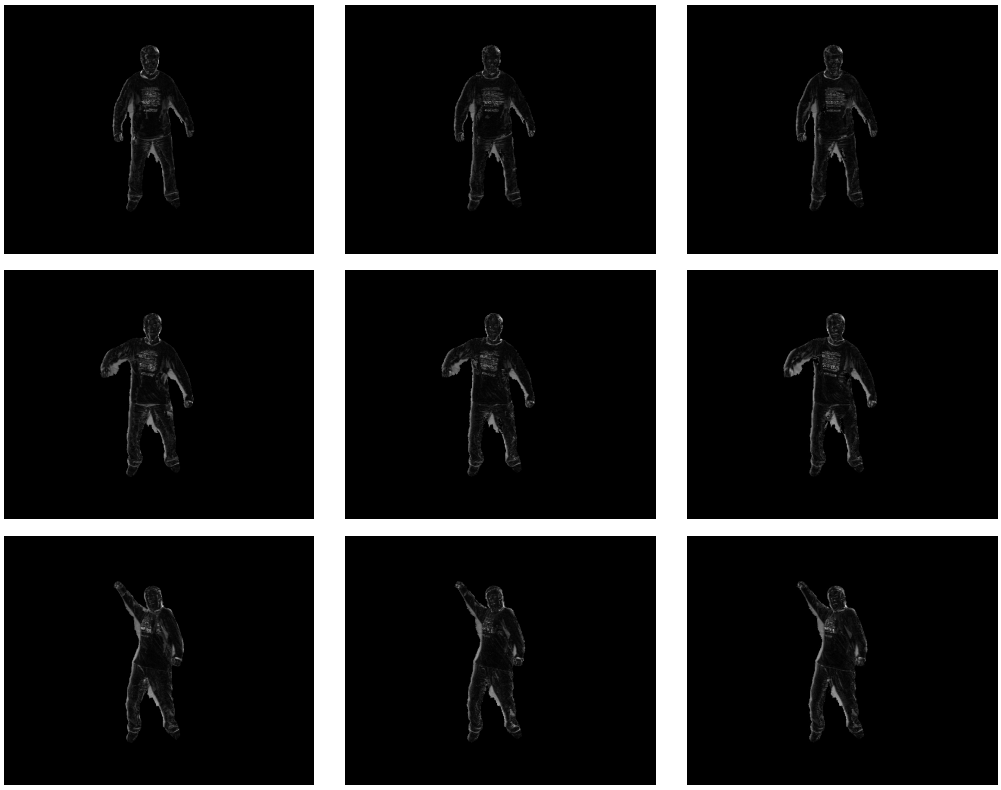
For the first frame, the rms error of the VDVH is 0.098; the error for reference view refinement is 0.085; the error for global refinement is 0.093. Quantitatively the reference view refinement produces the best result, and the global refinement provides a significant improvement over the visual hull.

5.5 Conclusions

Refinement of view-dependent surfaces in projective ray space for application in free-viewpoint video has been presented. The method narrows the search space for refinement using the VDVH allowing the use of wide baseline views. Rims are recovered using silhouette contours from the original views by constructing a graph optimisation problem from the boundary of the VDVH. Surface refinement is formulated as a graph optimisation problem in projective ray space with rim constraints from all views. Results demonstrate that using rims as constraints reduces artefacts due to excessive refinement in unconstrained global optimisation. Multiple view image+depth is used to represent the reconstructed scene by adding a depth channel to the captured images.

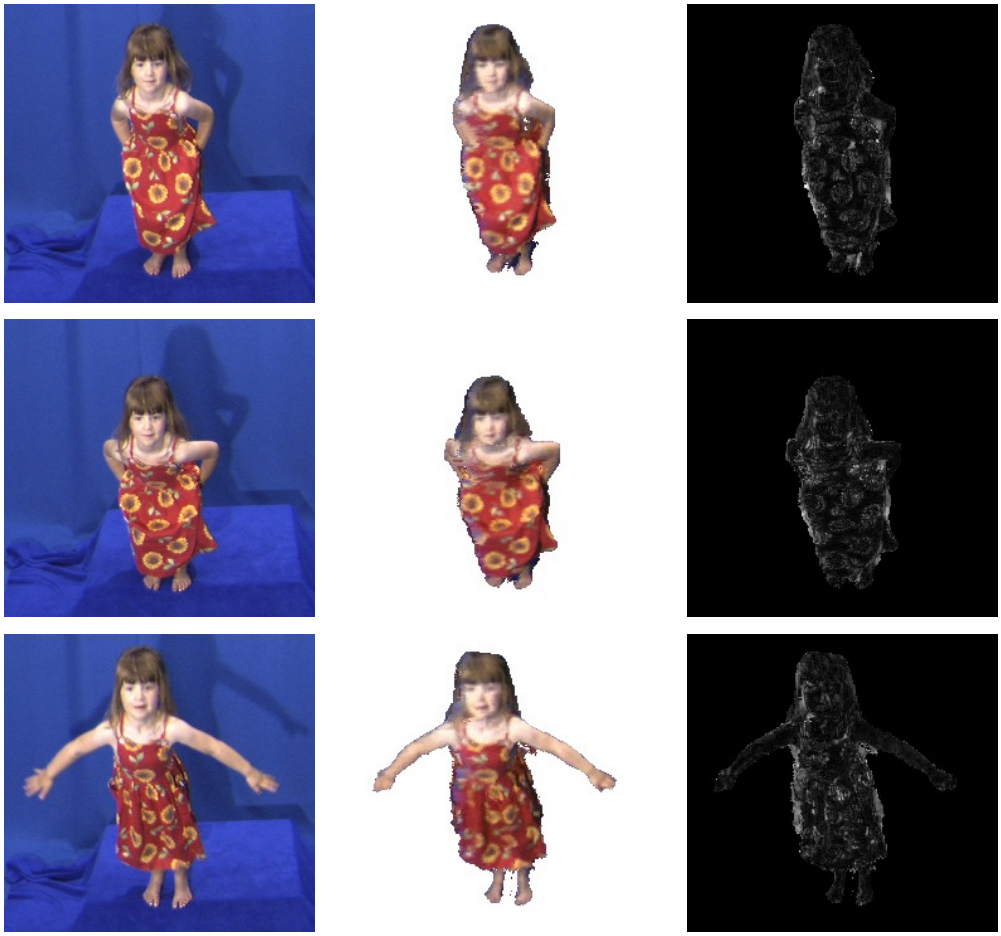
(a) Original (b) Refinement (c) Error

Figure 5.10: The images above are three frames of a sequence from a camera which was removed from processing to be used as ground truth. (a) shows the original images, the synthesised view via global constrained refinement is shown in (b), and the error intensity image is shown in (c).



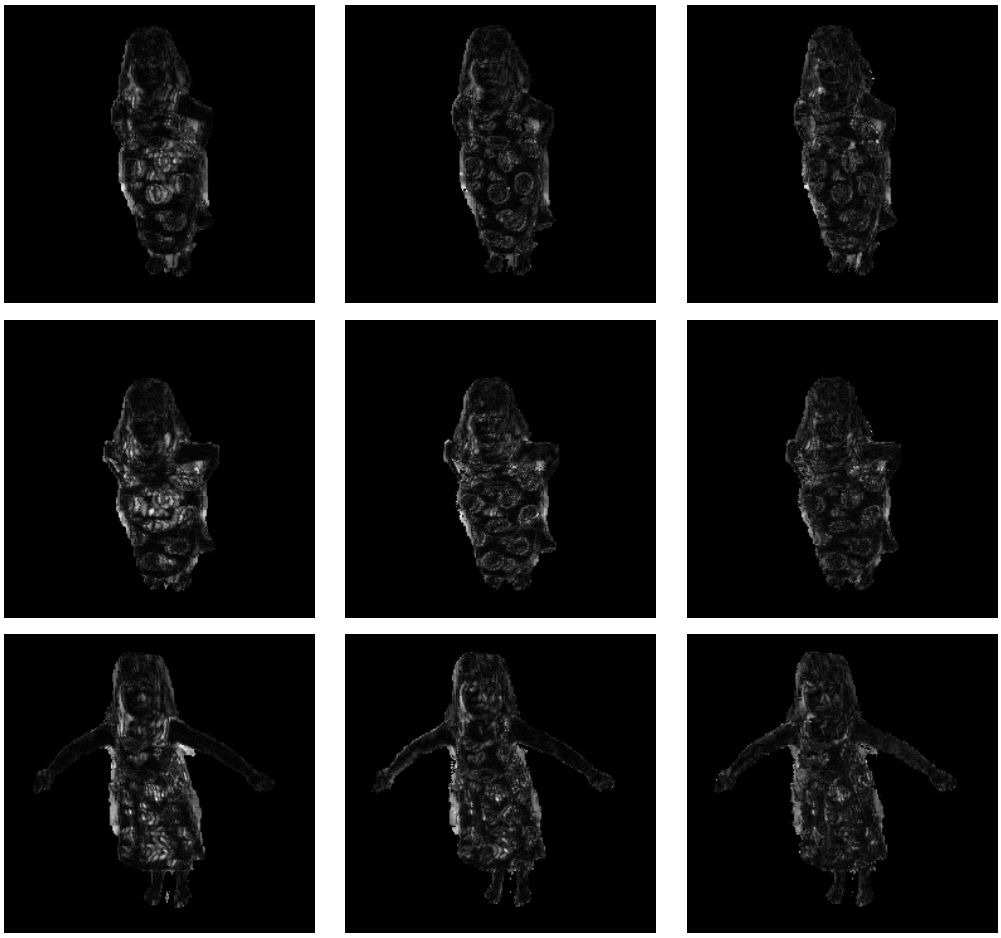
(a) VDVH Error (b) Reference Error (c) Constrained Error

Figure 5.11: The images above are the error intensity images from Figures 3.16, 4.11 and 5.10. (a) shows the error with the ground truth of the synthesised view via VDVH, (b) shows the error with reference view refinement, and (c) shows the error with global constrained optimisation.



(a) Original (b) Refinement (c) Error

Figure 5.12: The images above are three frames of a sequence from a camera which was removed from processing to be used as ground truth. (a) shows the original images, the synthesised view via global constrained refinement is shown in (b), and the error intensity image is shown in (c).



(a) VDVH Error

(b) Reference Error

(c) Constrained Error

Figure 5.13: The images above are the error intensity images from Figures 3.17, 4.14 and 5.12. (a) shows the error with the ground truth of the synthesised view via VDVH, (b) shows the error with reference view refinement, and (c) shows the error with global constrained optimisation.

Free-viewpoint video is rendered at above 25Hz on consumer graphics hardware allowing interactive viewpoint control. Results for a wide baseline studio setup have demonstrated the high quality images possible with this approach. Detailed surface areas in the clothing and face are accurately reproduced in the rendered results.

The work could be improved by adding the concept of uncertainty to the rims to account for calibration and matting errors. For pixel chains where no detailed features exists or visibility of the intervals from the cameras is low the extracted rim will not be reliable. An additional score could be added to the rims in the global refinement representing the reliability of their location. Further work is needed to optimise the boundary of the refined surfaces to allow for smoother blending of the image+depth representation.

Figure 5.14: Virtual views rendered around a static subject, each view at the mid-point between two existing views (with a 45° baseline)

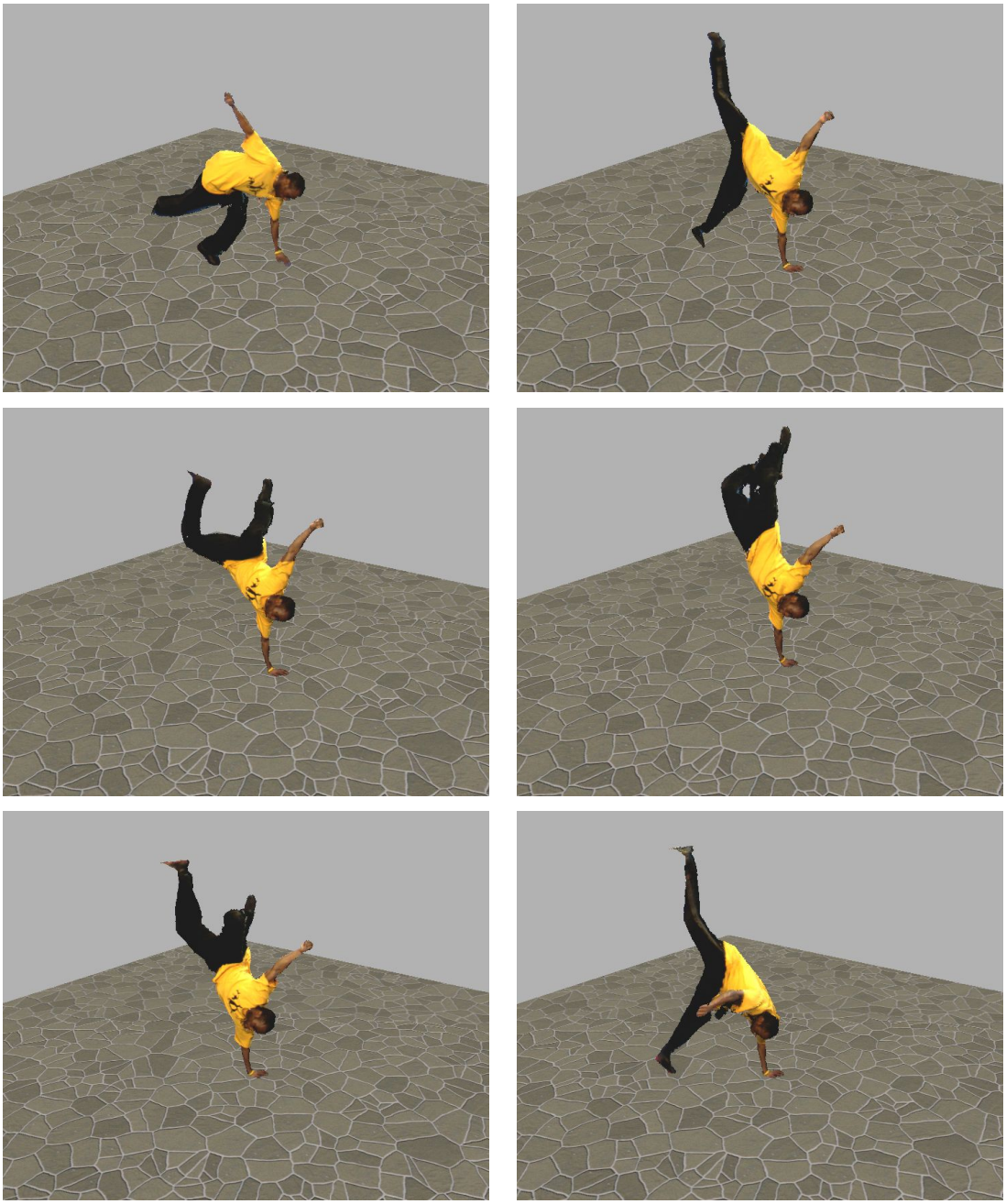


Figure 5.15: Novel rendered views from a static viewpoint for a dynamic scene, illustrating the high quality of this method (with a 45° baseline).

Chapter 6

Safe Hulls

> "Saladin Chamcha ... had even sprouted, from the base of his spine, a fine tail that lengthened by the day and had already obliged him to abandon the wearing of trousers; he tucked the new limb, instead, inside baggy salwar pantaloons from [his landlady's] generously tailored collection."
>
> **Salman Rushdie**, *The Satanic Verses*

This chapter presents the *safe hull*, a novel contribution to the visual hull literature which overcomes inaccuracies of the visual hull by removing phantom volumes. Consequently the visual quality of novel views rendered using the safe hull as a proxy surface is improved by eliminating visual artefacts. This is achieved without increasing the number of cameras or using heuristic methods.

The goal of this research was to increase the reliability and accuracy of the initial surface used for free-viewpoint video. The novel contribution of this chapter is an additional constraint on visual hull construction which produces visual hull surfaces guaranteed to contain the foreground. There are two main applications for this technique:

- Isolate all definite foreground (safe) regions and use only these for refinement and rendering. This is especially useful for human bodies in dynamic scenes (demonstrated in Section 6.3).
- Identify unsafe regions and mark them for further processing, such as feature matching to categorize them as foreground or background.

The surface produced from a visual hull reconstruction comes with two major problems. Silhouettes are unable to represent concavities of objects and so neither can the visual hull (e.g. it could not reconstruct the inside surface of a coffee mug). Research has concentrated particularly on this problem, especially in free-viewpoint video, where colour matching and model fitting are used to recover more accurate shape.

The second problem, which the research in this chapter addresses, is phantom volumes: surfaces produced in the reconstruction that do not represent objects in the scene. They are a product of multiple or non-convex objects, illustrated in Figure 6.1(a), and are consistent with the original silhouettes. The perceived realism in a synthesised view can be negatively affected by the odd shapes they form, as shown in Figure 6.7(c). These often have to be removed by hand, or through heuristic methods which may incorrectly remove surfaces belonging to a foreground object or not remove phantoms at all.

Previous approaches have attempted to remove phantom volumes by adding more cameras[8], however this does not guarantee a surface with reduced artefacts. A common issue for reconstruction of people is extra limbs, such as a 'tail' (Figure 6.5(c)), that appear at the location where two surfaces join (e.g. the 'tail' is produced where the legs meet the torso). This particular type of artefact appears as a connected phantom volume, a surface which does not represent a foreground object but is connected to a surface which does. Removal of this volume via visual hull requires a camera positioned to look directly between the legs at all

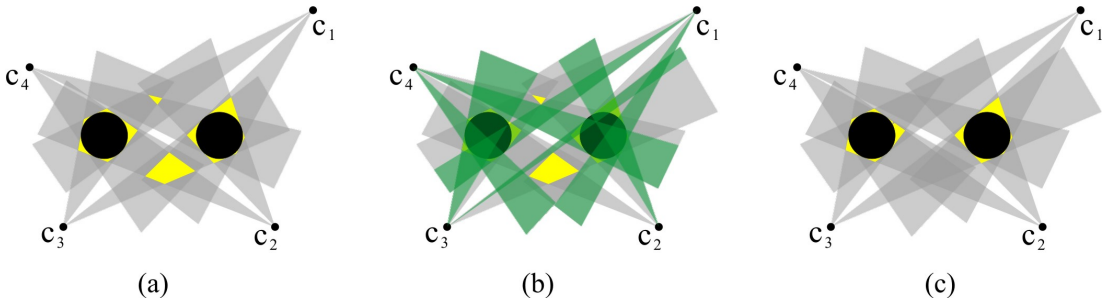


Figure 6.1: Phantom volumes are caused by multiple objects in the scene, shown in (a). The black circles represent scene objects, the gray areas represent silhouette cones from the cameras and the yellow shapes represent the result of visual hull reconstruction. The green areas in (b) represent the safe zones defined by the cameras, and the safe hull reconstruction is shown in (c), with phantom volumes removed.

times, which is impossible for a dynamic subject.

Although additional cameras can reduce the size and number of phantom volumes, studios generally do not have many cameras due to time and financial constraints, therefore research into free-viewpoint video is often targeted toward a minimal number of well-placed cameras. This highlights the importance of a solution to phantom volumes, since it increases the quality of the results without requiring additional cameras.

The research presented in this chapter illustrates how to identify volumes in three dimensions which are part of the foreground i.e. safe zones which definitely do not contain phantom volumes, and to reclassify the remaining occupied space as unsafe. The unsafe space can be removed completely, therefore guaranteeing removal of all phantom volumes (including those connected to the subject), it can be processed further, for example using colour constraints to identify foreground, or it can be rendered differently (such as using transparency).

The novel contributions of this chapter were published in *Safe Hulls*, Conference

on Visual Media Production, 2007 [59].

6.1 Alternative Techniques

Other approaches to free-viewpoint video do not use visual hull and so do not suffer from phantom volumes, but are more constrained. Carranza et al. used a model-based approach with silhouette initialisation[13], which requires prior knowledge of the captured subject and so reconstruction of an arbitrary scene (such as the juggling example) is not possible. The novel view system presented by Zitnick et al.[95] simultaneously estimates image segmentation and stereo correspondence to produce video quality virtual views, but is restricted to a narrow baseline camera setup (8 cameras over 30°). Goesele et al.[32] present a multi-view stereo reconstruction system that produces high quality surfaces with a large number of narrow baseline views, which is prohibitive for dynamic scenes. Adopting the visual hull as a basis allows for arbitrary dynamic scenes to be reconstructed from a relatively small number of widely spaced cameras, provided the foreground and background can be separated.

The problem of removing phantom volumes from a visual hull reconstruction has largely been ignored in previous research. The addition of more cameras may reduce the problem but artefacts still occur. Crowd surveillance techniques have applied visual hull with temporal filtering and heuristic methods based on size to remove phantom volumes[90]. These approaches can be unreliable, for example in juggling (Figure 6.7) the balls may be removed by a threshold on size, and temporal filtering would not work on a connected phantom volume (such as the tail in Figure 6.5).

Utilising colour information for phantom volume removal can also lead to errors in surface construction. If a refinement operation is allowed to remove surface which is considered inconsistent, there is a risk of leaving phantom volumes in the

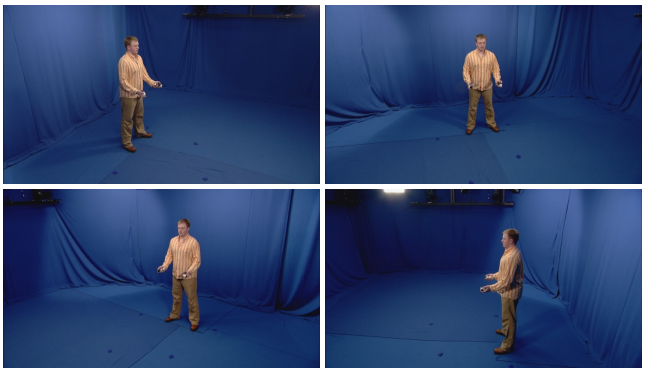
scene and also of removing real surface. If the scene contains repetitive texture or similar colour across a region, colour consistency tests will not remove phantom volumes. Regions which are visible by zero views or one view due to occlusion cannot be tested for consistency, and false colour matches may remove real surface volumes. Using the method presented here colour is not required to produce a visual hull surface without phantom volumes.

6.2 Safe Hulls

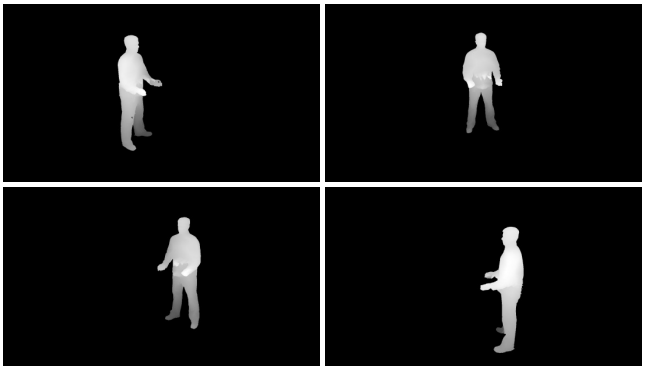
This section presents a novel method for detecting real volumes in a visual hull reconstruction, using a theoretical basis to construct the *safe hull*. The construction of the safe hull is accomplished via a two-pass algorithm, where the full visual hull is constructed and analysed to supply information about the original images. The information is used to define *safe zones* in the original images: regions known not to back-project from the camera centre to phantom volumes. A second construction takes place, similar to visual hull but incorporating the safe zones so that all phantom volumes are excluded. The final result is a scene partitioned into definite foreground, definite background and a middle ground which may contain both phantom and real volumes.

The *visual hull* is constructed from the set of captured images $\mathcal{I} = \{\mathcal{I}_n : n = 1, \ldots, N\}$ using the corresponding set of silhouettes $\mathcal{S} = \{\mathcal{S}_n : n = 1, \ldots, N\}$ produced via foreground extraction, where N is the number of calibrated views. The process and notation are both described in detail in Chapter 3. A phantom volume is an artefact of visual hull reconstruction where a surface is created where no scene object exists. This is due to multiple objects in a scene, or a non-convex object causing occlusion. Examples of a phantom volume are shown in Figure 6.1(a).

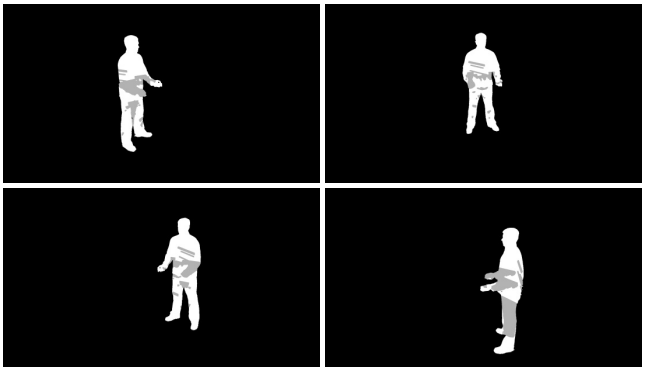
Using a method which constructs a global representation to compute the multi-



(a) original images



(b) VDVH depth maps



(c) safe zones (white)

Figure 6.2: From the original images, (a), a depth image is produced for each, (b), and analysed to identify safe zones (white) and unsafes zone(grey), (c).

layer depth images would require us to find the intersection of rays from each camera with the resulting surface, which involves multiple resampling steps. The VDVH, a local representation, was chosen to form the basis of this technique because it efficiently produces an exact sampling of the visual hull surface with no additional quantisation, and the multi-layer depth image it produces directly represents the intervals of the visual hull with respect to a particular view. This is required for determining the location of phantom volumes, as described in the following section.

6.2.1 Foreground Detection

The algorithm relies upon the ability to detect regions in an image which definitely do not contribute to a phantom volume and are therefore part of the foreground. The following results demonstrate how this can be accomplished. This first result is the basis of the method:

Theorem 6.1. *Given the set of pixels $Q = \{q : q \in \mathcal{I}_n\}$ which lie on the projection of a phantom volume in image $\mathcal{I}_n$ and the multi-layer depth image $\mathcal{V}_n$ produced using VDVH, every depthel in the set $D_n = \{d(q) : d(q) \in \mathcal{V}_n, q \in Q\}$ has more than one interval.*

Proof. Define the set of pixels $P = \{p : p \in \mathcal{S}_n\}$, and the set of rays $R = \{\boldsymbol{r}(p) : p \in P\}$ through P from the camera centre $\boldsymbol{c}_n$, each ray $\boldsymbol{r} \in R$ has at least one interval which lies inside the real object described by $\mathcal{S}_n$. Phantom volumes are consistent with all views' silhouettes (from the visual hull definition), therefore they exist inside the silhouette cones for real objects. Now define the set of pixels $Q = \{q : q \in \mathcal{S}_n,\ q \text{ corresponds to a phantom volume}\}$ (so $Q \subseteq P$). Since each depthel $d \in D_n$ already has at least one interval for the real object, the phantom volume intersected by $\boldsymbol{r}(q) \in R, q \in Q$ introduces at least one more. Therefore d must have a minimum of two intervals. □

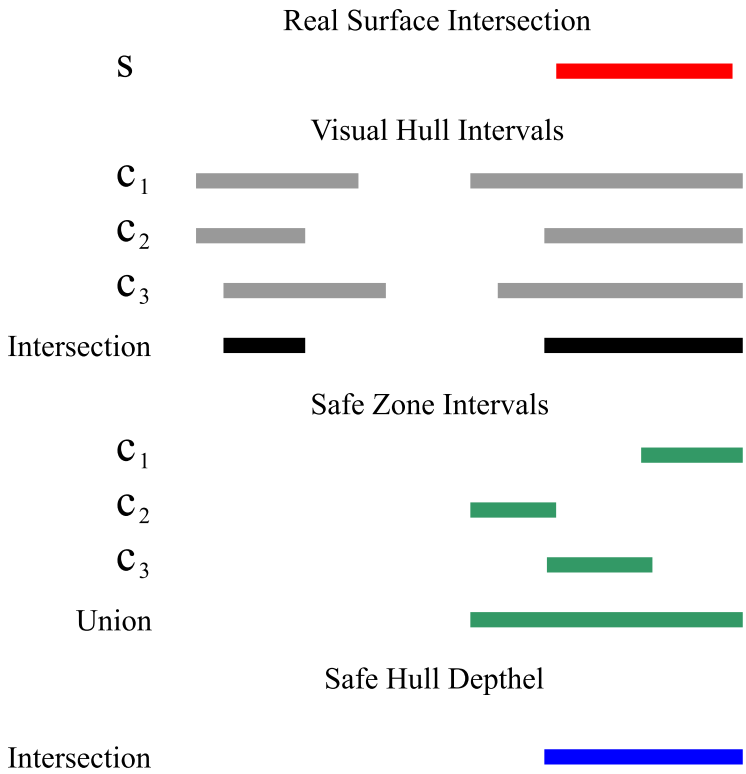


Figure 6.3: The grey lines represent the intervals from three cameras projected onto a virtual ray, and the visual hull represented below them as the intersection of all three. The green lines represent the safe zone intervals from these cameras, and below them their union to define which volumes are definitely not phantom. The blue line shows the result of an intersection of the visual hull depthel with the safe zone depthel: the safe hull depthel. Notice that the object to the left has been removed, and may have been a phantom.

Theorem 6.1 does not work in the reverse: regions of the image with multiple intervals are not necessarily phantom volumes, they could for example be an arm occluding the body. However, we can use it to deduce the following result:

Corollary 6.2. *Depthels which have only one interval represent a real volume and do not contain phantom volumes.*

Proof. Theorem 6.1 states that depthels containing phantom volumes must have more than one interval, so it follows that depthels with one interval describe a real volume. □

This allows us to partition each image in $\mathcal{I}$ into three regions: we can mark regions of $\mathcal{I}_n$ with more than one interval in $\mathcal{V}_n$ as 'unsafe zones', regions with only one interval as 'safe zones', and the rest remains as background. This leads to the important result which allows us to remove phantom volumes:

Observation 6.1. Any point in the visual hull whose projection lies inside a safe zone of a single image does not contain a phantom volume.

This is important because it shows that for any point in the volume, only *one* view with this point's projection in the safe zone is required for it to be considered part of a real volume, as shown in Figure 6.1. There is no need for all views to agree; exactly the opposite concept to the visual hull.

Since all views have their own safe zones, the union of visual hull volumes corresponding to each forms the safe hull and completely eliminates phantom volumes. The volumes in the visual hull excluded from the safe hull contain all phantom volumes and parts of the object which did not project to safe zones (assuming that the calibration and segmentation are correct). These can be examined manually for phantom volume removal, or further processed, for example using colour consistency as a constraint.

The algorithm for safe hull construction is as follows:

1 Construct the visual hull

2 Find safe zones in the original images

   (a) Find intersections of rays from occupied pixels in original views with the visual hull surface

   (b) Partition occupied pixels in the silhouette into safe and unsafe zones (mark pixels with one interval as safe)

3 Construct safe hull

   For a given point in the visual hull volume, accept it if it lies in a safe zone in at least one camera. Otherwise reject it.

6.2.2 Safe Zones

The first step is to construct the VDVH with respect to each real viewpoint, as explained in Section 3.9. The result is a multi-layer depth image containing the set of intervals inside the visual hull surface, which immediately gives us the required form for partitioning the foreground into safe and unsafe zones. A safe zone is made up of the pixels in the VDVH whose rays contain a single visual hull interval. The depth images which result from VDVH construction are shown in Figure 6.2(b). Pixels with depthels of only one interval are marked as safe, and every other occupied pixel marked as unsafe. Figure 6.2(c) shows the safe zones as white areas and the unsafe zones as grey areas.

The safe hull cannot be constructed by removing the unsafe zones from the silhouette, since these regions may correspond to a volume that has been declared safe by another camera. Instead the unsafe zones are used to determine the validity of points in the volume, or in the case of the VDVH, to select the correct interval.

6.2.3 Safe Hulls

The VDVH is constructed by casting rays out through the pixels of the virtual image, projecting them onto the real view's images and finding the intervals where the projected rays are inside the silhouette. The intervals are projected onto the original rays from the real view, and the mathematical intersection of intervals on each ray provides the depthel of the visual hull for that pixel (shown in the top section of Figure 6.3).

Safe hulls are constructed in a similar way to visual hull, with an additional selection process. The intersections of a projected ray with a silhouette and the intersections with the safe zone in that image can be found simultaneously. The safe zone intervals are projected onto the original ray as well as the silhouette intervals. As for visual hull, the mathematical intersection of the silhouette intervals gives the depthel of the visual hull. The depthel for the safe hull is provided by computing the mathematical intersection of the visual hull depthel with the union of all cameras' safe zones intervals (illustrated in Figure 6.3).

The equivalent process in a volumetric formulation is to test that a voxel is consistent across all silhouettes and that it appears in at least one safe zone, and should therefore be accepted.

Figure 6.7(a) displays a visual hull reconstruction of a person, with phantom volumes in front of the body, between the body and the arm, and around the inside of the legs. Figure 6.7(b) shows a safe hull reconstruction with these shape artefacts removed.

6.3 Results

This section presents results which demonstrate the effectiveness of safe hull construction, and how it enhances the realism of a virtual scene. Four different

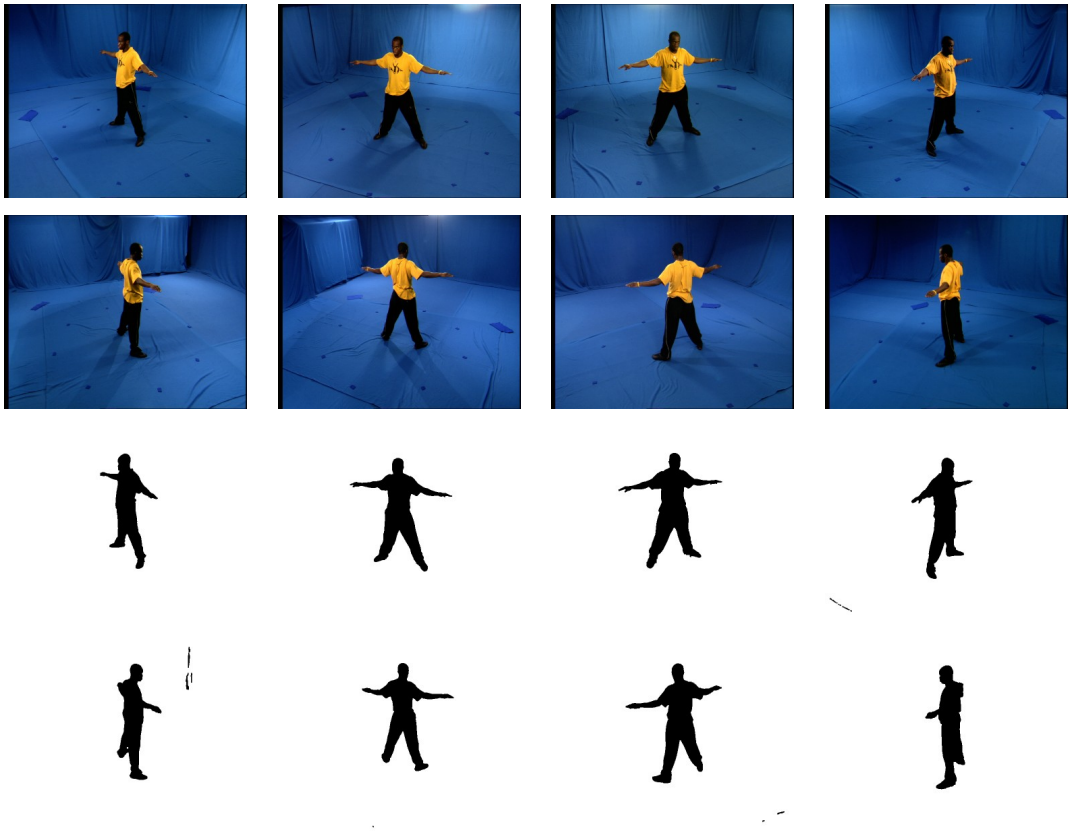
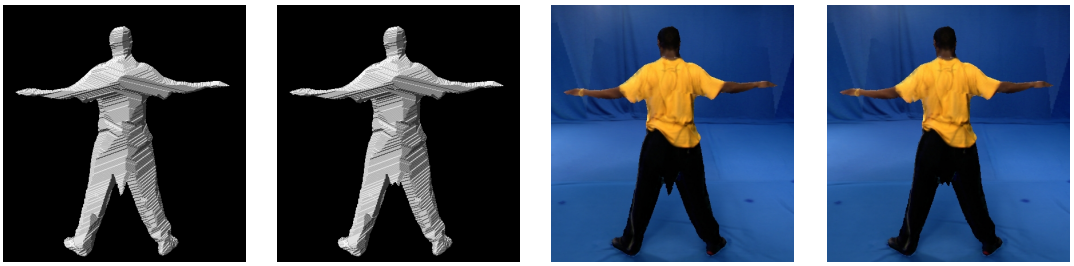


Figure 6.4: The original images and silhouettes from a single frame of a studio capture against a blue screen.



(a) VDVH surface (b) Safe hull surface (c) VDVH rendered (d) Safe hull rendered

Figure 6.5: This example illustrates the most common situation for phantom volumes to appear when capturing humans. This is a connected phantom volume, and often appears between the legs or under the shoulders. The quality of a synthesised view (c) is dramatically decreased when a subject spontaneously grows a 'tail', and once removed the image quality is improved (d).

acquisition systems were used for testing:

Setup 1 Eight equally spaced cameras in an approximate circle of radius $6m$, baseline 45°, each capturing at 25Hz SD resolution (720×576) progressive scan. The original images from a capture from this setup can be seen in Figure 6.4.

Setup 2 Eight cameras in an arc of 180° pointed towards the subject approximately $4m$ away, each capturing at 25Hz HD resolution (1920×1080) progressive scan. The original images from a capture from this setup can be seen in Figure 6.6.

Setup 3 A Fuji s6500fd digital camera recorded single images at 2048×1536 for analysing static scenes.

Setup 4 The synthetic setup from Chapter 3 is used to evaluate the surface quality. The silhouettes from this setup can be seen in Figure 3.18.

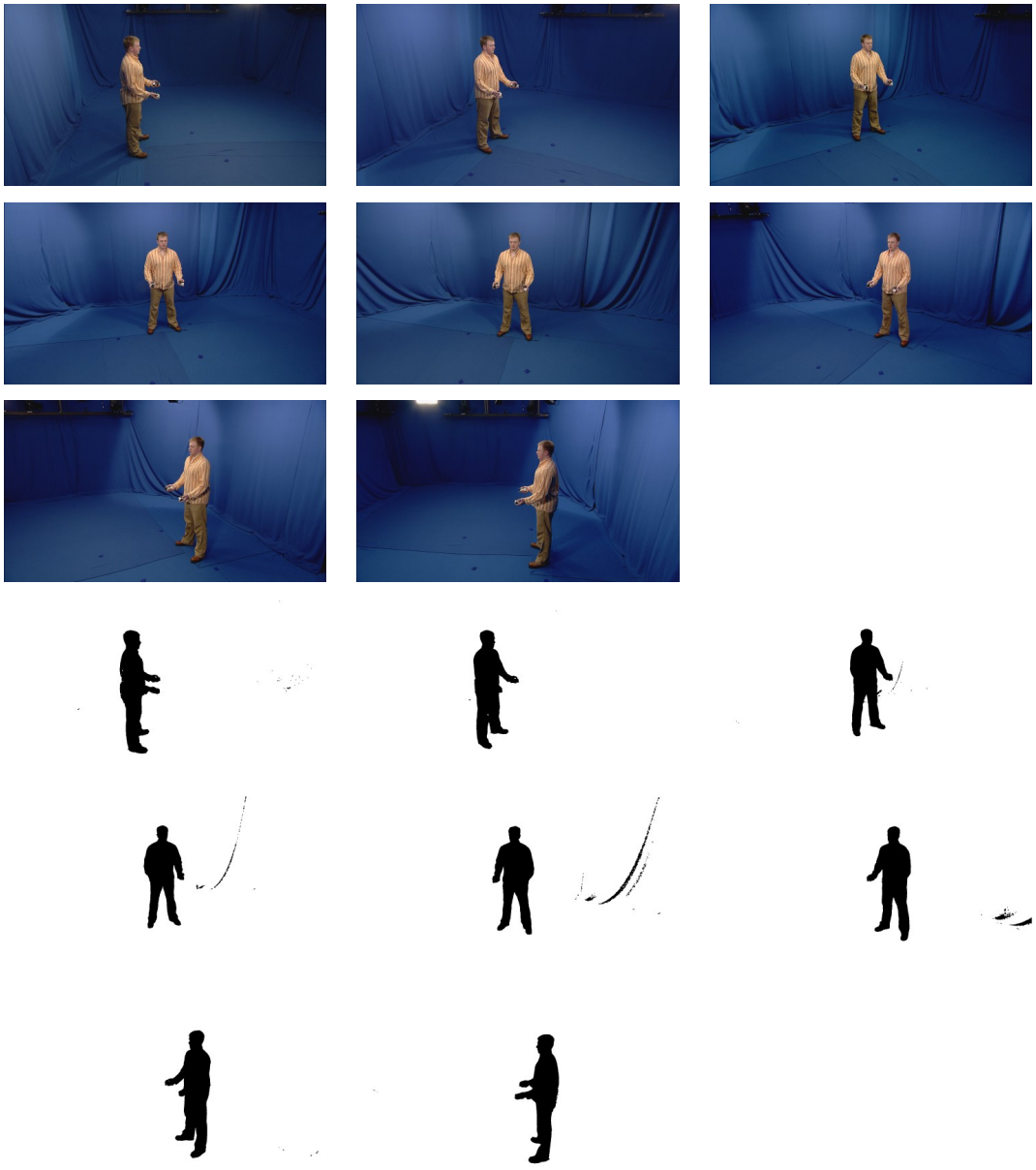
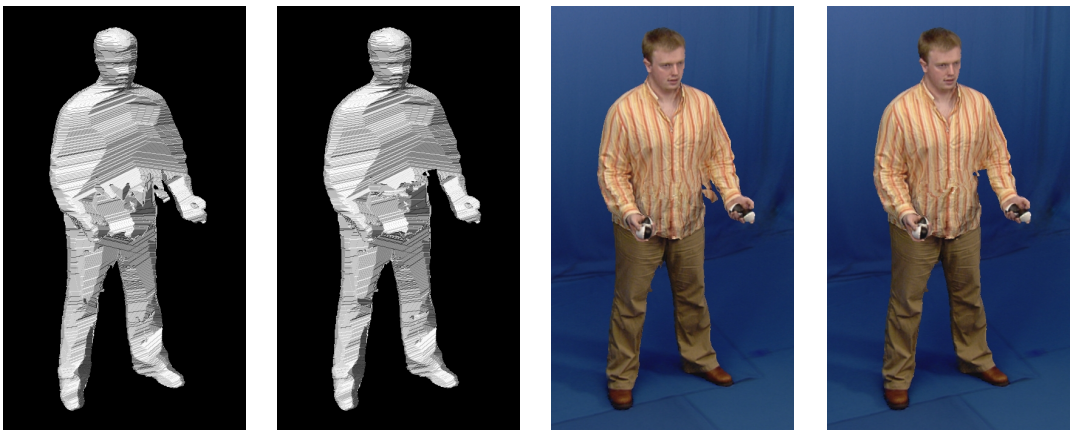
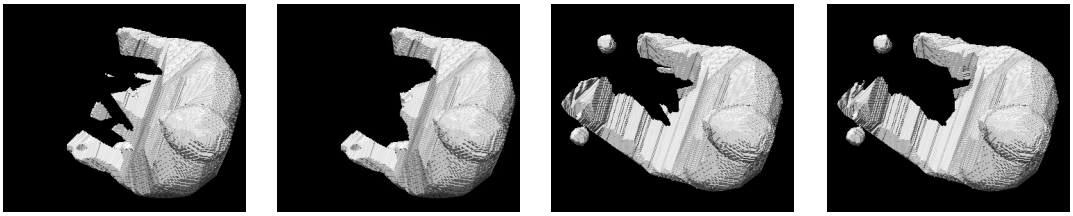


Figure 6.6: The original images and silhouettes from a single frame of a studio capture against a blue screen.



(a) VDVH surface | (b) Safe hull surface | (c) VDVH rendered | (d) Safe hull rendered



(e) VDVH surface | (f) Safe hull surface | (g) VDVH surface | (h) Safe hull surface

Figure 6.7: Top row: Taken from a juggling sequence, (b) shows the surface of the object after the phantom volumes from (a) have been removed. The rendered views of these surfaces are shown in (c) and (d). The quality of the synthesised view is severely affected by the presence of a phantom volume between the arm and body in (c). As a result of safe hull construction, (d) is much more realistic. Bottom row: Surfaces viewed from above: image (f) demonstrates removal of entire phantom volumes from (e); image (h) shows the safe hull reconstruction of (g), with the juggling balls intact - a heuristic solution based on size may have removed them. This would also be more difficult to produce using a model-based method.

For setups 1 and 2 intrinsic camera parameters were estimated in both cases using the public domain calibration toolbox [7] and the extrinsics via wand calibration[62]. Setup 3 was calibrated using the GML Calibration Toolbox[84]. Tests were performed on an AMD 3100+ Sempron with 2GB RAM and results rendered using OpenGL on an nVidia 6600 graphics card.

Figure 6.5 shows the most common problem with multiple view video capture. This frame is from a sequence captured in Setup 1, and illustrates the problem of connected phantom volumes. These appear generally at the meeting point of two objects, and form a cone shape. Safe hull reconstruction removes these since they do not appear in any safe zone, and the generated result is of a higher visual quality. The slight stump in Figure 6.5(d) remaining in the safe hull belongs to a safe zone and is part of an interval in the visual hull surface containing the foreground.

The images shown in Figure 6.7 are produced from sequences captured in Setup 2. The visual hull produced a surface with phantom volumes in front of the person, between the body and the arm, and around the legs. Figure 6.7(b) shows the safe hull with these shape artefacts removed. Figure 6.7(e-h) shows a top-down view demonstrating safe hull removal of the phantom volumes, and the juggling balls left intact after safe hull reconstruction. Heuristic approaches based on size or temporal surface shape may remove the balls from the reconstruction. This also demonstrates a situation where a model-based approach would fail to represent the entire scene.

From Figure 6.5(d) and 6.7(d) it can be seen that the synthesised novel view is improved after safe hull construction.

Setup 3 was used to capture the images in Figure 6.9. This illustrates a worst-case scenario where there are very few original images (three in this case) of a subject with multiple surfaces and each view has an occlusion. The visual hull reconstruction in Figure 6.9(a) shows the outcome of multiple occlusions: a large

phantom volume in the centre of the subject, and some smaller phantoms at the top edges. Figure 6.9(b) shows the improvement the safe hull reconstruction has made, where only definite foreground remains and the phantoms have been removed. Some small parts of the foreground surface were also removed in the process, since the original views did not provide sufficient coverage for all real surfaces to be included in safe zones.

The computation times for each of the three tests described so far are presented in Figure 6.8. Virtual view safe hull construction after the VDVH pre-computation is very similar in computation time to normal virtual view VDVH construction. The pre-computation of VDVH for all real views takes the most time, and could be improved by using a global representation of visual hull and a hardware implementation for safe zone identification.

The images in Figure 6.10 demonstrate the improvement in surface quality using the ground truth comparison from Figure 3.19 in Section 3.11. The intensity of the images represents the associated error in the surface, and a number of artefacts have been removed in the safe hull reconstruction. The pointed surface regions on the chest, thighs and under the arms have been removed, leading to a surface which better represents the original scene.

The safe hull technique works well for subjects such as humans as shown in Figures 6.5 and 6.7, and also for more complicated objects such as that in Figure 6.9. The results images demonstrate the higher quality of the rendered views using safe hull construction rather than visual hull.

6.4 Conclusions

This chapter has introduced the first known geometric constraint which allows phantom volume removal from visual hull reconstructions. This improves recon-

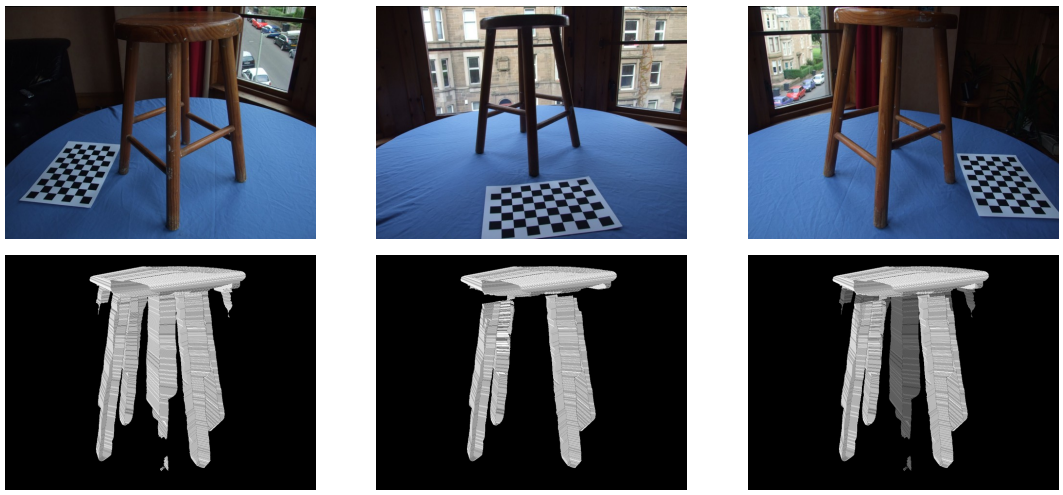
Setup	Resolution (original/virtual)	VDVH Pre-compute (all views)	Safe Hull (virtual view)
1	720×576/720×576	∼16	∼6
2	1920×1080/1280×960	∼48	∼20
3	2048×1536/1280×960	∼24	∼22

Figure 6.8: The approximate time taken (in seconds) to pre-compute the visual hull for all real views and then perform a single virtual view safe hull reconstruction.

struction accuracy and the overall quality of novel view synthesis. The approach presented here uses information from the visual hull and is reliable since it does not use heuristics or require additional cameras to remove shape artefacts. Also it does not rely on photo consistency constraints and can therefore be used on objects of uniform appearance.

The surface produced by the safe hull reconstruction is limited by the number of safe zones in the original images. If there are too few safe zones due to many occlusions and not enough viewpoints then parts of the real surface may be removed. All real surface must be visible against the background in the original images in at least one view to be included in the safe hull. For highly complex objects such as plants with multiple inter-occlusions this may require a large number of views, as would the visual hull. However for many setups, especially those involving people, the safe hull produces good results with a small number of cameras.

For future work further processing of the surface not marked as definite foreground will be investigated. This surface may contain real surface which can possibly be identified by applying feature matching and colour constraints. Safe hull reconstruction requires the rendering of visual hull surface to each original viewpoint and analysis of the intervals, an operation which could be implemented



(a) Visual hull surface

(b) Definite foreground surface

(c) Combined surface

Figure 6.9: The top row shows all original images used for the capture and the bottom row shows a virtual view of the surface. This capture illustrates a worst-case scenario with occlusion causing a large phantom volume to appear, shown in bottom row (a). The result in (b) shows the definite foreground areas, with the phantoms removed and some small sections where no safe zone existed. The final image in (c) shows the definite foreground with the rest of the surface rendered with transparency for comparison. (The braces connecting the legs of the stool were removed by hand during matting.)

efficiently in hardware.

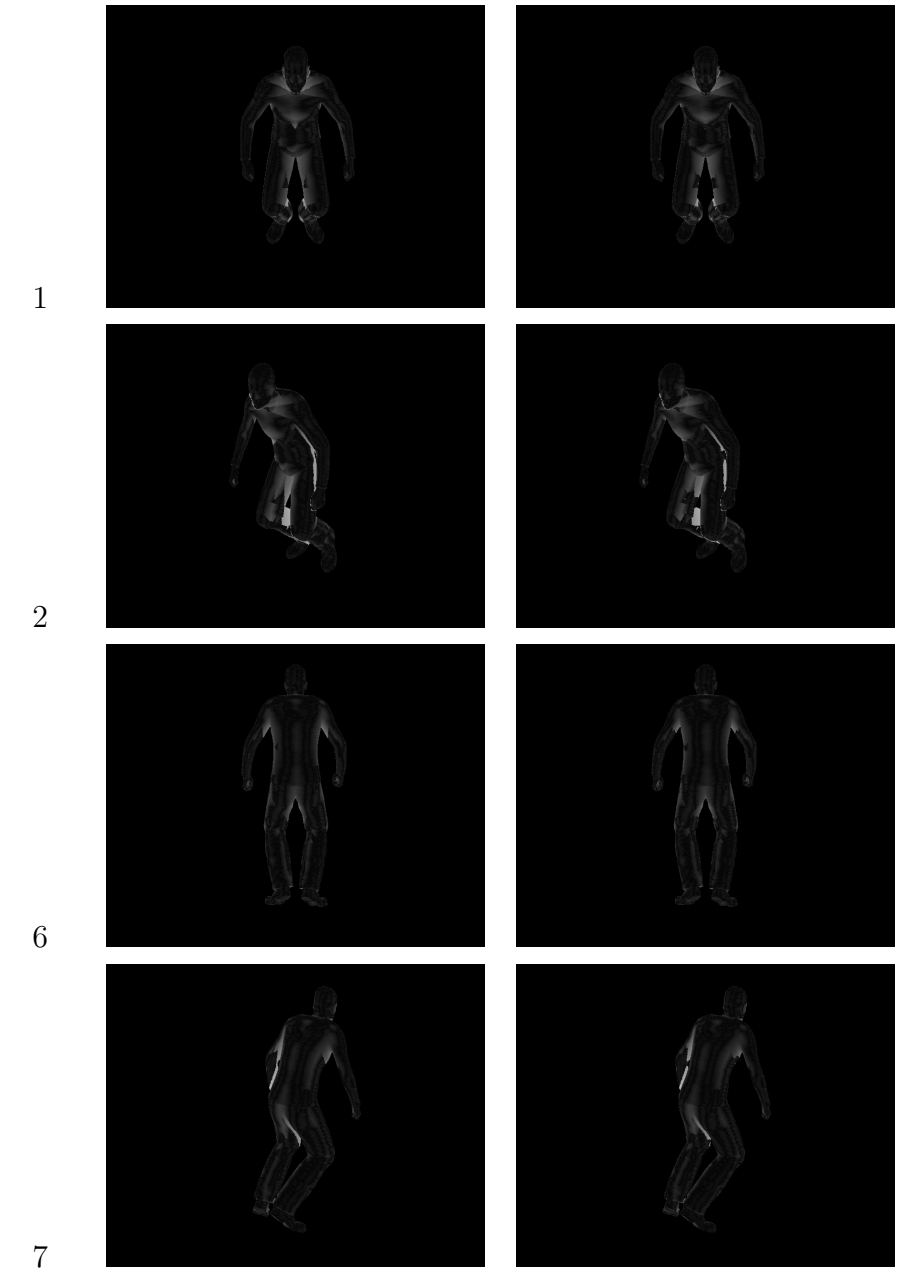


Figure 6.10: Error intensity images for selected views of the VDVH and the safe hull with respect to the ground truth (shown in Figure 3.19). Surface improvement can be seen on the chest, under the arms and on the thighs, where artefacts have been removed.

Chapter 7

Conclusions and Discussion

The research presented in this thesis has led to several novel contributions to the 3D reconstruction computer vision literature.

The exact view-dependent visual hull (VDVH) presented in Chapter 3 is an efficient method for producing the visual hull with respect to a particular view. The reconstruction is performed in the image domain to increase efficiency: this is done by using the cross ratio to order silhouette intersections across views. The correct intersection is selected using the novel visual hull visible intersection theorem. Working in the image domain also has the advantage that the reconstruction is scale-independent, unlike some approaches that construct a global representation.

The VDVH has been shown to provide a good approximation to the scene surface, with improved surface accuracy over a volumetric visual hull reconstruction. It provides an exact sampling of the visual hull surface without requiring any additional quantisation or other approximations. The VDVH is designed to be constructed for a given viewpoint, and therefore is not suitable for techniques requiring a global representation.

The intermediate view refinement technique introduced in Chapter 4 utilised the

VDVH to allow localised surface refinement using wide-baseline views. Without an approximation such as the VDVH, stereo correspondence across wide-baseline views is a much more difficult problem. This technique optimised a surface for transitioning between adjacent views by constructing the VDVH for this view, identifying surface points with inconsistent colour in adjacent views and refining only these points using stereo correspondence.

This technique was extended to refine surfaces constructed with respect to each real camera instead of using an intermediate view. Reference view refinement (refining the surface generated with respect to a reference view) allows more general camera setups to be used in future, and also has the advantage that all visible surface will be rendered, which cannot be guaranteed for the intermediate view refinement due to possible occlusion. While the intermediate view refinement produced slightly better results at the extreme case (the midpoint between views), the reference view refinement still synthesised high quality virtual images, as shown in comparison to ground truth images.

Since the refinement operation only refines surface points whose colour is inconsistent between adjacent views, the operation is efficient when compared to others. While this technique reduced artefacts, the local refinement introduced depth artefacts in highly refined regions, which are more visible in synthesised video. The following refinement technique was designed to overcome these artefacts by optimising the entire surface.

Chapter 4 also introduced an image+depth representation for rendering view-dependent surfaces. The representation is pre-computed and rendered in real-time for use in an interactive free-viewpoint video application. View-dependent surfaces are individually rendered based on their distance from the virtual view. This has the advantage of using surfaces refined specifically for the virtual between adjacent cameras.

The global refinement presented in Chapter 5 used the VDVH as an initialisation

and then formulated a flow network problem for a refinement operation. This optimisation was designed to produce a continuous surface that was colour consistent between adjacent views so that depth artefacts from local refinement were removed and the quality of the synthesised view increased. Additional silhouette constraints in the form of rims are incorporated to stop the optimisation from over-refining the surface. Results demonstrate the smoothness of the surface compared to that produced by local refinement techniques and the high quality of the synthesised images.

The safe hull, described in Chapter 6, introduced the first geometric constraint on a visual hull reconstruction to guarantee a surface which does not contain any phantom volumes. This has been demonstrated to reduce visual hull artefacts in common studio scenes and therefore increase the quality of novel synthesised views.

The evaluation of the view synthesis algorithms presented in this thesis has shown that it is possible to produce novel views with a quality comparable to captured video. The local refinement technique provides the best results when restricting the viewpoint between existing views. Global refinement produces surfaces with low variation in surface normal which makes them suitable for rendering further away from the original views, as shown in the results of the missing view test with the 72° baseline between cameras.

The research presented in this thesis has advanced the state of the art in high quality novel view rendering from multiple wide-baseline cameras. The exact view-dependent visual hull produces an accurate sampling of the visual hull surface which gives the first approximation to the scene. The approach developed emphasises quality of the produced surface as well as efficiency, and preserves as much information from the original images as possible. Using the visual hull as a constraint allows surface refinement via stereo correspondence between wide-baseline views, to produce a proxy surface capable of synthesising high quality

novel views. Intermediate views are generated using the refined proxy surfaces to generate high quality video sequences with sharp features and visibly reduced artefacts. The safe hull has considerably advanced the state of the art for visual hull, demonstrating for the first time a mathematical based approach to constructing surfaces guaranteed to only contain the real object. This is a considerable improvement over previous heuristic methods, and brings the possibility of reducing the number of cameras required to capture a dynamic scene with multiple objects.

The objective of this work has been to produce novel views with a quality comparable to captured video. While the research presented here has produced high quality output the images still suffer from artefacts. As with all visual hull based approaches, better calibration and matting improves the initial approximation and therefore the final result. The main artefacts in the novel rendered views occur at the border of surfaces defined by the input images (e.g. the edges of a person) and optimisation of the border could reduce artefacts, possibly by using reliable alpha matte silhouettes. A better approach would be to use other cameras to refine the border of the surface in 3D and inject the result back into the original method as an improved silhouette.

There are several exciting directions research can take to extend the work in this thesis. Synthesising views of complicated scenes such as those containing multiple objects will become more important, especially as free-viewpoint video tools become popular in, for example, sports broadcasting. High quality synthesis of complicated subjects such as long flowing hair is still a major challenge, where calibration errors can remove small surfaces and automatic matting techniques do not exist for translucent or transparent objects. As the world moves into an age of ubiquitous cameras, the challenge is to extract meaningful information and present it in novel ways. The work presented in this thesis will hopefully lay some of the foundations for this future work.

Bibliography

[1] H. Baker, D. Tanguay, I. Sobel, D. Gelb, M. Goss, B. Culbertson, and T. Malzbender. The coliseum immersive teleconferencing system. In *Proceedings of the International Workshop on Immersive Telepresence*, 2002.

[2] H. H. Baker and T. O. Binford. Depth from edge and intensity based stereo. In *International Joint Conference on Artificial Intelligence*, pages 631–636, 1981.

[3] H. Harlyn Baker, Donald Tanguay, Irwin Sobel, Dan Gelb, Michael E. Goss, W. Bruce Culbertson, and Thomas Malzbender. The coliseum immersive teleconferencing system. Technical Report HPL-2002-351, Hewlett Packard Labs, 2002.

[4] J. Batlle, E. Mouaddib, and J. Salvi. A survey: Recent progress in coded structured light as a technique to solve the correspondence problem. *Pattern Recognition*, 31(7):963–982, July 1998.

[5] James F. Blinn and Martin E. Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19(10):542–547, 1976.

[6] B. Boufama and K. Jin. Towards a fast and reliable dense matching algorithm. In *Proceedings of the Conference on Visio Interface*, page 178, 2002.

[7] J-Y Bouguet. Camera calibration toolbox for matlab:

www.vision.caltech.edu/bouguetj/calib-doc. Technical report, MRL-INTEL, 2003.

[8] Edmond Boyer and Jean Sebastien Franco. A hybrid approach for computing visual hulls of complex objects. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, pages 695–701. IEEE Computer Society, 2003.

[9] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. In *Proceedings of the Internatinal Conference on Computer Vision*, pages 377–384, 1999.

[10] Matthew Brand, Kongbin Kang, and David B. Cooper. An algebraic solution to visual hull. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, pages 30–35, 2004.

[11] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured Lumigraph Rendering. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 425—432, 2001.

[12] N. D. F. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla. Automatic 3d object segmentation in multiple views using volumetric graph-cuts. In *Proceedings of the British Machine Vision Conference*, pages 530–539, September 2007.

[13] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, 22(3):569–577, 2003.

[14] Shenchang Eric Chen. Quicktime VR: an image-based approach to virtual environment navigation. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 29–38. ACM Press, 1995.

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

[16] Kong Man Cheung, Simon Baker, and Takeo Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, pages 77–84, June 2003.

[17] Kong Man Cheung, Simon Baker, and Takeo Kanade. Visual hull alignment and refinement across time: A 3d reconstruction algorithm combining shape-from-silhouette with stereo. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 375–382, June 2003.

[18] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990.

[19] A. Criminisi, J. Shotton, A. Blake, and P. Torr. Gaze manipulation for one-to-one teleconferencing. In *Proceedings of the International Conference on Computer Vision*, pages 191—198, 2003.

[20] Antonio Criminisi, Jamie Shotton, Andrew Blake, and Philip Torr. Gaze manipulation for one-to-one teleconferencing. In *Proceedings of the International Conference on Computer Vision*, pages 191–198, Nice, France, 2003.

[21] Geoff Cross, Andrew W. Fitzgibbon, and Andrew Zisserman. Parallax geometry of smooth surfaces in multiple views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 323–329, 1999.

[22] W. Bruce Culbertson, Thomas Malzbender, and Gregory G. Slabaugh. Generalized voxel coloring. In *Proceedings of the International Workshop on Vision Algorithms*, pages 100–115, London, UK, 2000. Springer-Verlag.

[23] K. Daniilidis, J. Mulligan, R. KcKendall, D. Schmid, G. Kamberova, and R. Bajcsy. Real-time 3-D Teleimmersion. *Kluwer*, pages 253—266, 2000.

[24] James Davis, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 359–366, June 2003.

[25] L. di Stefano, M. Marchionni, S. Mattoccia, and G. Neri. A fast area-based stereo matching algorithm. In *Proceedings of the Conference on Visio Interface*, page 146, 2002.

[26] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, Massachusetts, 1996.

[27] Andrew Fitzgibbon, Yonatan Wexler, and Andrew Zisserman. Image-based rendering using image-based priors. In *Proceedings of the International Conference on Computer Vision*, page 1176, Washington, DC, USA, 2003. IEEE Computer Society.

[28] Andrew W. Fitzgibbon, Geoff Cross, and Andrew Zisserman. Automatic 3d model construction for turn-table sequences. In R. Koch and L. VanGool, editors, *Proceedings of Workshop on Structure from Multiple Images in Large Scale Environments*, volume 1506 of *Lecture Notes in Computer Science*, pages 154–170. Springer Verlag, June 1998.

[29] Jean-Sbastien Franco, Marc Lapierre, and Edmond Boyer. Visual shapes of silhouette sets. In *Proceedings of the Conference on 3D Processing, Visualization and Transmission*, pages 397–404, June 2006.

[30] Jean-Sébastien Franco and Edmond Boyer. Exact polyhedral visual hulls. In *Proceedings of the British Machine Vision Conference*, pages 329–338, September 2003. Norwich, UK.

[31] O. Ghita, J. Mallon, and P. F. Whelan. Epipolar line extraction using feature matching. In A. C. Winstanley, editor, *Proceedings of the Irish Machine Vision and Image Processing Conference*, pages 87–95, NUI Maynooth, September 2001.

[32] Michael Goesele, Steven M. Seitz, and Brian Curless. Multi-view stereo revisited. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2402–2409, June 2006.

[33] B. Goldluecke and M. Magnor. Space-Time Isosurface Evolution for Temporally Coherent 3D Reconstruction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages S–E, Washington, D.C., USA, July 2004. IEEE Computer Society, IEEE Computer Society.

[34] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 43–54. ACM Press, 1996.

[35] O. Grau. A studio production system for dynamic 3d content. In *Proceedings of Visual Communications and Image Processing, Proceedings of SPIE*, pages 80–89, 2003.

[36] Oliver Grau, Adrian Hilton, Joe Kilner, Gregor Miller, Tim Sargeant, and Jonathan Starck. A free-viewpoint video system for visualisation of sport scenes. *International Broadcasting Conference*, 2006.

[37] Oliver Grau, Adrian Hilton, Joe Kilner, Gregor Miller, Tim Sargeant, and Jonathan Starck. A free-viewpoint video system for visualisation of sport scenes. *From IT to HD*, 2006.

[38] Oliver Grau, Adrian Hilton, Joe Kilner, Gregor Miller, Tim Sargeant, and Jonathan Starck. A free-viewpoint video system for visualisation of sport scenes. *SMPTE Motion Imaging*, 2007.

[39] E. Grosso and M. Tistarelli. Active/dynamic stereo vision. *Transactions on Pattern Analysis and Machine Intelligence*, 17(9):868–879, 1995.

[40] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[41] Paul S Heckbert. Survey of texture mapping. *IEEE Computer Graphics Applications*, 6(11):56–67, 1986.

[42] S. Ivekovic and E. Trucco. Articulated 3d modelling in a wide-baseline disparity space. In *Proceedings of the European Conference on Visual Media Production*, pages 1–10. IET, November 2007.

[43] T. Kanade and P. Rander. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE MultiMedia*, 4(2):34—47, 1997.

[44] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *Proceedings of the European Conference on Computer Vision*, pages 82–96, 2002.

[45] A. Koschan. What is new in computational stereo since 1989: A survey on current stereo papers. Technical Report 93-22, Department of Computer Science, Technical University of Berlin, August 1993.

[46] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. Technical Report TR692, University of Rochester, 1998.

[47] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

[48] Svetlana Lazebnik, Edmond Boyer, and Jean Ponce. On computing exact visual hulls of solids bounded by smooth surfaces. In *Proceedings of the*

Conference on Computer Vision and Pattern Recognition, volume 1, pages 156–161, Kauai, Hawaii, 2001.

[49] Svetlana Lazebnik, Yasutaka Furukawa, and Jean Ponce. Projective visual hulls. *International Journal of Computer Vision*, 74(2):137–165, August 2006.

[50] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 31–42. ACM Press, 1996.

[51] Ming Li, Marcus Magnor, and Hans-Peter Seidel. Hardware-accelerated visual hull reconstruction and rendering. In *Proceedings of Graphics Interface*, pages 65–71, June 2003.

[52] Ming Li, Marcus Magnor, and Hans-Peter Seidel. Improved hardware-accelerated visual hull rendering. In *Proceedings of the Conference on Vision, Modeling and Visualization*, pages 151–158, November 2003.

[53] Candocia F. M. and Adjouadi M. A similarity measure for stereo feature matching. *IEEE Transactions on Image Processing*, 6(10):1460–1464, 1997.

[54] D. Marr and T. Poggio. A theory of human stereo vision. In *Proceedings of the Royal Society of London*, volume B204, pages 301–328, 1979.

[55] Wojciech Matusik, Chris Buehler, and Leonard McMillan. Polyhedral visual hulls for real-time rendering. In *Proceedings of the Eurographics Workshop on Rendering Techniques*, pages 115–126. Springer-Verlag, 2001.

[56] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J. Gortler, and Leonard McMillan. Image-based visual hulls. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 369–374. ACM Press/Addison-Wesley Publishing Co., 2000.

[57] Leonard McMillan and Gary Bishop. Plenoptic modeling: an image-based rendering system. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 39–46. ACM Press, 1995.

[58] Gregor Miller and Adrian Hilton. Exact view-dependent visual hulls. In *Proceedings of the International Conference on Pattern Recognition*, pages 107–111. IEEE Computer Society, August 2006.

[59] Gregor Miller and Adrian Hilton. Safe hulls. In *Proceedings of the Conference on Visual Media Production*, pages 1–8. IET, November 2007.

[60] Gregor Miller, Adrian Hilton, and Jonathan Starck. Interactive free-viewpoint video. In *Proceedings of the Conference on Visual Media Production*, pages 52–61. IEE, November 2005.

[61] Gregor Miller, Jonathan Starck, and Adrian Hilton. Projective surface refinement for free-viewpoint video. In *Proceedings of the Conference on Visual Media Production*, pages 153–163. IET, November 2006.

[62] J. Mitchelson and A. Hilton. Wand-based calibration of multiple cameras. In *British Machine Vision Association Workshop on Multiple Views*, May 2002.

[63] Karsten Mühlmann, Dennis Maier, Jürgen Hesser, and Reinhard Männer. Calculating dense disparity maps from color stereo images, an efficient implementation. *International Journal Computer Vision*, 47(1-3):79–88, 2002.

[64] W. Niem. Robust and fast modelling of 3d natural objects from multiple views. In *SPIE Proceedings on Image and Video Processing II*, volume 2182, pages 388–397, February 1994.

[65] Luc Robert and Rachid Deriche. Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. In *Pro-*

ceedings of the European Conference on Computer Vision, volume 1, pages 439–451. Springer-Verlag, 1996.

[66] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, April-June 2002.

[67] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1067–1073, 1997.

[68] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2006.

[69] Sekhavat Sharghi and Farhad A. Kamangar. Geometric feature-based matching in stereo images. In Robin Evans, Lang White, Daniel McMichael, and Len Sciacca, editors, *Proceedings of Information Decision and Control 99*, pages 65–70, Adelaide, Australia, February 1999. Institute of Electrical and Electronic Engineers, Inc.

[70] Sudipta N. Sinha and Marc Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *ICCV*, pages 349–356, 2005.

[71] Greg Slabaugh, Bruce Culbertson, Tom Malzbender, and Ron Schafer. A survey of methods for volumetric scene reconstruction from photographs. In *Proceedings of the Joint IEEE TCVG and Eurographics Workshop*, pages 81–100. Springer Computer Science, 2001.

[72] Gregory G. Slabaugh, Ronald W. Shafer, and Mat C. Hans. Image-based photo hulls. Technical Report HPL-2002-28, Hewlett Packard Labs, 2002.

[73] Dan Snow, Paul Viola, and Ramin Zabih. Exact voxel occupancy with graph cuts. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 345–352, June 2000.

[74] J. Starck and A. Hilton. Model-based multiple view reconstruction of people. In *IEEE International Conference on Computer Vision*, pages 915–922, 2003.

[75] J. Starck and A. Hilton. Virtual view synthesis of people from multiple view video sequences. *Graphical Models*, 67(6):600–620, 2005.

[76] Jonathan Starck, Gregor Miller, and Adrian Hilton. Video-based character animation. In *Proceedings of the Symposium on Computer Animation*, pages 49–58. ACM, July 2005.

[77] Jonathan Starck, Gregor Miller, and Adrian Hilton. Volumetric stereo with silhouette and feature constraints. In *Proceedings of the British Machine Vision Conference*, volume 3, page 1189. British Machine Vision Association, September 2006.

[78] R. Szeliski. Real-time octree generation from rotating objects. Technical report, Cambridge Research Laboratory, HP Labs, 1990.

[79] R. Szeliski and R. Zabih. An experimental comparison of stereo algorithms. In *Vision Algorithms: Theory and Practice*, number 1883 in LNCS. Springer, sep 1999.

[80] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 251–258. ACM Press/Addison-Wesley Publishing Co., 1997.

[81] Arun P. Tirumalai, Brian G. Schunck, and Ramesh C. Jain. Dynamic stereo with self-calibration. *Transactions on Pattern Analysis and Machine Intelligence*, 14(12):1184–1189, 1992.

[82] Sundar Vedula, Simon Baker, and Takeo Kanade. Spatio-temporal view interpolation. In *Proceedings of the Eurographics Workshop on Rendering*, pages 65–76, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[83] G. Vogiatzis, P. H. S. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 391–398, Washington, DC, USA, 2005. IEEE Computer Society.

[84] V.Vezhnevets and A.Velizhev. Gml c++ camera calibration toolbox: http://research.graphicon.ru/calibration/gml-c++-camera-calibration-toolbox.html. Technical report, Moscow State University, 2005.

[85] Ingo Wald. The Utah 3d animation repository: http://www.sci.utah.edu/ wald/animrep/. Technical report, University of Utah, 2007.

[86] K.-Y. K. Wong, P. R. S. Mendonça, and R. Cipolla. Head model acquisition from silhouettes. In *Proceedings of the International Workshop on Visual Form*, pages 797–796, Capri, Italy, May 2001. Springer–Verlag.

[87] Daniel N. Wood, Daniel I. Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H. Salesin, and Werner Stuetzle. Surface light fields for 3d photography. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 287–296. ACM Press/Addison-Wesley Publishing Co., 2000.

[88] O. J. Woodford, I. Reid, P. Torr, and A. W. Fitzgibbon. On new view synthesis using multiview stereo. In *Proceedings of the British Machine Vision Conference*, September 2007.

[89] S. Wuermlin, E. Lamboray, and M. Gross. 3d video fragments: Dynamic point samples for real-time free-viewpoint video. *Computers and Graphics, Special Issue on Coding, Compression and Streaming Techniques for 3D and Multimedia Data*, 28(1):3–14, 2004.

[90] Danny B. Yang, Hector H. Gonzalez-Banos, and Leonidas J. Guibas. Counting people in crowds with a real-time network of simple image sensors. In *Proceedings of the International Conference on Computer Vision*, page 122, Washington, DC, USA, 2003. IEEE, IEEE Computer Society.

[91] Ruigang Yang, Greg Welch, and Gary Bishop. Real-time consensus-based scene reconstruction using commodity graphics hardware. In *Proceedings of the Pacific Conference on Computer Graphics and Applications*, page 225. IEEE Computer Society, 2002.

[92] Ioannis A. Ypsilos, Adrian Hilton, and Simon Rowe. Video-rate capture of dynamic face shape and appearance. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 117–122, Seoul, Korea, 2004. IEEE Computer Society.

[93] Li Zhang, Brian Curless, and Steven M. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 367–374, June 2003.

[94] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995.

[95] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 600—608, 2004.