INTERACTIVE MIXED REALITY MEDIA WITH REAL TIME 3D HUMAN CAPTURE

TRAN CONG THIEN QUI

(B.Eng.(Hons.), Ho Chi Minh University of Technology)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2005

Abstract

A real time system for capturing humans in 3D and placing them into a mixed reality environment is presented in this thesis. The subject is captured by nine firewire cameras surrounding her. Looking through a head-mounted-display with a camera in front pointing at a marker, the user can see the 3D image of this subject overlaid onto a mixed reality scene. The 3D images of the subject viewed from this viewpoint are constructed using a robust and fast shape-from-silhouette algorithm. The thesis also presents several techniques to produce good quality and speed up the whole system. The frame rate of this system is around 25 fps using only standard Intel processor based personal computers.

Beside a remote live 3D conferencing system, this thesis also describes an application of the system in art and entertainment, named Magic Land, which is a mixed reality environment where captured avatars of human and 3D virtual characters can form an interactive story and play with each other. This system also demonstrates many technologies in human computer interaction: mixed reality, tangible interaction, and 3D communication. The result of the user study not only emphasizes the benefits, but also addresses some issues of these technologies.

Acknowledgement

I would like to express my heartfelt thanks to the following people for their invaluable guidance and assistance during the course of my work.

- Dr. Adrian David Cheok
- Mr Ta Huynh Duy Nguyen
- Mr Lee Shangping
- Mr Teo Sze Lee
- Mr Teo Hui Siang, Jason
- Ms Xu Ke
- Ms Liu Wei
- Mr Asitha Mallawaarachchi
- Mr Le Nam Thang
- All others from Mixed Reality Laboratory (Singapore) who have helped me in one way or another.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

In the past few years, researchers have heralded mixed reality as an exciting and useful technology for the future of computer human interaction, and it has generated interest in a number of areas including computer entertainment, art, architecture, medicine and communication. Mixed reality refers to the real-time insertion of computer-generated graphical content into a real scene (see [7], [8] for reviews). More recently, mixed reality systems have been defined rather broadly with many applications demanding tele-collaboration, spatial immersion and multi sensory experiences.

Inserting real collaborators into a computer generated scene involves specialized recording and novel view generation techniques. There have been a number of systems focusing on the individual aspects of these two broad categories, but there

is a gap in realizing a robust real time capturing and rendering system which at the same time provides a platform for mixed reality based tele-collaboration and provides multi-sensory, multi-user interaction with the digital world. The motivation for this thesis stems from here. 3D-Live technology is developed to capture and generate realistic novel 3D views of humans at interactive frame rates in real time to facilitate multi-user, spatially immersed collaboration in a mixed reality environment.

Besides, this thesis also presents an application, named "Magic Land", a tangible interaction system with fast recording and rendering 3D humans avatars in mixed reality scene, which brings to users new kind of human interaction and self reflection experiences. Although, the Magic Land system itself only supports the recording and playback feature (because of the ability to self reflection and interaction with ones own 3D avatar), the system can be quite simply extended for live capture and live viewing.

Up to now, the idea of capturing human beings for virtual reality has been studied and discussed in quite a few research articles. In [9], Markus et al. presented "blue-c", a system combining simultaneous acquisition of video streams with 3D projection technology in a CAVE-like environment, creating the impression of total immersion. Multiple live video streams acquired from many cameras are used to compute a 3D video representation of a user in real time. The resulting video inlays are integrated into a virtual environment. In spite of the impression of the total immersion provided, blue-c does not allow tangible ways to manipulate 3D

videos captured. There are few interactions described between these 3D human avatars and other virtual objects. Moreover, blue-c is currently a single user per portal [9], and thus does not allow social interactions in the same physical space. Magic Land, in contrast, supports multi-user experiences. Using a cup, one player can tangibly manipulate her own avatar to interact with other virtual objects or even with avatars of other players. Furthermore, in this mixed reality system, these interactions occurs as if they are in the real world physical environment.

Another capture system was also presented in [10]. In this paper, the authors demonstrate a complete system architecture allowing the real-time acquisition and full-body reconstruction of one or several actors, which can then be integrated in a virtual environment. Images captured from four cameras are processed to obtain a volumetric model of the moving actors, which can be used to interact with other objects in the virtual world. However, the resulting 3D models are generated without texture, leading to some limitations in applying their system. Moreover, their interaction model is quite simple, only based on active regions of the human avatars. We feel it is not as tangible and exciting as in Magic Land, where players can user their own hands to manipulate the 3D full color avatars.

1.2 Contributions

The major technical achievements and contributions of this thesis to the research field can be summarized as follows:

- This thesis proposes a complete and robust real time and live human 3D recording system, from capturing images, processing background subtraction, to rendering for novel view points. Originating from the older and previous system [11], the novel system is developed by integrating new techniques to improve speed and quality.

- This thesis contributes new algorithm methods to compute visibility and blend color for the previous image-based novel view generation algorithm. These contributions have significantly improved quality and performance of the system, and are very useful for mixed reality researchers.

- Beside the image-based algorithm, this thesis also presents a novel algorithm to generate a 3D model of human. Reusing many techniques developed for the image-based algorithm, the new model-based algorithm aims to achieve the balance between speed and quality in acquiring human 3D models. Though this is only the first step, it opens a new trend for further developments.

- The real application, Mixed Reality Magic Land, is the cross-section where art and technology meet. It not only combines latest advances in human-computer interaction and human-human communication: mixed reality, tangible interaction, and 3D-live technology; but also introduces to artists of any discipline intuitive approaches of dealing with mixed reality content. Moreover, future development of this system will open a new trend of mixed reality games, where players actively play a role in the game story.

1.3 Thesis Organization

The structure of this thesis is as follows:

Chapter Two provides an overview of background and related work in mixed reality, novel view generation and remote tele-collaboration. Different approaches to generate novel views will be discussed in details. Advantages and disadvantages of each approach will be also presented.

Chapter Three describes the design of 3D-Live system. The hardware, software structure of the system is presented here. The system setup, including camera adjustment and calibration, is also described. Some parts of the software structure such as image processing and network communication are discussed in details here while the novel view generation algorithm will be described in the next chapter.

Chapter Four starts by giving an overview on the previous image-based novel view generation algorithm. The problems and issues of this algorithm will be described and, after that, novel algorithm methods to address these issues and improve the speed and quality will be presented.

Chapter Five presents the novel model-based algorithm. First, the motivations for model-based approaches for novel view generation will be discussed. After that, the chapter will present design methodologies and implementation of the novel algorithm. Finally, results of this algorithm will be evaluated.

Chapter Six presents the detailed design and implementation of Magic Land system, a typical mixed reality application of 3D Live system in art and enter-

tainment. The hardware and software design of this system is presented. This chapter also discusses about some modern well known mixed reality games, and makes a detailed comparison of Magic Land with these games. Results of a user study conducted for Magic Land will be also presented.

Chapter Seven provides the general conclusion and sets out the directions for future work. This chapter also provides some of my experience through important conferences and exhibitions where my work has been presented.

1.4 List of Publications

Four papers based on this thesis work have been published or accepted for the following international journals and conferences:

- Tran Cong Thien Qui, Ta Huynh Duy Nguyen, Asitha Mallawaarachchi, Ke Xu, Wei Liu, Shang Ping Lee, ZhiYing Zhou, Sze Lee Teo, Hui Siang Teo, Le Nam Thang, Yu Li, Adrian David Cheok, Hirokazu Kato, **"Magic Land: Live 3d Human Capture Mixed Reality Interactive System"**, *In CHI'05 Extended Abstracts on Human Factors in Computing Systems (Portland, OR, USA, April 02 - 07, 2005).* ACM Press, New York, NY, 1142-1143.

- Ta Huynh Duy Nguyen, Tran Cong Thien Qui, Ke Xu, Adrian David Cheok, Sze Lee Teo, ZhiYing Zhou, Asitha Mallawaarachchi, Shang Ping Lee, Wei Liu, Hui Siang Teo, Le Nam Thang, Yu Li, Hirokazu Kato, **"Real Time 3D Human Capture System for Mixed-Reality Art and Entertainment"**, *IEEE Transaction On Visualization And Computer Graphics (TVCG), 11, 6 (Nov. - Dec. 2005), 706 - 721.*

- Tran Cong Thien Qui, Ta Huynh Duy Nguyen, Adrian David Cheok, Sze Lee Teo, Ke Xu, ZhiYing Zhou, Asitha Mallawaarachchi, Shang Ping Lee, Wei Liu, Hui Siang Teo, Le Nam Thang, Yu Li, Hirokazu Kato, **"Magic Land: Live 3D Human Capture Mixed Reality Interactive System"**, *International Workshop: Re-Thinking Technology in Museums: Towards a new understanding of visitors experiences in museums*, Ireland. June 2005.

- Adrian David Cheok, Ta Huynh Duy Nguyen, Tran Cong Thien Qui, Sze Lee Teo, Hui Siang Teo, **"Future Interactive Entertainment Systems Using Tangible Mixed Reality"**, *International Animation Festival*, China, 2005.

Chapter 2

Background and Related Work

Initial studies such as [12] superimposed two-dimensional textual information onto real world objects. However, it has now become common to insert three-dimensional dynamic graphical objects into the world (e.g. [13]). Billinghurst et al. [14] used the augmented reality interface to display small 2D video streams of collaborators into the world in a video-conferencing application. In the first version of 3D-Live [11], these techniques were extended by introducing a full three-dimensional live captured image of a collaborator into the visual scene for the first time. As the observer moves his head, the view of the collaborator changes appropriately. This results in the stable percept that the collaborator is three-dimensional and present in the space with the observer.

The first version of 3D-Live [11] presented an image-based algorithm for generating an arbitrary viewpoint of a collaborator at interactive speeds, which was sufficiently robust and fast for a tangible augmented reality setting. 3D-Live is a

complete system for live capture of 3D content and simultaneous presentation in mixed reality. The user sees the real world from his viewpoint, but modified so that the image of a remote collaborator is rendered onto the scene. Fifteen cameras surround the collaborator, and the resulting video streams are used to generate the virtual view from any camera angle. Users view a two-dimensional fiducial marker using a video-see-through augmented reality interface. The geometric relationship between the marker and head-mounted camera is calculated, and the equivalent view of the subject is computed and drawn onto the scene.

The various technologies used in 3D-Live span multiple disciplines and have involved independently. Background Subtraction is the image processing step performed on the set of reference images, 3D-Live rendering is the implementation of an image based novel view generation algorithm, which involves computer vision and computer graphics. The relationship between the 2D fiducial marker and the user's head mounted camera is extracted by a toolkit developed by our lab called "MXRToolKit" [15] and the distributed capture-and-render system is implemented using socket programming principles.

The novel view generation problem can be stated as follows: "Given a finite number of 2D, calibrated reference images of a real world (3D) object, generate the viewpoint of the object as seen from a specified virtual camera". Note that the output is also a 2D image corresponding to the projection of the 3D object into the image plane of the specified virtual camera. However the reconstruction algorithm needs to create some form of 3D representation from the given camera reference

images. Interestingly this representation need not be an explicit 3D model, although some approaches may choose to do so [6]. In this thesis, approaches need to generate a complete 3D model is called Model-Based approaches while the others are called Image-Based approaches. Following, both of these approaches will be discussed in detail.

2.1 Model-based Approaches

Generally, model-based approaches can be categorized into two following groups:

- **Stereo-Based approaches:** Use stereo techniques to compute correspondences across images and then recover 3D structure by triangulation and surface-fitting.

- **Volume-Intersection approaches:** Approximate the visual hull. For each image, a cone silhouette will be generated. All these cones are then inserted with each other to create the 3D model.

Stereo-based approaches are more traditional and have been known for a long time. However, these approaches are based on correspondence estimation, and thus are neither very robust nor suitable for real-time applications. On the other hand, Volume-Intersection approaches appeared later, but have attracted more and more attention of researchers around the world. There are lots of researches on this, and can be sub-divided into three different groups: Voxel-based representation,

Line-based representation and Polyhedral-based representation. All of these will be presented in the Volume-Intersection section.

2.1.1 Stereo-based approaches

With these approaches, the correspondence between each pairs of image must first be computed. Usually, either correlation methods or feature-based methods are used [1].

Correlation methods can be described as: (see Figure 2.1)

- Choose a k x k window surrounding a pixel, P, in the first image of each pair.
- Compare this window against windows centered at neighbouring positions in the second image.
- The window that maximizes the similarity criterion will decide displacement of P from the first image to the second image.

Feature-based methods restrict the search for correspondences to a sparse set of features. Instead of image windows, they use numerical and symbolic properties of features, available from feature descriptors; Instead of correlationlike measures, they use a measure of the distance between feature descriptors. Corresponding elements are given by the most similar feature pair, the one associated to the minimum distance.

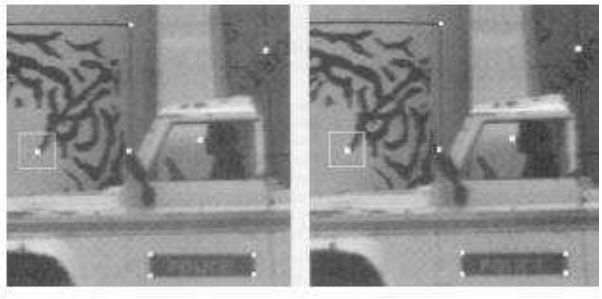


Figure 2.1: Correlation methods. Credit: E. Trucco and A. Verri [1]

.

After correspondences across images have been computed, the 3D structure will be recovered by triangulation. With any corresponding pair of points: $(P1, P2)$, the triangulation will generate two rays originating from the camera centers of each image and passing through $P1$ and $P2$. The intersection point of these two rays is the 3D point P. After finding sufficient 3D points, surface fitting techniques will then applied to produce the smooth surface connecting all these points.

Stereo-based approaches are especially effective with video sequences, where tracking techniques simplify the correspondence problem. Some representative papers on this area are [16] and [17].

Some of the disadvantages of Stereo-based approaches are: [18]

- Views must often be close together (i.e., small baseline) so that correspondence techniques are effective. Consequently, many cameras are required.
- Correspondences must be maintained over many views spanning large changes in viewpoint.

- Many partial models must often be computed with respect to a set of base viewpoints, and these surface patches must then be fused into a single, consistent model.

- If sparse features are used, a parameterized surface model must be fit to the 3D points to obtain the final dense surface reconstruction.

- There is no explicit handling of occlusion differences between views.

2.1.2 Volume Intersection approaches

The distinct feature of volume intersection approaches over stereo-based approaches is that: it does not need the point correspondence information in recovering the 3D object geometry, as required by stereo vision method.

Instead, these approaches try to approximate the visual hull of the captured objects. The visual hull of an object can be described as the maximal shape that gives the same silhouette as the actual object for all views outside the convex hull of the object. Volume intersection methods use a finite set of viewpoints to estimate the visual hull. Typically, one starts with a set of source images that are simply projections of the object onto N known image planes. Each of these N images must then be segmented into a binary image containing foreground regions to which the object projects; everything else is background. These foreground regions are then back-projected into 3D space and intersected, the resultant volume is the estimated visual hull of the object.

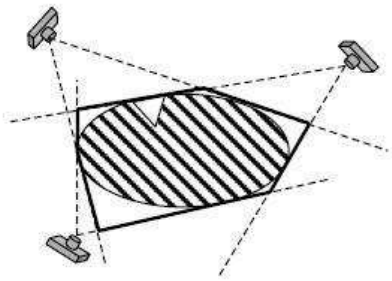


Figure 2.2: Visual hull reconstruction. Credit: G. Slabaugh et al. [2]

.

This estimate visual hull has following characteristics [2].

- It encloses the actual object.
- The size of the estimated visual hull decreases monotonically with the number of images used.
- Even when an infinite number of images are used, not all concavities can be modelled with a visual hull.

Regarding how to represent this volume, volume intersection approaches can be sub-divided into different approaches: voxel - based, line - based and polyhedral - based representations. The following details will describe research that has been done on each representation method.

2.1.2.1 Voxel - based representations

In this representation, the bounded area in which the objects of interest lie is divided into small cubes, called voxels (Volume Element). One important issue of

this representation is how big voxels are. If voxels are big, the resolution of the model is low and the model generated will miss some parts of the target object. This will lead to noticeable gaps in the result. In contrast, high resolution will result in a long computing process. To balance, usually the octree-representation is used. The octree space is modelled as a cubical region consisting of $2n$ x $2n$ x $2n$ unit cubes, where n is the resolution parameter [19]. Each unit cube has value 0 or 1, depending on whether it is outside or inside objects. The octree representation of the objects is obtained by recursively dividing the cubic space into octants. An octant is divided into eight if the unit cubes contained in the octant are not entirely 1's (opaque) or entirely 0's (transparent).

The result of the recursive subdivision process is represented by a tree of degree eight whose nodes are either leaves or have eight children. Thus, the tree is called an octree. Using the octree representation, the size of cubes (voxel) is not uniform. Voxels completely inside and completely outside are bigger while voxels at the boundary of the object are smaller. This octree representation is very highly efficient in terms of storage requirement and processing time.

Up to now, there has been a lot of work using this octree-representation to construct the 3D model. The main step in these algorithms is the intersection test [18]. Some methods back-project the silhouettes, creating an explicit set of cones that are then intersected either in 3D [20], [21], or in 2D after projecting voxels into the images [22]. Alternatively, it can be determined whether each voxel is in the intersection by projecting it into all of the images and testing whether it

is contained in every silhouette [23].

All the above methods use only information getting from the silhouettes. Using only this information, these algorithms can only generate the visual hull, which typically is not very correct [2]. Moreover, these visual hulls cannot include any concavities in captured 3D objects. To increase the geometry accuracy, more information than silhouettes must be used during reconstruction. Color is an obvious source of such additional information. Many researchers have attempted to reconstruct 3D scenes by analyzing colors across multiple viewpoints. Specifically, they try to generate a 3D model that, when projected on the reference views, can reproduces the original photographs (not only original silhouettes as visual hull). This color consistency can be used to distinguish surface points from other points in a scene. As shown in Figure 2.3, cameras with an unoccluded view of a non-surface point see surfaces beyond the point, and hence inconsistent (i.e., dissimilar) colors, in the direction of the point. On the left image, two cameras see consistent colors at a point on a surface, while on the right image, the cameras see inconsistent colors at a point not on the surface.

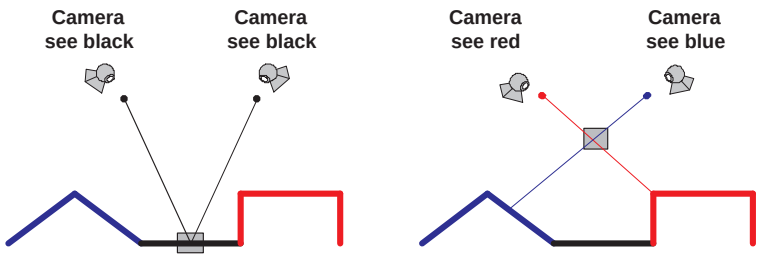


Figure 2.3: Color consistency. Credit: Slabaugh et al. [2].

The consistency of a set of colors can be defined as their standard deviation or, alternatively, the maximum of the L_1, L_2, or L_∞ norm between all pairs of the colors. Any of these measures can be computed for the colors of the set of pixels that can see a voxel; the voxel is considered to be on a surface if the measure is less than some threshold.

Real world scenes often include surfaces with abrupt color boundaries. Voxels that span such boundaries are likely to be visible from a set of pixels that are inconsistent in color. Hence, for such voxels, color consistency can fail as a surface test. This problem can be solved with an adaptive threshold that increases when voxels appear inconsistent from single images [2].

Seitz and Dyer [24] demonstrated that a sufficiently colorful scene could be reconstructed using full-color-based consistency alone, without volume intersection. They called their algorithm Voxel Coloring. The Voxel Coloring algorithm begins with a reconstruction volume of initially opaque voxels that encompasses the scene to be reconstructed. As the algorithm runs, opaque voxels are tested for color consistency and those that are found to be inconsistent are carved, i.e. made transparent. The algorithm stops when all the remaining opaque voxels are color consistent. When these final voxels are assigned the colors they project to in the input images, they form a model that closely resembles the scene.

Opaque voxels occlude each other from the input images in a complex and constantly changing pattern. To test the color consistency of a voxel, its visibility (the set of input image pixels that can see it) must first be determined. Since

this is done many times during a reconstruction, it must be performed efficiently. Calculating visibility is a subtle part of algorithms based on color consistency and several interesting variations have been developed.

To simplify the computation of voxel visibility and to allow a scene to be reconstructed in a single scan of the voxels, Seitz and Dyer imposed what they called the ordinal visibility constraint on the camera locations. It requires that the cameras be placed such that all the voxels are visited in a single scan in near-to-far order relative to every camera. Typically, this condition is met by placing all the cameras on one side of the scene and scanning voxels in planes that are successively further from the cameras. Thus, the transparency of all voxels that might occlude a given voxel is determined before the given voxel is checked for color consistency. This insures that the visibility of a voxel stops changing before it needs to be computed, which is important since every voxel is visited only once. An occlusion bit map, with one bit per input camera pixel, is used to account for occlusion. These bits are initially clear. When a voxel is found to be consistent, meaning it will remain opaque, all the occlusion bits in the voxel's projection are set, as shown in Figure 2.4. On the left image, a voxel is found to be consistent, and a bit in the occlusion bitmap is set for each pixel in the projection of a consistent voxel into each image. On the right, visibility of the lowest voxel is established by examining the pixels to which the voxel projects. These pixels are shown in black. If the occlusion bits have been set for these pixels, then the voxel is occluded, as is the case for the two middle cameras.

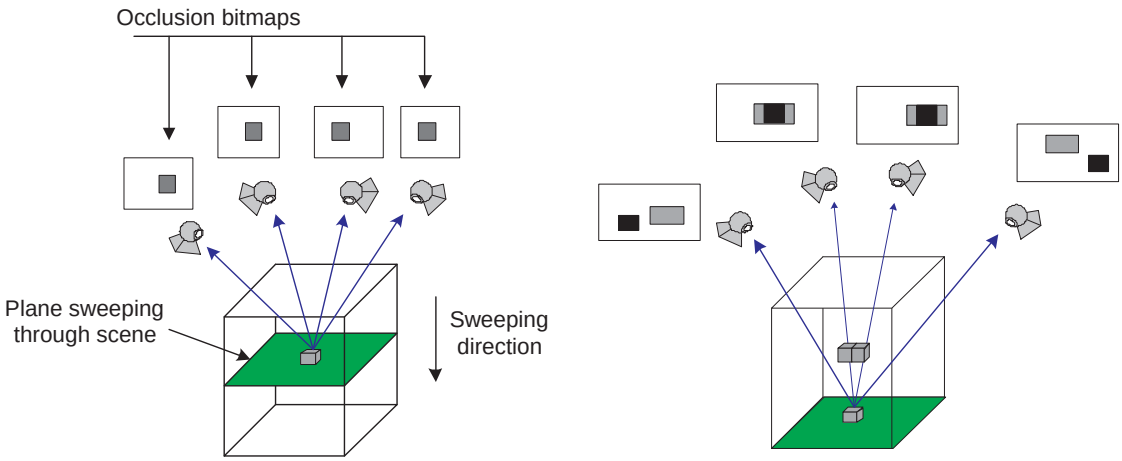


Figure 2.4: Using occlusion bitmaps. Credit: Slabaugh et al. [2].

This algorithm is quite effective. It can avoid backtracking - carving a voxel affects only voxels encountered later. However, the ordinal visibility constraint is a significant limitation. Since the voxels can be ordered from near to far relative to all the cameras, the cameras cannot surround the scene [2]. In such an arbitrary camera placement, a multiple-scan algorithm must be used. One of the algorithms for arbitrary camera placement is the Space Carving algorithm, implemented by Kutulakos and Seitz [3]. In their algorithm, the volume is scanned along the positive and negative directions of each of three axes. Space Carving forces the scans to be near-to-far, relative to the cameras, by using only images whose cameras have already been passed by the moving plane. Thus, when a voxel is evaluated, the transparency is already known of other voxels that might occlude it from the cameras currently being used.

Using this algorithm, the result is nearly perfect. One output is illustrated in figure 2.5. The left image is one of the 16 input images and the right image is

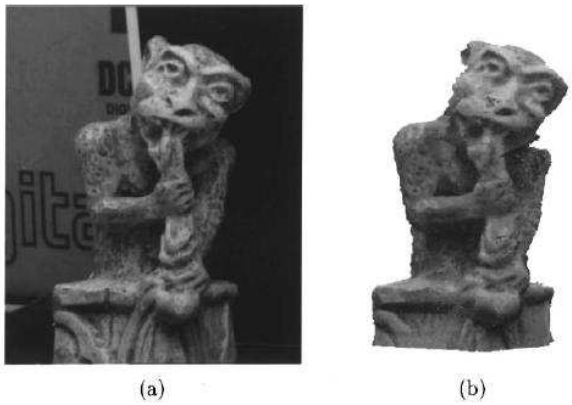


Figure 2.5: Output of Space-Carving Algorithm implemented by Kutulakos and Seitz [3].

the views of the reconstruction from the same viewpoints. As we can see, there are only a few errors. However, the processing time of this reconstruction is quite long. In their experiments, it took up to 250 minutes to generate the model of this gargoyle sculpture on an SGI 02 R1000/175 MHz workstation.

One of the efforts to increase the speed of the Space-Carving algorithm is due to Slabaugh et al. In their papers [4], they claimed that the performance of the Space-Carving algorithm depends heavily on two factors.

- **Visibility:** The method of determining of the pixels from which a voxel V is visible. We denote these pixels: Π^V.

- **Photo-consistency test:** A function that decides, based on Π^V, whether a surface exists at V.

Thus, to increase the performance, they introduced new ways to compute visibility and photo consistency. For the visibility, they proposed a new scene re-

Table 2.1: Runtime statistics for the toycar and Ghirardelli data sets

Data Set	**Algorithm**	**Time (m:s)**	**Memory**
Toycar	Space-Carving	32:31	156MB
Toycar	GVC-IB	36:16	74MB
Toycar	GVC-LDI	29:16	399MB
Ghir.	Space-Carving	2:35:43	337MB
Ghir.	GVC-IB	2:01:27	154MB
Ghir.	GVC-LDI	0:47:01	275MB

construction approach, Generalized Voxel Coloring (GVC), which introduces novel methods for computing visibility during reconstruction. This includes two sub-methods. The first GVC algorithm, GVC Item Buffer (GVC-IB), uses less memory than the other. It also uses incomplete visibility information during much of the reconstruction yet, in the end, computes the photo hull using full visibility. The other GVC algorithm, GVC Layer Depth Image (GVC-LDI), uses full visibility at all times, which greatly reduces the number of photo-consistency checks required to produce the photo hull, in other words, reduces the processing time. Table 2.1 presents runtime statistics of their experiments. As we can see, GVC-IB effectively reduces memory used while GVC-LDI significantly reduces processing time.

Regarding the consistency tests, they have proposed many approaches. Figure 2.6 presents the reconstructions of the shoes data set using different consistency tests. (a) is a photograph of the scene that was not used during reconstruction. (b)

was reconstructed using the likelihood ratio test, (c) using the bounding box test, (d) using standard deviation, (e) using standard deviation and the CIELab color space, (f) using the adaptive standard deviation test, and (g) using the histogram test.

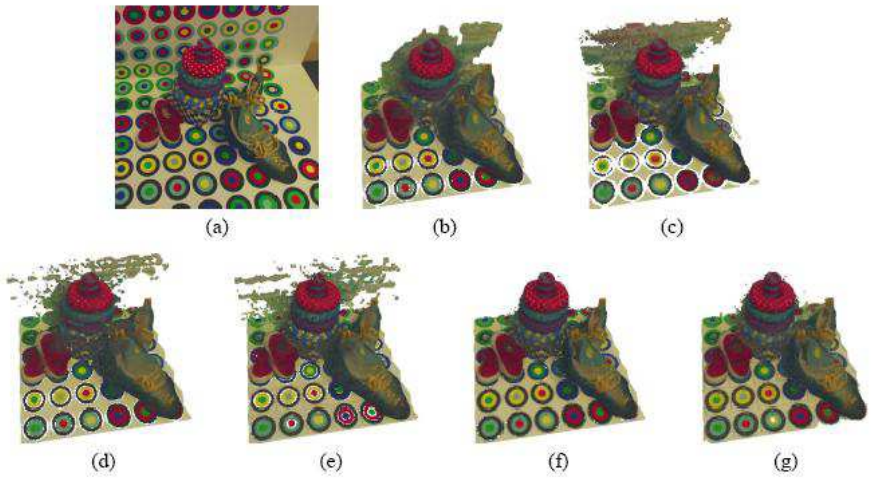


Figure 2.6: Results of different methods to test color consistency, implemented by Slabaugh et al. [4].

From the above output, we can see that the results of voxel-based methods are very good. However, the significant limitation of it is very long processing time. This makes it unsuitable for real-time application.

2.1.2.2 Line-segment based representation

With this representation, instead of using voxel model, researchers use a line-based geometry model. A line-based geometry model used to fit the 3D object is defined as a 2D array of line segments that have the same length and are perpendicular to

a base plane at the regular grid point [5], as shown in Figure 2.7.

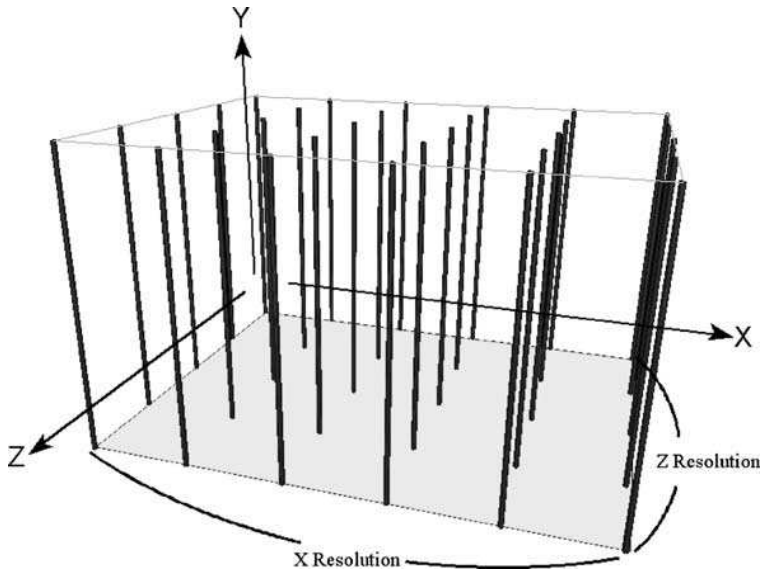


Figure 2.7: A line-based geometry. Credit: Y. H. Fang, H. L. Chou, and Z. Chen [5].

The uniform spacing between the grid points determines the spatial resolution of the line segments. In the reconstruction process, each 3D line of the model will be projected to each 2D image plane based on the camera calibration parameters. Then, for each projected line segment, we calculate the 2D line sections that intersect with object silhouette. After that, each of these found 2D line segments is back-projected to find the corresponding 3D line section on the chosen 3D line segment. Finally, all the 3D line sections obtained from all views are inserted. All these processes are illustrated in Figure 2.8.

The object line-based geometric model obtained above is a collection of line sections that is obviously not bounded. In order to finish the 3D shape reconstruction process, this model needs to be converted to a bounded triangular mesh model. To

do this, usually, the line-based model is first converted to the solid prism model, which, in turn, will be changed to the bounded triangular mesh model [5].

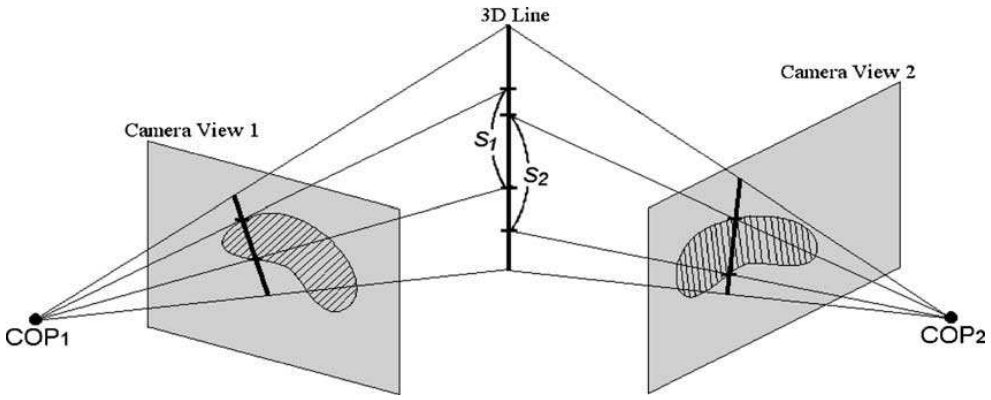


Figure 2.8: Reconstruction process of line-based models. Credit: Y. H. Fang, H. L. Chou, and Z. Chen [5].

Two representative researches on this representation are due to Martin and Aggarwal [25] and Y.H. Fang et. al. [5]. Martin's research is the first research on this representation. Since this is the first, there are lots of limitations about performance and quality of the reconstructing process. After that, to improve this algorithm, Fang has developed a technique to dynamically adjust line resolution. Similar to voxel-resolution in the voxel-based representation, the line resolution in this line-based representation is also very important. If the line resolution is not high enough, it may miss some details of the object. Conversely, if the line resolution is high, the reconstruction will be quite long. To address this issue, Fang proposed using two-phase reconstruction process. In the first phase, the used line-based model has a fixed and low resolution. In the second phase, the algorithm will check any adjacent line segments for possible loss of the object details. If these

Table 2.2: Processing time (*seconds*) of Fang's system

	Low resolution	**Dynamic resolution**	**High resolution**
Teapot	(25 x 16,0) 0.4	(25 x 16,2) 2	(97 x 61,0) 6
Rifle	(100 x 8,0) 1	(100 x 8,2) 4	(100 x 8,2) 4
Flower	(30 x 30,0) 2	(30 x 30,2) 10	(30 x 30,2) 10

is such a possibility, a new line segment is inserted between the two original line segments (i.e. increase the line resolution locally) to capture the possible details of the object. The checking and inserting process is repeated until a user-specified maximum subdivision level is reached or until no new line insertion is needed. Using this technique, the speed of reconstructing is increased significantly. Table 2.2 listed the processing time of their experiments.

In Table 2.2, for abbreviation: these model parameters are presented by (MxN,R), indicating the 2D array dimension is M x N and the maximum subdivision level in the dynamic line resolution scheme is R. If $R = 0$, the phase-two is skipped and the model is reconstructed with a fixed line resolution. Figure 2.9 shows some results of Fang's system. (a) the teapot using the (25 x $16, 2$) setting, (b) the rifle using the (100 x $8, 2$) setting and (c) the flower using the (30 x 30,2) 4 setting.

The advantage of line-based approaches is the relatively short processing time. However, researches on this approach have not mentioned about how to texture the visual hull. Including this process can make these algorithms even slower.

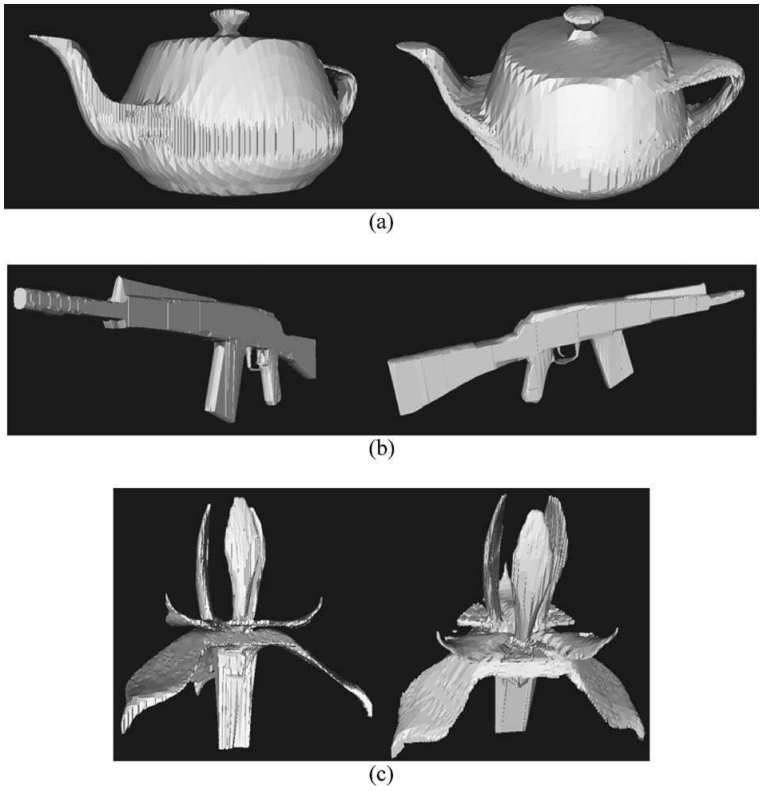


Figure 2.9: Some results of Fang's system [5].

2.1.2.3 Polyhedral-based representations

Unlike voxel-based and line-based approaches, which are solid model representation, polyhedral-based representation is a surface representation. This representation uses an exact polyhedral to represent for the surface of the visual hull. The important advantage of surface representations over solid model representations is that they are well-suited for rendering with graphics hardware, which is optimized for triangular mesh processing. Moreover, this representation can also be computed and rendered just as quickly as sampled representations, and thus it is useful for real-time applications [6].

As described above, for a volume intersection approach, the 3D models are generated by intersecting all the silhouette cones. Silhouette cones are defined as cones originating from the camera's center of projection and extending infinitely while passing through the silhouette's contour on the image plane. In this approach, the resulting visual hull, which is a polyhedron, is described by all of its faces. One important note they drew is that: the faces of this polyhedron can only lie on the faces of the original cones, and the faces of the original cones are defined by the projection matrices and the edges in the input silhouettes.

Using this note, their algorithm for computing the visual hull can be described: For each input silhouette Si and for each edge e in the input silhouette Si, they compute the face of the cone. Then they intersect this face with the cones of all other input silhouettes. The result of these intersections is a set of polygons that define the surface of the visual hull.

To reduce the processing time, 3D insertions of a face of a cone with other cones are reduced to simpler intersections in 2D. More detailed, to compute the intersection of a face f of a cone $cone(Si)$ with a cone $cone(Sj)$, we project f onto the image plane of silhouette Sj (see Figure 2.10). Then we compute the intersection of projected face f with silhouette Sj. Finally, we project back the resulting polygons onto the plane of face f.

Besides, in order to speed up the intersection of projected cone faces and silhouettes, they utilize the Edge-Bin data structure. The edge-bin structure spatially partitions a silhouette so that we can quickly compute the set of edges that a pro-

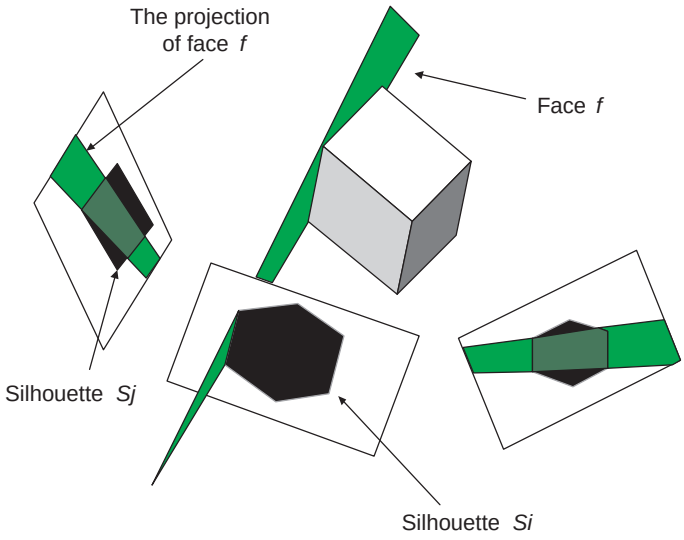


Figure 2.10: A single silhouette cone face is shown, defined by the edge in the center silhouette. Its projection in two other silhouettes is also shown. Credit: Matusik et al. [6].

jected cone face intersects. Using this data structure, instead of intersecting the entire projected cone face f with silhouette Sj, we just needs to intersect the two boundary lines of f with some edges of Sj which are selected based on the edge-bin data structure. After that, all intersection points are connected to each other to produce the intersection polygon.

Using all above techniques and algorithms, Matusik et al. have implemented a real-time rendering system. This system used four calibrated cameras to capture 3D objects. Each camera captured the video stream at 15 fps. A central computer (2x933 MHz Pentium III PC) will receive all these images and then generate the 3D model for each frame received. Their system can compute polyhedral visual hull models at a peak 15 frames per second. Although the speed of this system is

quite fast, the quality of the results as can be seen in Figure 2.11 is not very good. Further developments need to be done to improve the quality of rendering.

Figure 2.11: One output of Matusik's algorithm [6].

2.2 Image-based Approaches

In the previous section we explored various algorithms used to generate an explicit 3D model from a given set of calibrated reference images. However if the main goal of the system is to produce a given novel viewpoint, then an explicit model creation is not necessary. In "Image Based Visual Hulls" [26], an image-based visual hull texturing algorithm is described. This is of fundamental importance to the 3D-Live system, where the output is solely viewpoint dependant. As this algorithm is at the heart of 3D-Live rendering, it is described in detail in Chapter 4.

Chapter 3

3D-Live System Overview and Design

This chapter will describe the 3D-Live system in details. Firstly, we will look at the hardware components, their functions and connection diagram. After that, some system setup procedures such as camera adjustment and calibration will be described. Finally, we will look at the software components of the system.

3.1 Hardware and System Description

3.1.1 Hardware

Figure 3.1 represents the overall system structure. Eight Dragonfly FireWire cameras from Point Grey Research [27], operating at 30 fps, 640 x 480 resolution, are equally spaced around the subject, and one camera views him/her from above.

Three Sync Units from Point Grey Research are used to synchronize image acquisition of these cameras across multiple FireWire buses [27]. Three Capture Server machines, each one being DELL Precision Workstation 650 with Dual 2.8 GHz Xeon CPUs and 2 GB of memory, receive the three 640 x 480 video-streams in Bayer format at 30 Hz from three cameras each, and pre-process the video streams. The pre-processing stage will be described later in more detail.

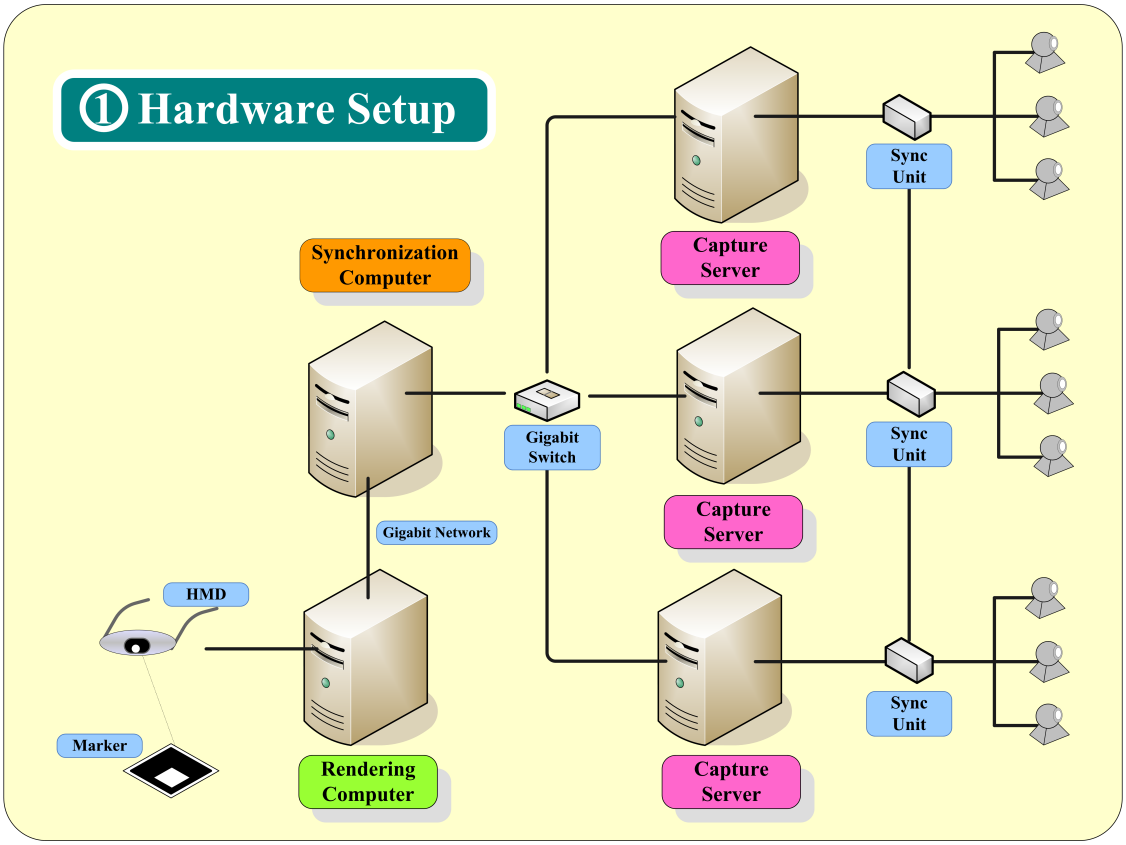


Figure 3.1: Hardware Architecture

The Synchronization machine is connected with three Capture Sever machines through a Gigabit network. This machine receives nine processed images from three Capture Server machines, synchronizes them, and sends them also via gi-

gabit Ethernet links to the Rendering machine, which is another DELL Precision Workstation 650.

The user views the scene through a video-see-through head mounted display (HMD) connected directly to the Rendering machine. A Unibrain firewire camera, capturing 30 images per second at a resolution of 640x480, is attached to the front of this HMD. The Rendering machine obtains images from this Unibrain camera, tracks the marker pattern on these images, calculates the position of the virtual viewpoint, generates a novel view of the captured subject from this viewpoint and then superimposes this generated view to the images obtained from the Unibrain camera and display it on the HMD. Details of each step will be discussed later in section 3.2.

3.1.2 System Setup

First of all, in order to generate the novel view of the subject from any angle/position of the virtual viewpoint, the zoom level, angle and position of each Dragonfly camera must be adjusted so that it can capture the whole subject even as he/she moves around. Moreover, to guarantee that the constructed visual hull is close enough to the object's shape, the zoom level and the position of each camera should be adjusted so that the camera looks at the subject at a far enough distance. The camera on top to view the subject from above is also to serve this purpose.

The system is very sensitive to the cameras' intrinsic and extrinsic parameters, because the visual hull construction algorithm bases on the relative distances

among cameras as well as the distances between the subject and the cameras. Consequently, after being adjusted, the position, zoom level, and angle of each camera have to be fixed, so that the camera's parameters are not changed anymore. The next step is to calibrate all the cameras to get the necessary parameters. Both the Unibrain camera attached to the HMD, and the Dragonfly cameras which capture the subject have to be calibrated. The intrinsic parameters of these cameras can be estimated using standard routines available with ARToolkit [28] or MXR-Toolkit [15].

For the Dragonfly cameras, we must not only estimate the intrinsic parameters, but also the extrinsic parameters to get the spatial transformation between each of the cameras. Calibration data is gathered by presenting a large checker-board to all of the cameras. For our calibration strategy to be successful, it is necessary to capture many views of the target in a sufficiently large number of different positions. Standard routines from Intel's OpenCV library [29] are used to detect all the corners on the checkerboard, in order to calculate both a set of intrinsic parameters for each camera and a set of extrinsic parameters relative to the checkerboard's coordinate system. Where two cameras detect the checkerboard in the same frame, the relative transformation between the two cameras can be calculated. By chaining these estimated transforms together across frames, the transform from any camera to any other camera can be derived [11], [30].

3.2 Software Components

3.2.1 Overview

All basic modules and the processing stages of system are represented in Figure 3.2. The Capturing and Image Processing modules are placed at each Capture Server machine. After Capturing module obtains raw images from the cameras, the Image Processing module will extract parts of the foreground objects from the background scene to obtain the silhouettes, compensate for the radial distortion component of the camera mode, and apply a simple compression technique.

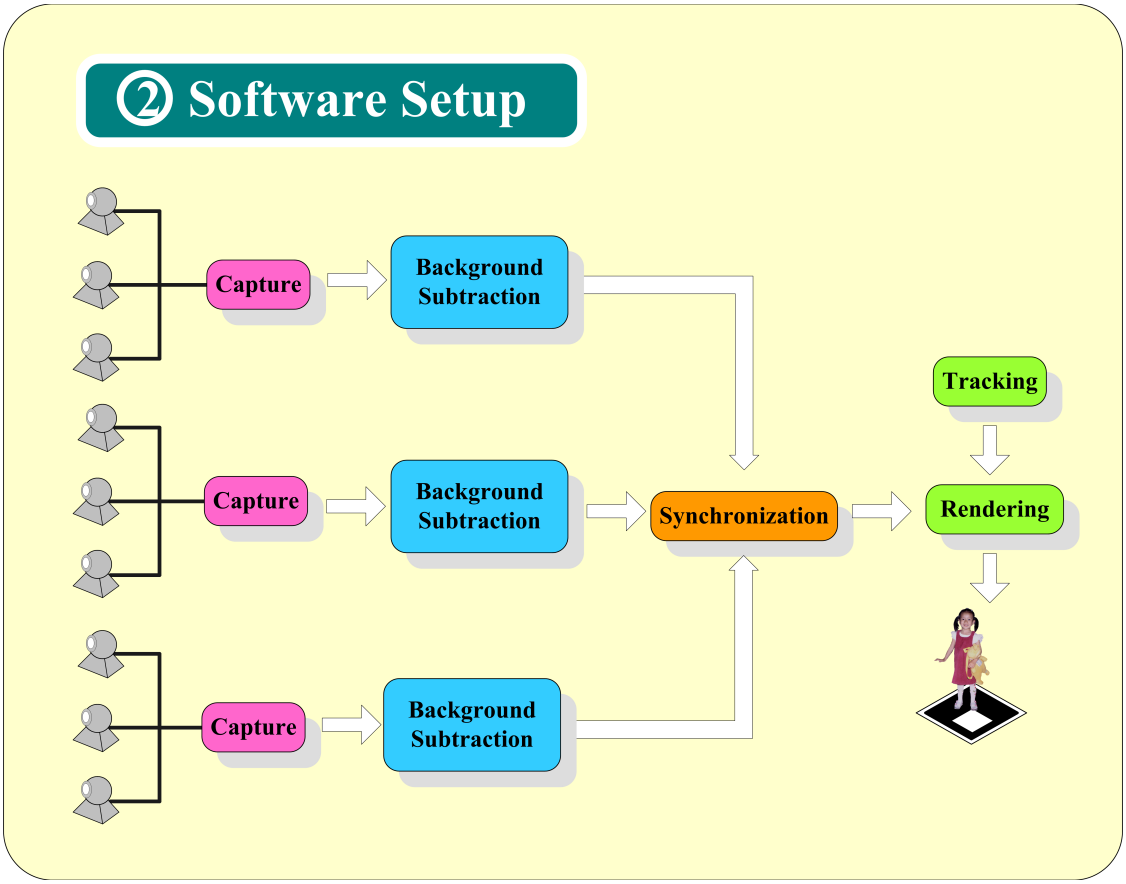


Figure 3.2: Software Architecture

The Synchronization module, on the Synchronization machine, is responsible for getting the processed images from all the cameras, and checking their timestamps to synchronize them. If those images are not synchronized, basing on the timestamps, Synchronization module will request the slowest camera to continuously capture and send back images until all these images from all nine cameras appear to be captured at nearly the same time.

The Tracking module will obtain the images from the Unibrain camera mounted on the HMD, track the marker pattern and calculate the Euclidian transformation matrix relating the marker co-ordinates to the camera co-ordinates. Details about this well-known marker based tracking technique can be found at [30], [28], or [15].

After receiving the images from the Synchronization module, and the transformation matrix from the Tracking module, the Rendering module will generate a novel view of the subject based on these inputs. The novel image is generated such that the virtual camera views the subject from exactly the same angle and position as the head-mounted camera views the marker. This simulated view of the remote collaborator is then superimposed on the original image and displayed to the user.

The subsequent parts will discuss more detail about the techniques used in each module.

3.2.2 Image Processing Module

The Image Processing module processes the raw captured image in three steps: background subtraction (which extracts parts of the foreground objects from the

image to obtain the silhouettes), radial distortion compensation, and image size reduction. The second step is done by applying the intrinsic parameters of the camera to estimate the correct position of each pixel. The remaining of this part will concentrate on the background subtraction and image size reduction steps.

3.2.2.1 Background subtraction

The result of visual hull construction in the Rendering module depends largely on the output of background subtraction step. This pre-processing step is one of the most crucial steps to determine quality of the final 3D model. Not only having to produce the correct foreground object, the chosen background subtraction algorithm must be very fast to fulfill the realtime requirement of this system. Another important requirement to guarantee the good shape of the visual hull is that the background subtraction algorithm must be able to eliminate the shadow caused by the objects.

There are many works on background subtraction, which produce rather good results, such as [31], [32], [33]. However, there normally exist the significant trade-off between processing time and quality of the result. The simple statistical method used in the previous work on 3D-Live [30] is very fast, but does not produce a good enough quality. To fulfill our needs, we use a modified method based on the scheme of Horpraset [31], which has the good capabilities of distinguishing the highlighted and shadow pixels. However, this algorithm has been modified in our research to reduce the computational intensiveness and optimize for the real time constraints

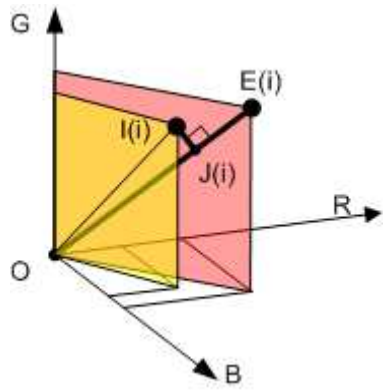


Figure 3.3: Color model

of this system.

The main idea of this method is to learn the statistics of properties of each background pixel over N pre-captured background frames, and obtain the statistical values modelling for the background. The pixel properties to be calculated here are the chromaticity and the brightness which is obtained from a new model of the pixel color. Basing on this, the algorithm can then classify each pixel into "foreground", "background", "highlighted background" or "shadow/shading background" after getting its new brightness and chromaticity color values. In our application, we only need to distinguish the "foreground" type from the rest.

The new color model which separates the brightness from the chromaticity component is summarized in Figure 3.3.

Regarding to Figure 3.3, in the RGB color space , the point $I(i)$ represents the color value of pixel i_th, and $E(i)$ represents the expected color value of this pixel, which coordinates $(\mu_R(i), \mu_G(i), \mu_B(i))$ are the mean values of the R, G, B components of this pixel obtained from the learning stage. $J(i)$ is the projection

of $I(i)$ on the line $OE(i)$.

The brightness distortion (α_i) and color distortion (CD_i) of this pixel are defined and calculated as:

$$\alpha_i = \frac{J(i)}{E(i)} = argmin_{\alpha_i} \left[\left(\frac{I_R(i) - \alpha_i \mu_R(i)}{\sigma_R(i)} \right)^2 + \left(\frac{I_G(i) - \alpha_i \mu_G(i)}{\sigma_G(i)} \right)^2 + \left(\frac{I_B(i) - \alpha_i \mu_B(i)}{\sigma_B(i)} \right)^2 \right] \tag{3.1}$$

$$CD_i = \sqrt{\left(\frac{I_R(i) - \alpha_i \mu_R(i)}{\sigma_R(i)} \right)^2 \left(\frac{I_G(i) - \alpha_i \mu_G(i)}{\sigma_G(i)} \right)^2 \left(\frac{I_B(i) - \alpha_i \mu_B(i)}{\sigma_B(i)} \right)^2} \tag{3.2}$$

In the above formula, $\sigma_R(i), \sigma_G(i), \sigma_B(i)$ are standard deviations of the i_th pixel's red, green, blue values computed in the learning stage. In our version, we assume that the standard deviations are the same for all pixels to make CD_i formula simpler:

$$CD_i = \left(I_R(i) - \alpha_i \mu_R(i)\right) \left(I_G(i) - \alpha_i \mu_G(i)\right) \left(I_B(i) - \alpha_i \mu_B(i)\right) \tag{3.3}$$

Another assumption is that the distributions of α_i and CD_i are the same for all pixel i. With this assumption, we do not need to normalize α_i and CD_i as was being done in the previous work of [31].

These modifications reduce the complexity of the formula and quite drastically increases the calculation speed from 33ms/frame to 13ms/frame, but produce more small misclassified pixels than the original algorithm. However, these small errors can be easily filtered in the next step.

3.2.2.2 Filtering

The filtering step is necessary to remove the small misclassified regions. There are many filtering methods to process the images after background subtraction. However, regarding the real-time constraint, we use the simple morphological operators open and close to filter out small misclassified regions.

3.2.2.3 Data size for real time network constraints

One very important factor is the amount of data to transfer over the network. In order to reach the fastest network speed, the size of data has to be as small as possible. In our system, we try to optimize the data size by using two main following methods:

- Reducing the image size by only storing the smallest rectangular region containing the foreground objects. An algorithm is implemented to find out the contour of the foreground and base on this result to calculate the smallest bounding box. This finding the contour algorithm also acts as another filtering method, which filters all small misclassified foreground regions which contour lengths are less than a predefined threshold. The size of this smallest rectangular region bounding the foreground objects depends on how close the camera look at the object, and how large the object is. As mentioned in Section 3.1.2 System setup, all cameras must be adjusted so that they view the object from a far enough distance to guarantee quality of the visual hull. Consequently, for each camera, the average size of this bounding box of the

foreground is normally less than 1/8 the size of the whole image, which is a significant reduction in the data size.

- Using Bayer format [34] with background in-formation encoded to store the images. Instead of using 3 bytes to encode 3 color components Red, Green, Blue for each pixel, we encode the whole image in Bayer format, which costs only 1 byte for each pixel. Moreover, for each pixel, the background information is encoded in the least significant bit of the byte at the position of this pixel in the Bayer image, value 1 for background pixel and 0 for foreground pixel. Obviously, this method of storing images leads to some color information lost. However, because the lost information is not much, the color quality of the output images is still good. Consequently, the lost information is trivial, compared with the benefit of reducing much data size, which is at least 3 times smaller than the RGB format with background information encoded.

3.2.2.4 Results

The quality of the image processing step is shown in the sample results of Figure 3.4. We can see that there are small errors after we subtract the background by our optimized algorithm. In the figure, the small green pixels inside the body is the foreground pixels misclassified as background ones, and the small black pixels outside the body is the background pixels misclassified as foreground ones. However, these errors are completely removed after the filtering step. The speed of this

Figure 3.4: Results of Background subtraction: before and after filtering

step is only around 15ms/frame. Compared with the non-simplified algorithm, which is 37ms/frame including the filtering step, the optimized algorithm is fast enough for this real time application.

3.2.3 Synchronization

The main function of Synchronization module is receiving and synchronizing images which have been processed by Image Processing module. The purpose of synchronization is to ensure that all images are captured at the same time.

Figure 3.5 describes the data transferred from Image Processing to Synchronization. It includes three parts. The first part is the image which is processed by Image Processing Module. Instead of sending the whole image, we only transmit the smallest rectangle area of the original image that contains the silhouette. This significantly reduces the amount of data to be transmitted. The second part is the pixel-weights for this image. These weights will be used for blending color in the

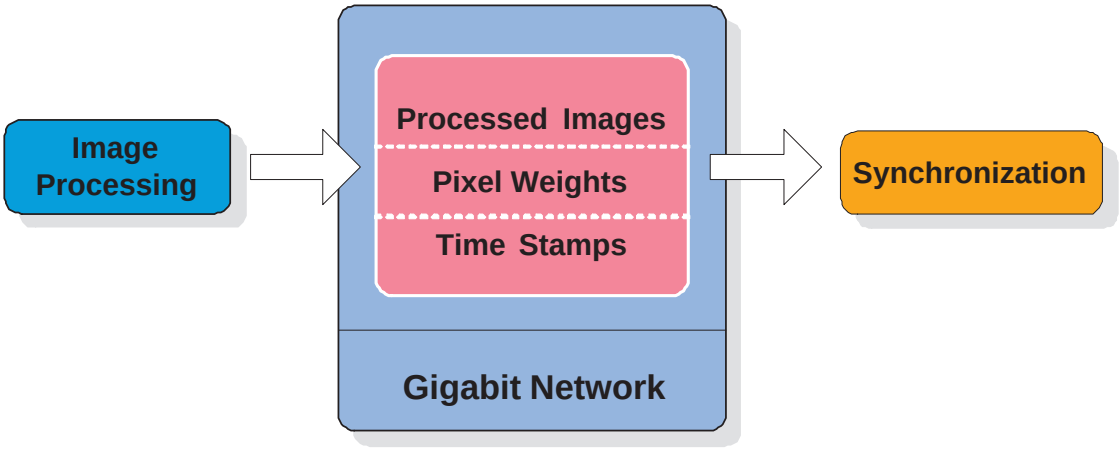


Figure 3.5: Data Transferred From Image Processing To Synchronization

rendering steps. I will present more about this weight in the Rendering section of this thesis. The last part to be transmitted is the Time Stamp, which is the time when this image is captured. Using this timing information, the Synchronization module will synchronize images captured from all nine cameras.

Once receiving one set of images from nine cameras, the time stamp of each image will be compared. If the difference in time between the fastest camera and the slowest camera is larger than 30 ms, the Synchronization Module will require Image Processing Module to provide a new image from the slowest camera. This synchronizing process will keep looping until the difference is smaller than 30ms. The reason to choose 30ms as the threshold is because our system operates at 30 fps.

3.2.4 Rendering

The rendering algorithm used in this system is a image-based novel view generation algorithm. This is one of the main focuses of this thesis and will be discussed in the next chapter.

Chapter 4

Image based Novel View Generation

The rendering algorithm used in this system is a new development over the previous algorithm which is described in [11]. To improve the speed and quality, this thesis introduces new ways to compute visibility and blend color in generating images for novel viewpoints. In this section, the main algorithm will be first briefly described. After that, improvements for speed and quality will be presented.

4.1 Overview of the 3D Human Rendering Algorithm

This rendering algorithm proceeds entirely on a per-pixel basis. In this thesis, the desired image is denoted as the "virtual camera image" and its constituent pixels

as "virtual pixels". The virtual camera can be determined by taking the product of the (head mounted) camera calibration matrix and the estimated transformation matrix. Given this 4 x 4 camera matrix, the center of each pixel of the virtual image is associated with a ray in space that starts at the camera center and extends outward. Any given distance along this ray corresponds to a point in 3D space. We calculate an image based depth representation by seeking the closest point along this ray that is inside the visual hull. This 3D point is then projected back into each of the real cameras to obtain samples of the color at that location. These samples are then combined to produce the final virtual pixel color. In summary, the algorithm must perform three operations for each virtual pixel:

- Determining the depth of the virtual pixel as seen by the virtual camera.
- Finding corresponding pixels in nearby real images.
- Determining pixel color based on all these measurements.

We briefly describe each of these operations in turn.

4.1.1 Determining Pixel Depth

The depth of each virtual pixel is determined by an explicit search starting at the virtual camera projection center and proceeding outward along the ray corresponding to the pixel center (see Figure 4.1). Each candidate 3D point along this ray is evaluated for potential occupancy. A candidate point is unoccupied if its projection into any of the silhouettes is marked as background. When a point is found

for which all of the silhouettes are marked as fore-ground, the point is considered occupied, and the search stops.

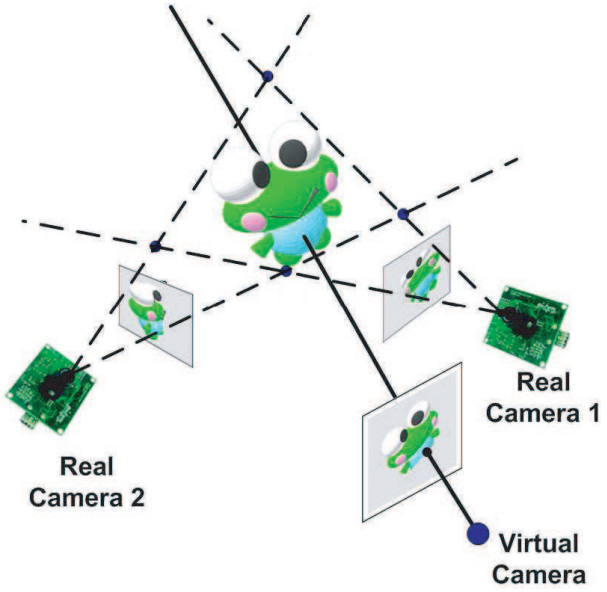


Figure 4.1: Novel View Point is generated by Visual Hull

Using this method, we can generate the visual hull very efficiently. One problem with visual hull is that the geometry it reconstructs is not very accurate. When photographed by only a few cameras, the scene's visual hull is much larger than the true scene [35]. One well-known improvement for visual hull which have been discussed in [3], [24], [2], [35], [4], and [36] is to utilize color constraint. Although, using this constraint, we can generate "photo-hull" which is a better approximation than visual-hull, the rendering speed will be decreased significantly and thus not suitable for real-time applications. Alternatively, we reduce the errors of visual hull by using more cameras and a larger recording room.

4.1.2 Finding Corresponding Pixels in Real Images

The resulting depth is an estimate of the closest point along the ray that is on the surface of the visual hull. However, since the visual hull may not accurately represent the shape of the object, this 3D point may actually lie outside of the object surface. Hence, care needs to be taken in choosing the cameras from which the pixel colors will be combined. Depth errors will cause incorrect pixels to be chosen from each of the real camera views.

To minimize the visual effect of these errors, it is better to choose incorrect pixels that are physically closest to the simulated pixel. So the optimal camera should be the one minimizing the angle between the rays corresponding to the real and virtual pixels. For a fixed depth error, this minimizes the distance between the chosen pixel and the correct pixel. We rank the cameras proximity once per image, based on the angle between the real and virtual camera axes.

We can now compute where the virtual pixel lies in each candidate cameras image. Unfortunately, the real camera does not necessarily see this point in space - another object may lie between the real camera and the point. If the real pixel is occluded in this way, it cannot contribute its color to the virtual pixel. In the previous versions of this research, we increase the system speed by intermediately accepting points that are geometrically certain not to be occluded. However, this geometrical information does not always provide true occlusion. As we can see in Figure 4.4, in the left image, we still can see false shadows of two hands over the body. These false hand shadows are generated because these parts of the body

are occluded from the reference cameras by the two hands, but the geometrical-based method cannot detect it. To achieve better results, in this new version, we introduce a new method to compute occlusion.

4.1.3 Determining Virtual Pixel Color

After determining the depth of a virtual pixel and which cameras have an unoccluded view, all that remains is to combine the colors of real pixels to produce a color for the virtual pixel. In the previous research, we took a weighted average of the pixels from the closest N cameras, such that the closest camera is given the most weight. This method can avoid producing sharp images that often contain visible borders where adjacent pixels were taken from different cameras. However, there are still some errors along the edge of the silhouette. In next section, we propose a new method to blend color which can overcome this problem.

4.2 New Algorithm Methods for Speed and Quality

4.2.1 Occlusion Problem

As said above, one of the main issues of this algorithm is the occlusion problem. In order to compute visibility, one basic approach is searching in 3D space. To determine if a point A is visible from one camera, we can simply search point by

point from A toward the center O of this camera. If any point in this ray belongs to the visual hull, A is considered to be invisible from this camera (Figure 4.2).

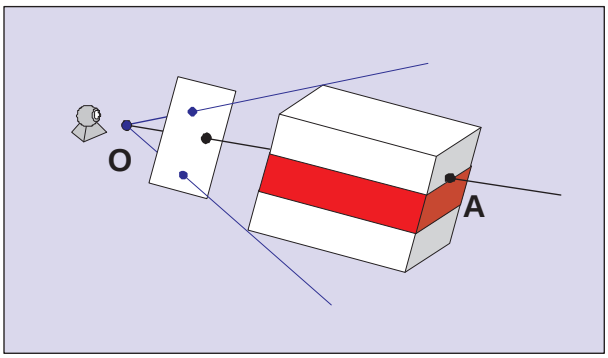


Figure 4.2: Example of Occlusion. In this figure, A is occluded from camera O.

Instead of brute-force searching in 3D space, [35] proposed a more efficient way which only need to step along epipolar lines. However, with this method, we still need to search on all captured images. To further increase the speed, we introduce a new method which only requires searching on one captured image.

To compute visibility, Matusik introduced a novel algorithm which can effectively reduce 3D visibility computation to the 2D visibility computation [6]. The main idea of this algorithm can be illustrated in Figure 4.3. In this figure, camera K is chosen so that the projection Q of P on this camera lies on the edge of silhouette. This algorithm bases on the fact that the 3D point P has to be visible from the camera K if on the image plane of one camera K, the 2D point Q is visible from the epipole E (the projection of the center of projection of camera K onto the image plane of camera J). In their paper, they use this algorithm to determine visibility of each face of the visual hull, but we apply it to compute visibility of

each point of the image-based visual hull. Our algorithm can be summarized as follows:

To determine if point P is visible from camera K, the three following steps will be processed:

1. Find one camera J where the project Q of P lies on the edge of the silhouette.

2. Find the epipole E of camera K on the image plane of camera J

3. If there is any foreground pixel lying on the line connecting point Q and point E, i.e. Q is occluded from point E, then P will be considered to be occluded from camera K. Otherwise, P will be consider to be visible from camera K.

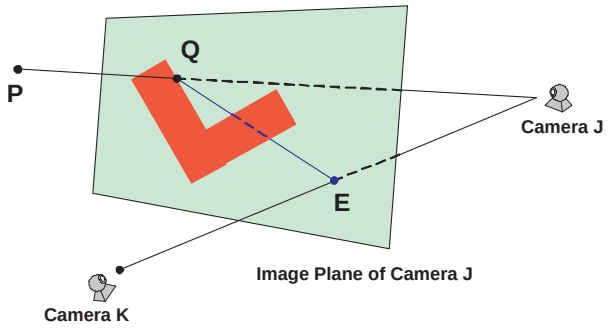


Figure 4.3: Visibility Computation: since the projection Q is occluded from the epipole E, 3D point P is considered to be invisible from camera K

Using this algorithm, we can avoid 3D searching while still able to detect occlusion whenever it happens. However, this algorithm is over-conservative [6]. It never considers a point visible if this point is occluded, but sometimes it considers a point occluded which is in fact visible. As a result, some points in visual hull will

be computed to be occluded from all cameras, which leads to holes in the results. To compensate for this, whenever a point is computed to be invisible from all cameras, we do not accept that but use the previous version to recompute visibility. The negative effect of this, for some points, we need to run both methods, but normally there are only few points like that. Thus, it does not affect the overall speed in any significant way.

Figure 4.4 shows example rendering results. In the left image, we use geometrical information to compute visibility while in the right, we use the above described visibility computing algorithm. As one can see, in the upper image, there are false shadows of two hands over the body while there is not in the lower image.

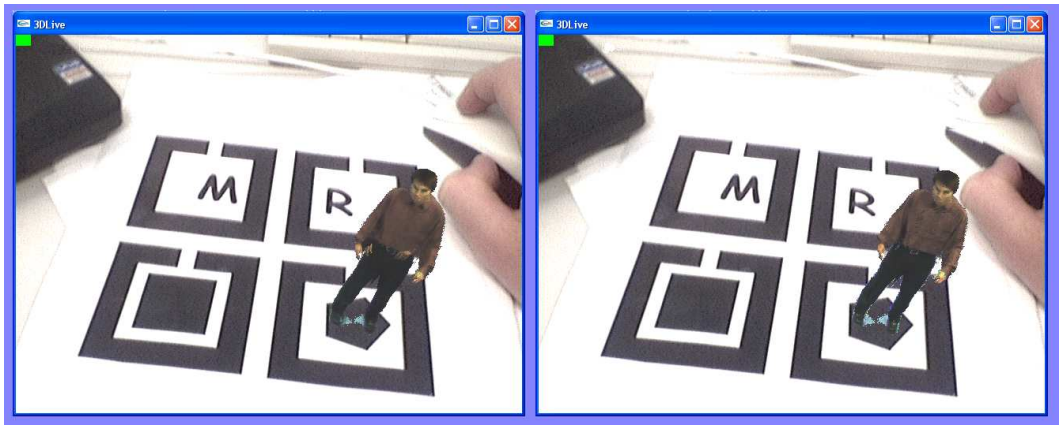


Figure 4.4: Rendering Results: In the left image, we use geometrical information to compute visibility while in the right, we use our new visibility computing algorithm. One can see the false hands appear in the upper image.

Table 4.1 shows the frame rate we can achieve with our algorithm. All three visibility algorithms: 3D searching, geometrical-based and our new algorithm are

Table 4.1: Rendering Speed

Image Size	**3D Searching**	**New Algorithm**	**Geometrical-Based**
320 x 240	7 fps	23 fps	27 fps
640 x 480	3 fps	11 fps	13 fps

tested. We also tested with two different resolutions: 320 x 240 and 640 x 480. As we can see, our new method is much faster than 3D searching method. With this new algorithm, we can achieve 23 fps at 320 x 240 and 11 fps at 640 x 480, while with 3D searching, it is only 7 fps and 3 fps respectively. Compared with the geometrical-based method, the new method is a little slower but it provides better results.

4.2.2 New method for blending color

The second improvement is a new method to blend color for visual hull. Most of current shape-from-silhouette algorithms use the angles between the desired view and reference views to decide the weights for blending. However, it can cause errors along the edges of foreground images, because background subtraction usually generates errors in these areas. For example, in Figure 4.5, if we base on the angles of cameras, point A will get color from camera 2 which is closer angle to the novel viewpoint. However, the projection of A to camera 2 is at the edge of the silhouette which usually contains some errors due to the background subtraction.

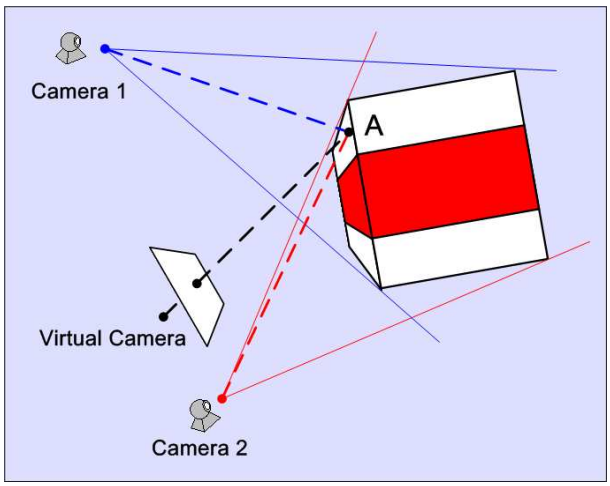


Figure 4.5: Example of Blending Color

To address this issue, we utilize a technique from image-mosaicking. In this subject of image-mosaiking, to reduce visible artifacts that is, to hide the edges of the component images, one can usually use a weighted average with pixels near the center of each image contributing more to the final composite [36]. Similar to this idea, in our algorithm, to determine the color of the virtual pixel, we take a weighted average with pixels near the center of each silhouette having higher weights. Thus, in Figure 4.5, if we use this blending method, A will get color from camera 1, where the projection of A is closer to the center of silhouette. This new blending method makes the visual hull smoother along the edges of silhouettes.

One problem with this blending method is that it requires more memory and time to store and calculate the weights, as each pixel of each reference images got different weights. To increase the speed, instead of computing these pixel weights during rendering, we calculate them during the image processing process. In such way, we can run this calculation on three different computers, each in charge of

images captured from three cameras. This will triple the speed. Thus, for each captured image, the Image Processing module will calculate the weights for each pixel and then pass these weights for the rendering module.

Figure 4.6 shows one set of images from nine cameras and their corresponding pixel weights. The brighter one pixel is, the higher weight it gets. Figure 4.7 shows two rendering results. The left is rendered with camera weights while the right with pixel weights. As we can see, using pixel weights, the result is better and smoother, especially along the edge of silhouettes.

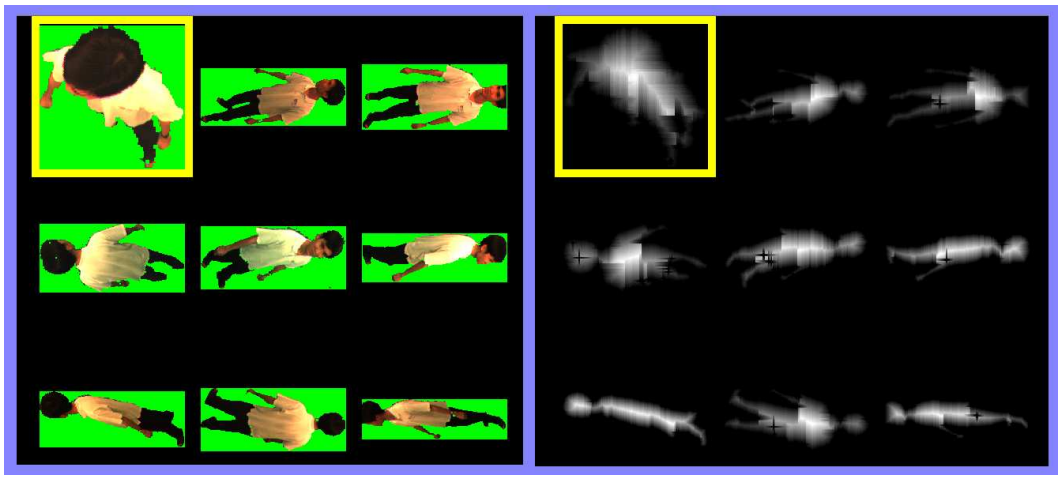


Figure 4.6: Original Images and Their Corresponding Pixel Weights

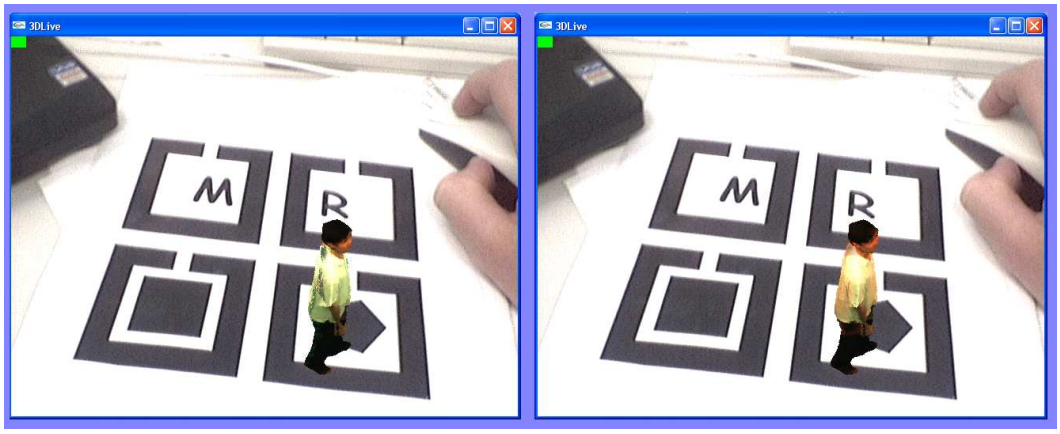


Figure 4.7: Rendering Results: The right is with the pixel weights algorithm while the left is not. The right image shows a much better result especially near the edges of the figure.

Chapter 5

Model Based Novel View Generation

This chapter gives the details of the work on a novel method of creating an explicit 3D model from the a finite set of calibrated reference images. As discussed in Chapter 4, image-based algorithms proceed entirely in image space, and there is no explicit model creation involved. This results in a fast implementation suited for real time applications. However in certain contexts, a model based approach would be more desirable.

5.1 Motivation

The main disadvantage of image-based approaches is that it does not generate the complete model. In many applications, model-based approaches are necessary. For

examples, in augmented reality applications, interactions between human-captured avatars and virtual objects can only be implemented when we have the whole 3D models of these avatars. Moreover, generating human 3D model is also very useful in many other areas, such as: medical applications, military applications, 3D movies, etc.

Model-based approaches also gain many advantages with multi-user applications where the human-captured objects are viewed by many users at the same time. When used for those applications, a image-based algorithm has to process the whole generating algorithm for each of the users even when two users have the same view points. Meanwhile, model-based algorithms only need to generate the whole 3D model once for all, and then project it to the view point of each user. The more users, the bigger advantage a model generation algorithm gets.

Another issue with the current 3D-Live system is the large amount of data transfer involved. With 9 cameras capturing 640x480 images at 30 fps for 20 seconds, it generates 1658 MB of data. Because the foreground is about 1/8 of the total image area, the data size to be transmitted is reduced to about 200 MB. This is still a large amount of data to be transmitted. If a 3D model can be extracted, many efficient representations have been proposed that would significantly reduce this file size. This is essential for 3D teleconferencing applications.

Another appeal is that if a polygonal model is created, generating novel viewpoints become much easier and faster because graphics hardware have been optimized for these functions. Rendering 3D models can be performed using a graphics

API like OpenGL or DirectX. OpenGL is the preferred library in the scientific community for its portability and mathematical consistency with standard literature in computer graphics. To render a 3D object with OpenGL one needs to define the objects in terms of a set of vertices defining a connected list of polygons.

5.2 Problem Formulation

In Chapter 4, I have already discussed about the way to determining the depth of the virtual pixel as seen by the virtual camera. In other words, given a novel viewpoint (a virtual camera), each pixel in the output image for this viewpoint can be associated with a 3D point in space. This will be used as the starting point for model construction as it provides a way of gathering 3D scene information.

Given this as input, our problem then becomes one of recovering a surface from a given set of 3D points. This turns out to be a key open-ended question in computer vision [37]. *Surface reconstruction from incomplete data sets is a classical problem in computer vision. The problem consists of finding a surface S that approximates a physical surface P by using a set of point coordinates sampled from the surface P. These point coordinates may be corrupted with noise, due to imperfections in the acquisition of the data. Like many other problems in computer vision, the problem of surface reconstruction is ill-posed. Prior knowledge about the world and the data acquisition process must therefore be used in order to make it solvable.* [38]

It was mentioned before that OpenGL needs a collection of vertices defining a

polygon mesh in order to render an object. The preferred method of representing and storing an arbitrary 3D object is through the use of a 3D model file (ex: a 3ds file), which stores not only a collection of 3D points (vertices), but also information of how a they should be connected to create a list of polygons. Additional information on the surface Normals, color and lighting information are also stored in a full fledged 3D model file but the Vertex and Polygon information are the most basic elements.

5.3 3D Model Generation Algorithm

The algorithm includes three following steps:

1. In order to accurately model the surface of the object, the depth points of the surface need to be sampled from several different viewpoints.

2. Recovering a surface from this given set of 3D points.

3. Combining several such surface representations into a single rigid object.

Following, each of these steps will be explained in detail.

5.3.1 Capturing a 3D Point Cloud

As mentioned above, the method described in Chapter 4 is used to compute the 3D location of one pixel at the surface of the visual hull. However, for each viewpoint,

only the visible portion of the visual hull surface would be revealed. Therefore, multiple viewpoints of the object has to be rendered to obtain the overall distribution of the surface depth points.

For reasons that will be explained in Section 5.4.3, one has to resort to judicious placement of the virtual camera, in order to get good results. In this section, it is taken as a given that location of the virtual camera is set to coincide with the real cameras.

After positioning the virtual camera in the required location, 3D points of the surface can be calculated. However, there is another problem; the surface 3D points are calculated in the virtual camera coordinates. The final aim is to obtain a collection of depth points that adequately describes the surface as a single rigid object, and for this all the points need to be expressed in a single reference frame. Therefore the 3D points computed from each virtual viewpoint needs to be transformed into a single fixed reference frame, either the world frame or one of the real camera's frame. The best choice is to save the points in world coordinate frame, as it makes the rendering easier.

5.3.2 Surface Construction

Now we already have an array of depth points representing a rigid object. To create a surface polygonal mesh out of this data, these points should be 'marked' specifying their connectivity. One assumption we can make is that on the local smoothness of the surface. Therefore one can device a scheme of linking up the

nearby depth points. If we have n depth points, in order to find the nearest neighbor for 1 point there are $n-1$ possibilities. making the whole operation potentially (upper bound) a nx$(n-1)$ 3D search problem. Therefore its desirable to look for additional constraints. Fortunately there is knowledge of the projective space locations of these 3D points. Therefore it is possible to make use of another assumption "nearby 3D points project to nearby image points", and reduce the problem into 2D. Even more encouragingly the 2D search can be eliminated altogether by the storing these vertices in a matrix form.

As shown in Figure 5.1, by using the assumption "nearby 3D points project to nearby image points, vertex 0 will be connected to vertices 1, 5, and 6, etc.

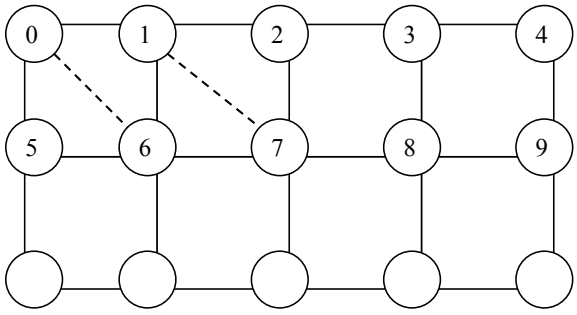


Figure 5.1: Construction of a Polygon List

The assumption of nearby 3D points project to nearby image points could be violated by objects with large depth discontinuities, or by self occlusion; as in the case when a human puts his hand in front of his body. In this case the original algorithm would create polygons linking depth points far away from each other. Currently there is a check on each triangle that limits the relative distance between

the 3 vertices. However the effectiveness has not been rigorously tested.

The main ideas of the model creation process is illustrated in Figure 5.2.

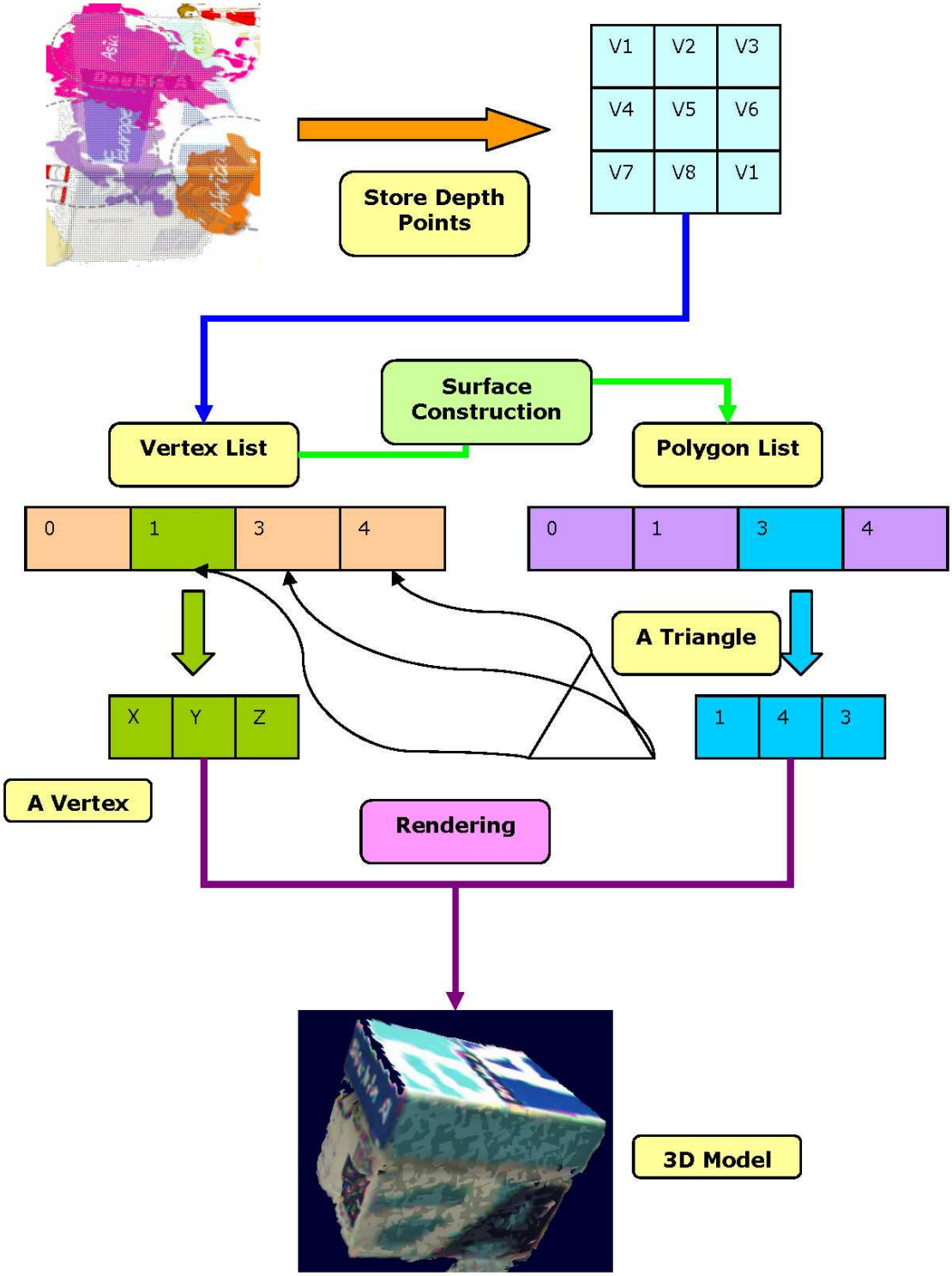


Figure 5.2: Illustration of the model creation process

5.3.3 Combining Several Surfaces with OpenGL

OpenGL and GLUT can be used to render the polygon list. In order to combine several surfaces, a pointer to a list of GLObjects are created. Each GLObject reads in one surface, and creates and holds its own polygon list. First, the centroid of the composite object is computed from the centroid of the individual surfaces. And in the above mentioned process of defining the vertices in the OpenGL display function, all the surfaces are now iterated and a collective surface representation is thus obtained. This is possible because the depth points of the different surfaces were converted into a common coordinate system. Finally the object is brought into the center of the world coordinate system by translation using the centroid. This makes the object visible in front of the screen.

5.4 Result and Discussion

This section presents the results of the model based novel view generation algorithm and provides a discussion on some of the issues in the implementation.

5.4.1 Capturing and Storing the Depth Points

Figure 5.3 shows some reference views obtained when we position the virtual camera at the real camera.

The images show good overall views of the object, however we can see that on some camera views there are some outlying pixels near the periphery of the object.

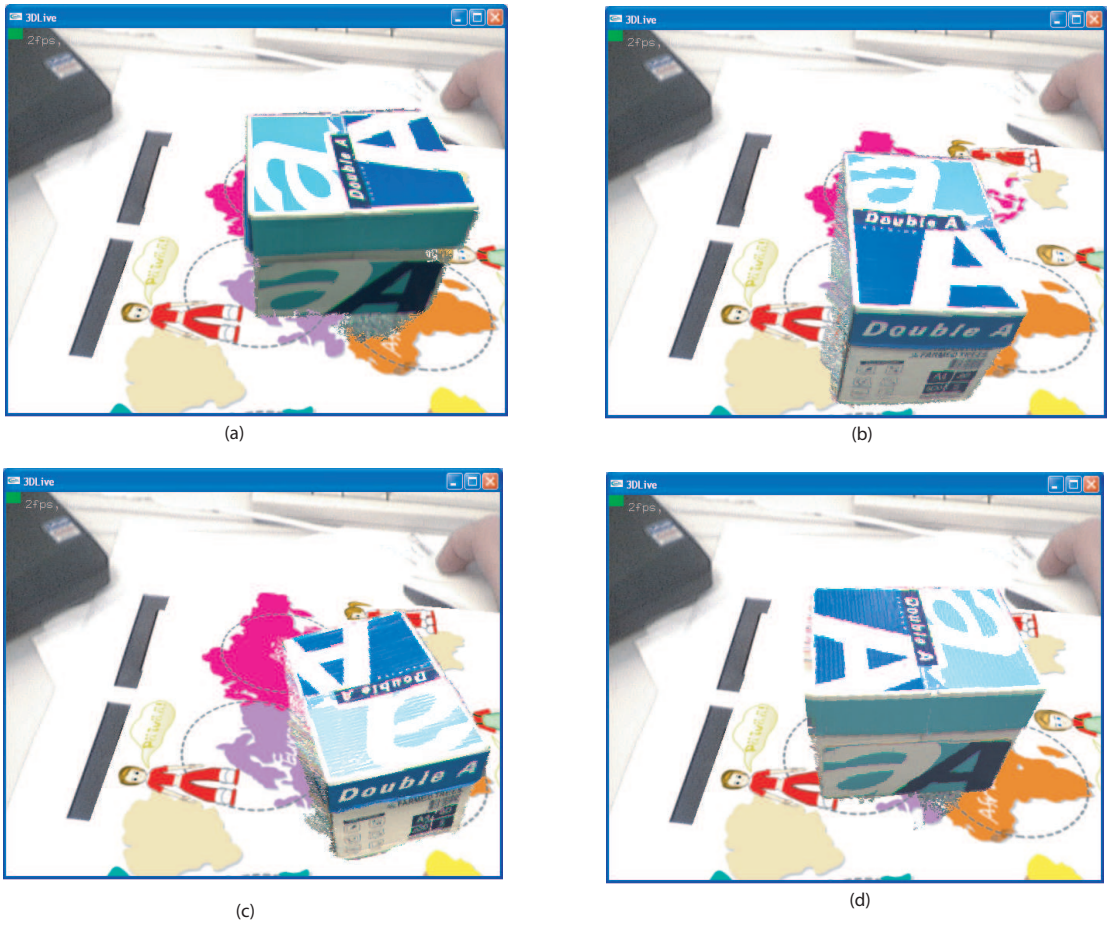
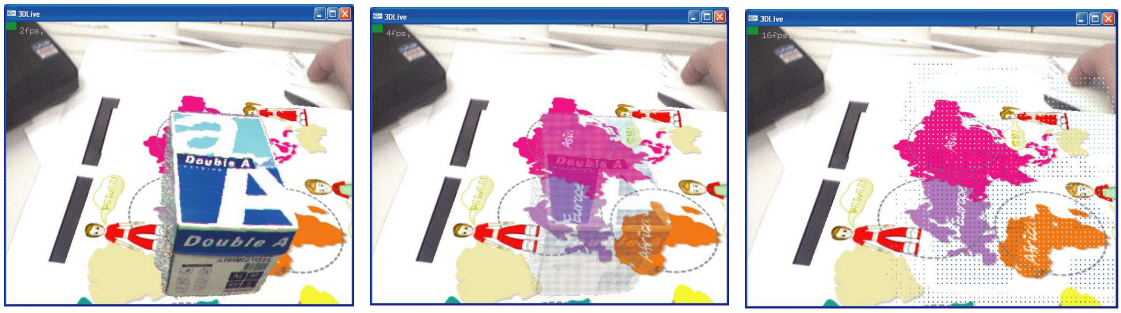


Figure 5.3: Four reference views generated by positioning the virtual camera at the real camera

This reveals that there are small errors in the process of positioning the virtual camera. This could be due some numerical inaccuracies of the original matrices propagating through the matrix inversions and multiplications.

The next issue concerns the sampling resolution of the storing of depth values. If the output resolution is very fine,it will generates a lot of samples, which results in an large vertex and polygon lists in the rendering phase. As the computation time for rendering scales up with the number of polygons, we have to reduce the sampling

rate to increase the rendering speed. Currently, the sampling rate was reduced to a $1/4^{th}$ of the pixel resolution (in both x and y directions). An illustration of reducing the sampling rate is shown in Figure 5.4. Sub-image (c) of Figure 5.4 reveals that at $1/4^{th}$ the pixel resolution the output image does not provide enough texture or shading information for the human eye to detect the shape of the object.



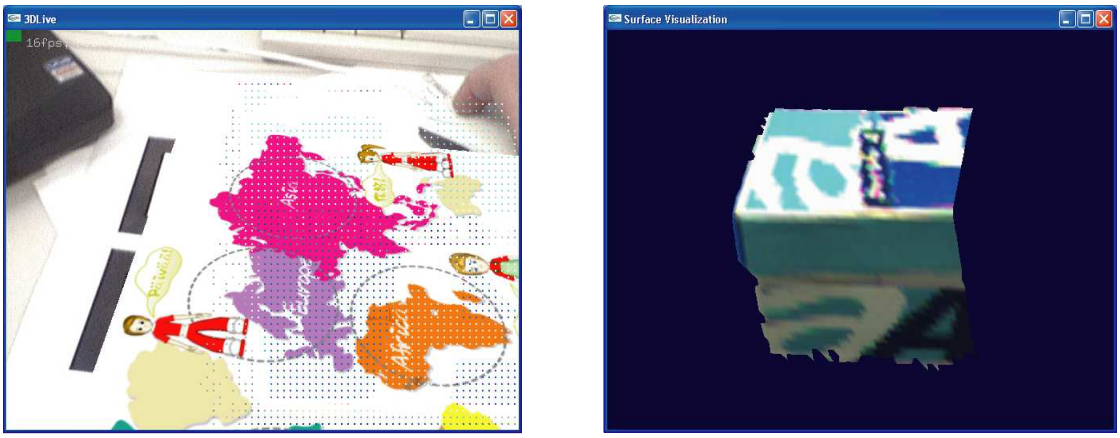
(a) Pixel Resolution (b) 1/2 Pixel Resolution (c) 1/4 Pixel Resolution

Figure 5.4: Reducing Sampling Rate

5.4.2 Creating the Polygon List and Rendering

Generating a surface from sampled depth points was successful. The algorithm used for this part are detailed in Section 5.3.2. Figure 5.5 illustrates the input and the output. The image (a) on the left shows the sampled input points and image (b) on the right shows the result of surface construction using OpenGL.

These results show that graphics hardware are optimized for rendering a polygon mesh. image (a) has very low spatial resolution, that is, the vertex data are only available at the 'dots' in sub image (a); And yet OpenGL is able to interpolate the colors for the other points on the polygon mesh.



(a) Sampled Points

(b) Generated Surface

Figure 5.5: Constructing a surface from sampled depth points

The rendering of the triangle list is better illustrated by the rendering of un-filled polygons as shown in Figure 5.6. This figure confirms the successful construction of a dense polygon mesh from the algorithms discussed in Section 5.5.

5.4.3 Composite Surfaces and Implications

The current implementation lacks the ability to filter out overlapping polygons. An exhaustive 3D search for overlaps would seriously hamper the chances of real time performance. The results of leaving OpenGL to handle overlaps causes some problems.

Figure 5.7 illustrates two individual surfaces (a) and (b) and the result of concurrently rendering both surfaces (c). It can be seen that when there are overlapping regions, and the surfaces are not smooth because only the surface which is nearer to the observer is displayed. Since the overall brightness between the reference images

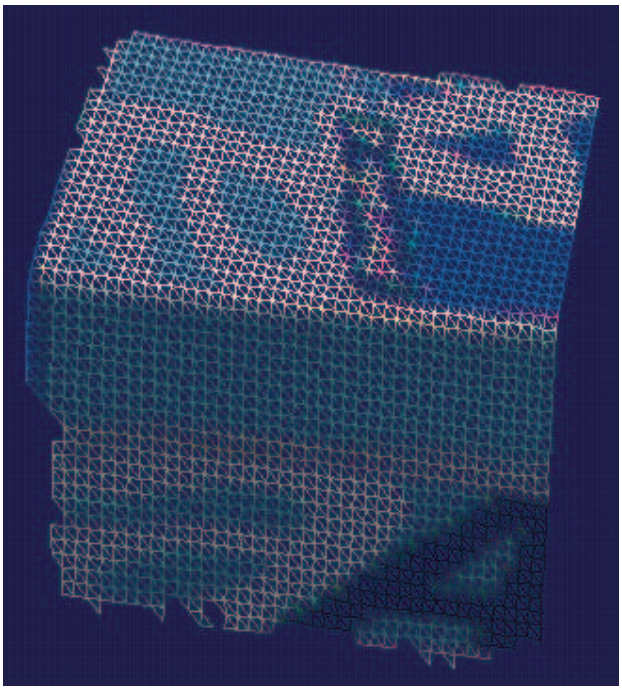
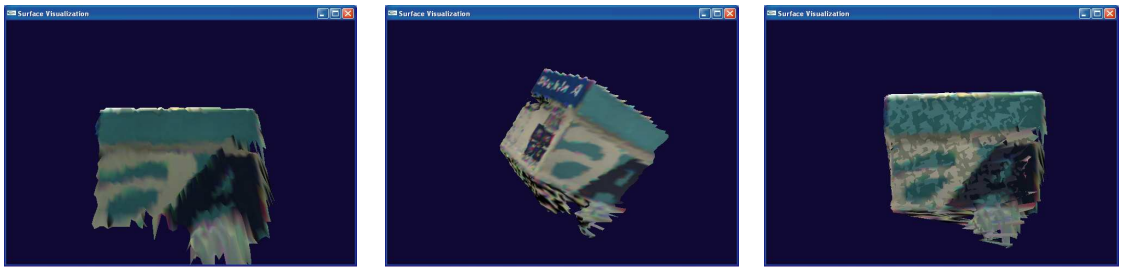


Figure 5.6: An un-filled polygon rendering of the object



(a) Surface 1 (b) Surface 2 (c) Combined Surface

Figure 5.7: Rendering of composite surfaces 1

is different, unwanted shading patterns appear in the final output. So far, the only way to get around this problem is by getting as far as possible, mutually exclusive virtual camera views. This is why judicious placement of the virtual camera is important to the success of this algorithm. Another result is shown in Figure 5.8.

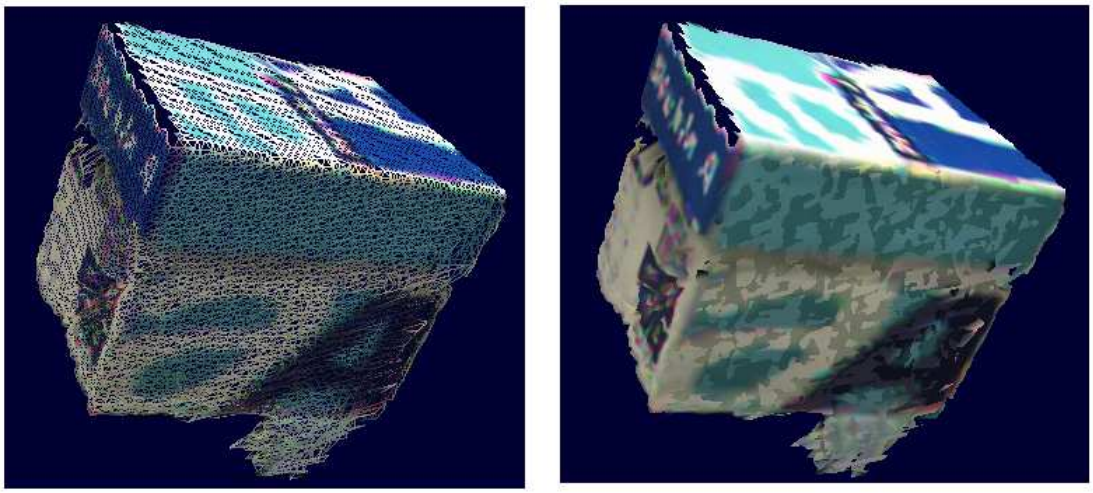


Figure 5.8: Rendering of composite surfaces 2

5.5 Conclusion

In this chapter, a model based generating algorithm is presented. It is developed more as a starting point for further developments than a complete algorithm. The current assumption "Nearby 3D points project to nearby image points" could be violated by objects with large depth discontinuities, or by self occlusion. One possible solution for this problem is checking the differences in depths of vertices. If two nearby vertices are too far from each other, they will not be connected. Beside this issue, further improvements will also be made in moderating brightness between the reference images in order to create smooth combined surfaces. These improvements will be approached in the future.

Chapter 6

Magic Land: an Application of the Live Mixed Reality 3D Capture System for Art and Entertainment

With the abilities of capturing, sending, regenerating the 3D images of live humans and objects in real time and displaying this objects' 3D images in the augmented reality environment, 3D-Live technology has many applications in various fields.

The first obvious application is a three-dimensional video-conferencing and collaboration system, which is much better than the traditional 2D video-conferencing system in term of communication benefits. It is because the 3D images displayed in real environment can fully represent non-verbal communication such as gestures,

which the traditional 2D system cannot. Moreover, using the 3D system, users not only can arrange markers representing several collaborators about them to create a virtual spatial conferencing space, but also can potentially conference from any location, and thus, the remote collaborators become part of any real world surroundings, potentially increasing the sense of social presence.

Another application of 3D-Live system in education and entertainment is an augmented book, in which a different fiducial marker is presented on each page, and associated with each is virtual content consisting both of 3D graphics and a narrator who was captured in our system. Others applications of this system in training, entertainment, computer games, etc. can be seen in [11].

The remaining of this part will fully describe a novel application of 3D-Live in art and entertainment. This system, named Magic Land, is the cross-section where art and technology meet. In technology viewpoint, it is a combination and demonstration of latest advances in human-computer interaction and human-human communication: mixed reality, tangible interaction, and 3D-Live technology. In artistic viewpoint, it aims to introduce tangible approaches of dealing with mixed reality content to artists of any discipline. These approaches, which allow artists to manipulate the mixed reality content intuitively and easily by using cups, was also presented in [39] for a city planning application.

Another main purpose of Magic Land system is to bring to all users a new special kind of human self reflection and human-human interaction. In this system, users can tangibly pick up themselves or their collaborators and watch them in 3D form

encountering with other virtual objects. In order to allow users to manipulate their own 3D recorded images in mixed reality environment, this version of Magic Land does not fully exploit the "live" capturing feature of 3D-Live, but instead utilizes the fast processing and rendering algorithms for fast 3D-Live record and playback features. However, another version of Magic Land, which can be built easily for live capture and live viewing, is discussed further in section 6.4. The artistic intention and motivation of the project will also be discussed further in section 6.3.

6.1 System Concept and Hardware Components

Magic Land is a mixed reality environment where 3D-Live captured avatars of human and 3D computer generated virtual animations play and interact with each other.

The system includes two main areas: recording room and interactive room. The recording room is where users can have themselves captured into live 3D models which will interact in the mixed reality scene. This room, which has nine Dragonfly cameras mounted inside, is a part of the 3D capture system described above. After the user gets captured inside the system, she can go to the interactive room to play with her own figure.

The interactive room consists of three main components: a Menu Table, a Main Interactive Table, and five playing cups. On top of these tables and cups are different marker patterns. A four cameras system (ceiling tracking system) is put

high above the Main Interactive Table to track the relative position of its markers with the markers of the cups currently put on it. The users view the virtual scenes and/or virtual characters which will be overlaid on these tables and cups via the video-see-through HMDs with the Unibrain cameras mounted in front and looking at the markers. The Main Interactive Table is first overlaid with a digitally created setting, an Asian garden in our case, whereas the cups serve as the containers for the virtual characters and also as tools for users to manipulate them tangibly. There is also a large screen on the wall reflecting the mixed reality view of the first user when he/she uses the HMD. If nobody uses this HMD for 15 seconds, the large screen will change to the virtual reality mode, showing the whole magic land viewed from a very far distant viewpoint.

An example of the tangible interaction on the Main Interactive Table is shown in the Figure 6.1. Here we can see a user using a cup to tangibly move a virtual panda object (left image) and using another cup to trigger the volcano by putting the character physically near the volcano (right image).

The Menu Table is where users can select the virtual characters they want to play with. There are two mechanical push buttons on the table corresponding with two types of characters: the human captured 3D-Live models on the right and VRML models on the left. Users can press the button to change the objects showed on the Menu Table, and move the empty cup close to this object to pick it up. To empty a cup (trash), users can move this cup close to the virtual bin placed at the middle of the Menu Table. In the Figure 6.2, in the left image, we can see

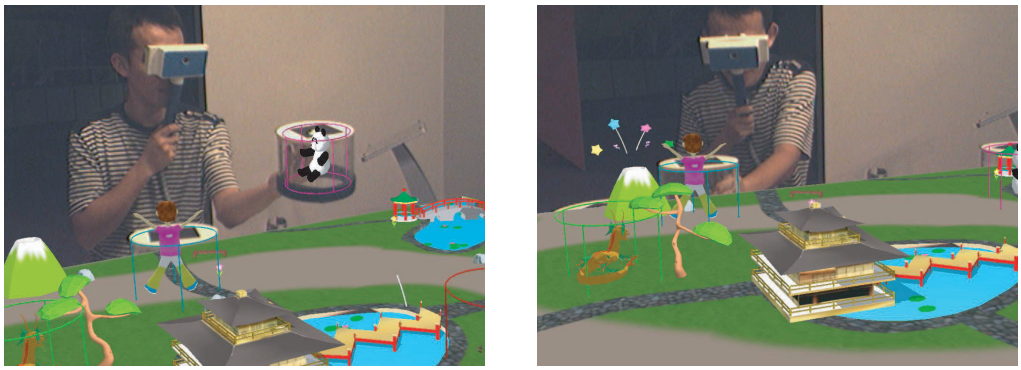


Figure 6.1: **Tangible interaction on the Main Table**: (Left) Tangibly picking up the virtual object from the table. (Right) The trigger of the volcano by placing a cup with virtual boy physically near to the volcano.

a user using a cup to pick up a virtual object, at the edge of the table closest to the user are two mechanical buttons. In the right image we can see the augmented view seen by this user. The user had selected a dragon previously which is inside the cup.

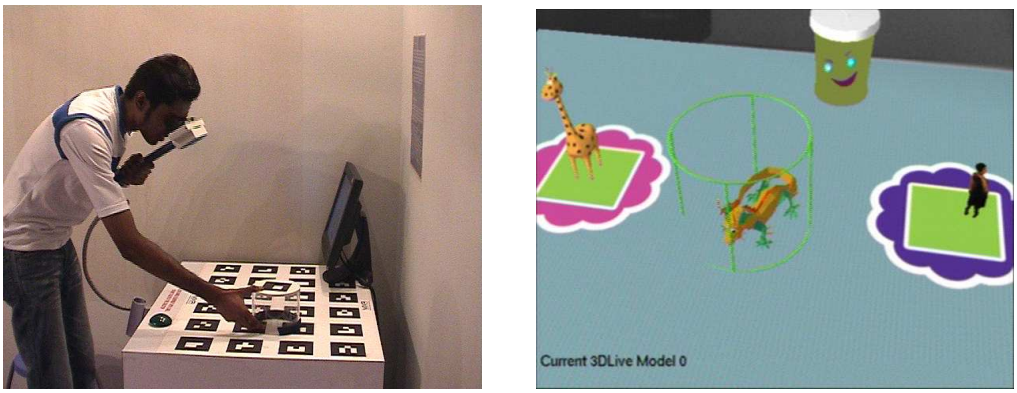


Figure 6.2: **Menu Table**: (Left) A user using a cup to pick up a virtual object. (Right) Augmented View seen by users

After picking up a character, users can bring the cup to the Main Interactive

Figure 6.3: **Main Table**: The Witch turns the 3D-Live human which comes close to it into a stone

Table to play with it. Consequently, there will be many 3D models moving and interacting in a virtual scene on the table, which forms a beautiful virtual world of those small characters. If two characters are close together, they would interact with each other in the pre-defined way. For example, if the dragon comes near to the 3D-Live captured real human, it will blow fire on the human. This gives an exciting feeling of the tangible merging of real humans with the virtual world. As an example of the interaction, in the Figure 6.3, we can see the interaction where the witch which is tangibly moved with the cup turns the 3D-Live human character which comes physically close to it into a stone.

6.2 Software Components

As shown in Figure 6.4., the software system of Magic Land consists of five main parts: 3D Live Recording, 3D Live Rendering, Main Rendering, Ceiling Camera Tracking, and Game Server. Beside these parts, there is a Sound module that

produces audio effects including background music and interactive sounds for the whole system.

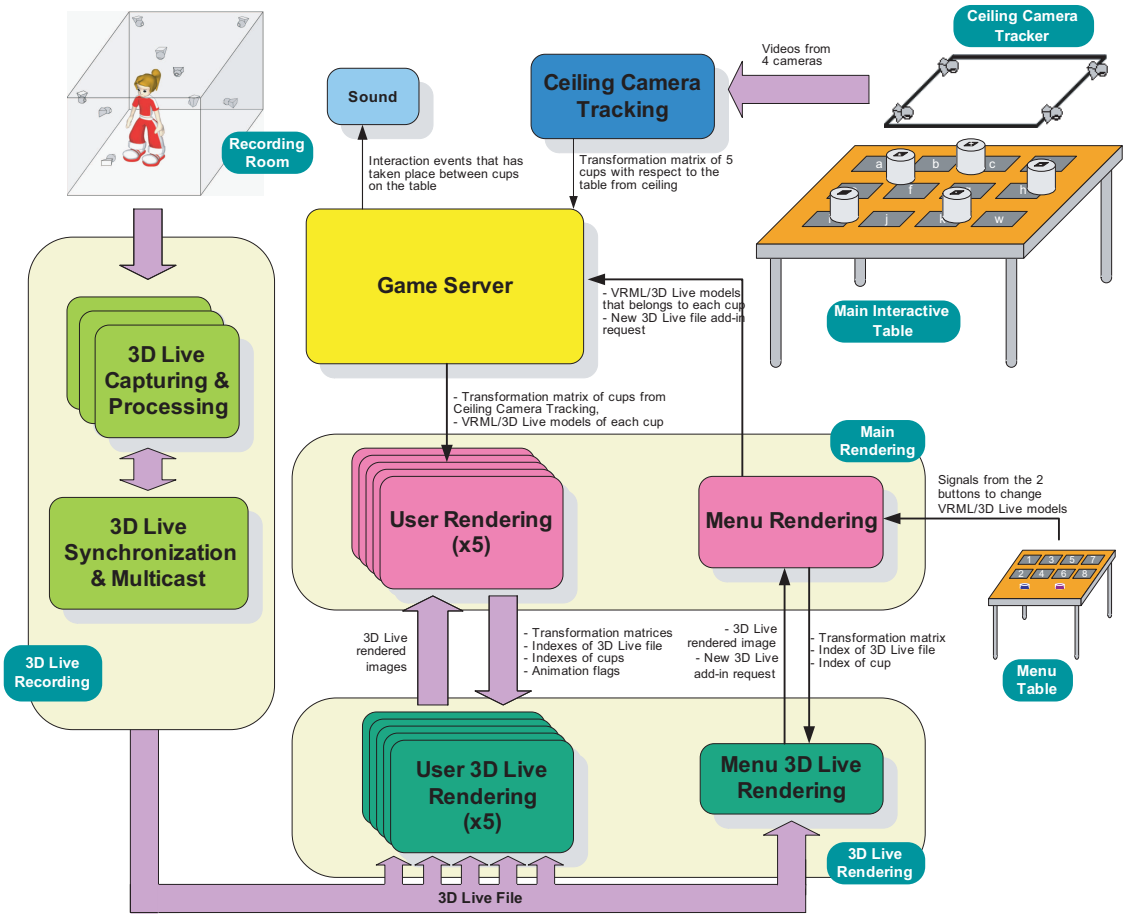


Figure 6.4: System Setup of Magic Land

In this system users can record their live model for playback. The 3D Live Recording and 3D Live Rendering parts are a recording capturing system described in the previous section. After going inside the recording room and pressing a button, the user will be captured for 20 seconds. The captured images are then processed and sent to all 3D Live Rendering modules. However, unlike the live version which sends the processed images of nine cameras immediately for each

frame, the recorded version sends all the processed images of all the frames captured in 20 seconds at a time. Another difference is that, instead of using TCP/IP to send the 3D Live data to each User 3D Live Rendering and Menu 3D Live Rendering module of the 3D Live Rendering part, we use multicast to send the data to all of them. This helps to utilize bandwidth of the network as well as to ensure that all the receivers finish receiving data at the same time.

The Main Rendering part includes a Menu Rendering module and five User Rendering modules. These modules track the users' viewpoints, and render the corresponding images to the users. First, they obtain images from the Unibrain cameras mounted on the users' HMDs, track the marker patterns and calculate the transformation matrix relating the coordinates of these markers with the coordinate of the camera. After that, basing on the transformation matrix, each module will render the image and output the result to the corresponding HMD. Especially, the Menu Rendering module also handles the users' inputs when they press the buttons on the Menu Table, or when they use the cups to select and remove virtual characters.

The Ceiling Camera Tracking module receives images from four cameras put above the Main Interactive Table. It tracks the markers of the table and cups, and calculates the transformation matrices of the cups relative to the table from top view. After that, it sends these matrices to the Game Server.

Last but not least, Game Server is the heart of the system, which links all the modules together. It receives and forwards information from the Ceiling Camera

Tracking, Menu Rendering and User Rendering modules. This Game Server coordinates and synchronizes what every user has in their cup in terms of type of the character and its animation, position and orientation. First of all, it receives the camera tracking data from the Ceiling Camera Tracking module and determines the interaction between the characters inside the cups, basing on the distances between cups. After that, it forwards this interaction information to the User Rendering and Sound modules so that these modules can render the respective animations and produce the corresponding interactive sound. The ceiling camera tracking data is also forwarded to the User Rendering modules for usage in the case that the users's camera lost the tracking of their cups' marker. When the users select a new character, the Game Server also receives the new pair of cup-character indexes from the Menu Rendering and forwards to all the User Rendering modules to update the change.

6.3 Artistic Intention

Magic Land demonstrates novel ways for users in real space to interact with virtual objects and virtual collaborators. Using the tangible interaction and the 3D Live human capture system, our system allows users to manipulate the captured 3D humans in a novel manner, such as picking them up and placing them on a desktop, and being able to "drop" a person into a virtual world using users' own hands. This offers a new form of human interaction where one's hands can be used to interact

with other players captured in 3D-Live models.

The artistic aspect of this installation introduces to artists easy, tangible and intuitive approaches in dealing with mixed reality content. The main challenge of the project is to create a new medium located somewhere between theater, movie and installation. The outcome of the project is an infrastructure that gives artists new opportunities to transport audiovisual information and encourage artists of any discipline to deal with those new approaches.

We can perceive Magic land as an experimental laboratory that can be filled by a wide range of artistic content, which is only limited by the imagination of the creators. To watch the scene from above with the possibility of tangible manipulation of elements creates a new form of art creation and art reception that generates an intimate situation between the artist and audience.

The project itself brings together the processes of creation, acting and reception in one environment. These processes are optimized to the visitors experience in order to better understand the media and lead to a special kind of self reflection. The recording area plays the role of the interface between human being and computer. It is also a special experience for the users to watch themselves acting in 3D on the interactive table from the external point of view like the “Bird in the sky”. In Figure 6.5 are two bird’s eye views of this system.

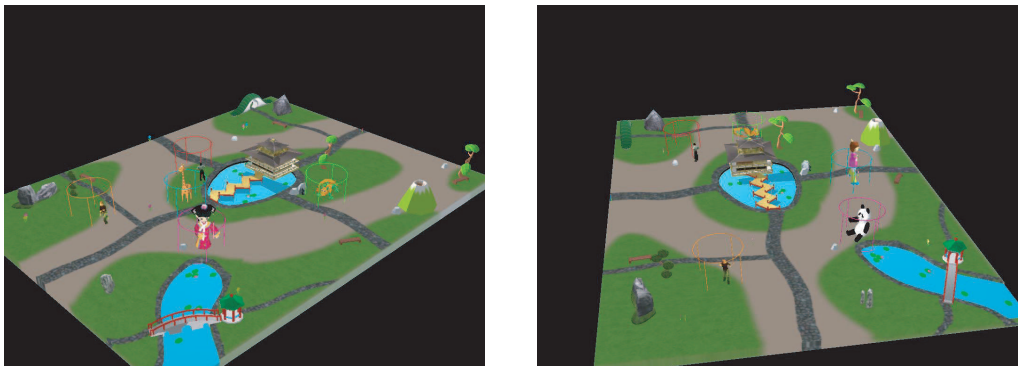


Figure 6.5: **Main Table**: The bird's eye views of the Magic Land. One can see live captured humans together with VRML objects

6.4 Future Work

Currently, we are developing a new version of Magic Land by exploiting the "real time" capability of 3D-Live technology, in which, outside players can see on the Main Interactive Table the 3D images of the one who is being captured inside the room in real time. Instead of sending all the processed images of all the frames captured in 20 seconds at a time, this version uses RTP [40] and IP multicast to stream the processed images to all User 3D Live Rendering modules immediately for each frame. To guarantee continuous rendering, User 3D Live Rendering modules will buffer these images for a number of received frames before generate the 3D images inside one of the special cup on the Main Interactive Table. Moreover, inside the recording room, the captured player wears an HMD to view the virtual environment in front of her at the viewpoint corresponding to the position of the cup on the table. The HMD is connected to a computer outside by a small cable going through the ceiling of the recording room. The cable is painted the same

color with the room and its width is small enough to be eliminated by the filter step of 3D Live background subtraction and image processing modules.

In this context, the captured player can actively interact with other virtual objects in virtual reality environment when seeing them on the HMD, and the outside players will have fun seeing her reaction the in mixed reality environment. In our further future work, we want to explore the problem of whether the cup which represent for the 3D-Live object can automatically move when the captured player moves inside the room. Such a system will give the captured person more freedom exploring the whole virtual world herself. Technologies in Touchy Internet [41] can be applied to automatically move the special cup around the table. Touchy Internet uses special sensors and wireless system to track the movement of a pet at home backyard and control the doll's movement placed at the office corresponding to the pet's movement.

The future version of Magic Land will open a new trend for mixed reality games, in which players can actively play a role of a main character in the game story, be submerged totally in the virtual environment, and explore the virtual world themselves, while at the same time in mixed reality environment, other players can view and construct the virtual scene and new virtual characters to challenge the main character. Consequently, the game story is not fixed but will depend on the players' creativity and imagination, and follow their reactions when they travel around the virtual word.

6.5 Magic Land's Relationship with Mixed Reality Games

Nowadays, computer games have become a dominating form of entertainment due to their higher level of attractiveness to game players. There are some superior advantages which make computer games more popular than traditional games. Firstly, it attracts people by creating the illusion of being immersed into imaginative virtual world with computer graphics and sound [42]. Secondly, the goals of computer games are typically more interactive than that of traditional games, which brings players stronger desire to win the game. Thirdly, usually designed with the optimal level of information complexity, computer games can easily provoke players' curiosity. Consequently, computer games intrinsically motivate players by bringing them more fantasy, challenge and curiosity, which are the three main elements contributing the fun in games [43]. Moreover, compared with many traditional games, computer games are also easier to play at any individual's preferred location and time.

However, the development of computer games has often decreased their physical activities and social interactions. Addressing this problem, growing trends of nowadays game, especially mixed reality games, are trying to fill in this gap by bringing more physical movements and social interactions into games while still utilizing the benefit of computing and graphical systems.

A typical VR game CAVE Quake [44] increases the player's sense of 3D space

by surrounding them with life-sized 3D virtual world, instead of constraining them within a limited 2D screen. However, CAVE Quake players still in lack of physical movement, tangible interactions and social communications.

AR2 Hockey [45], an air-hockey AR game in which users use real mallet to play with a virtual puck on a real table, enhances physical interactions and social communication, but does not utilize the graphical power of computer systems.

AquaGauntlet [46] is another AR game in which several players gather in a small place with some physical egg-shape objects to shoot computer-generated creatures superimposed onto the real scene as if they came from these egg-shape objects. This game enhances physical interactions and social communication, and also utilizes the graphical power of computer system. However, players of AquaGauntlet, as well as AR2 Hockey, still have limited movement and little interaction with the physical space (as they must stand in a fairly constant location).

Another embodied computing based mixed reality game which also enhances physical interactions and social communication is Touch-Space [47]. This game is carried out in the physical world with a room-size space where two players will collaboratively finish some tasks and then rescue a princess in castle controlled by a witch. This game provides different levels of interaction in different environments: physical environment, augmented reality, and virtual reality. However, all these interactions are limited in a room-size space and only for two users.

Pirates! [48] and Human Pacman [49] are two typical outdoor mixed reality games aiming for enhancing physical activities and social interactions as much ex-

tent as possible. Pirates! uses handheld computers and proximity-sensing technology to make real world properties, such as locations or objects, important elements of game mechanics. Meanwhile, in Human Pacman, the player who acts as "Pacman" wearing a wearable computer and an HMD goes around the physical game space to collect cookies, where as other player acting as 'ghost' will find and touch to kill the Pacman. There are two other players acting as Pacman's and Ghost's helpers sitting inside offices, using computer's graphical information to search their enemy's locations in order to help their partners. These games are very successful in term of enhancing physical interactions and social communications, however, they have not utilized fully graphical power of computing system to create an appealing imaginative virtual world. Pirates is played on a PDA screen which does not allow a 3D mixed reality experience. Human pacman requires quite heavy and bulky wearable computers and equipment.

Compared with the above typical AR/VR games, as an indoor mixed reality and tangible interaction game, Magic Land exploits physical tangible interaction, social interaction and also utilizes 3D graphics rendering to create an attractive imaginative virtual world. Moreover, the act of putting 3D images of real human beings in to that inventive world and making them new characters of the game story is unique in game context. Most importantly, Magic Land is a kind of "free play" game [50], in which players are free to use their imagination and creativity to design the game story and rules. Thus, as mentioned before, the game story and rules is not fixed but depends on players' imagination and decision.

Table 6.1: Comparison of Magic Land with other mixed reality games

Games	**Advantages**	**Disadvantages**
CAVE Quake	Significantly increase players' sense of 3D space by fully immersing them into a 3D virtual world. Provide beautiful graphics and interesting game story.	Very limited physical movement. No tangible interaction and social communication.
AR2 Hockey	Provide 3D mixed reality experience and tangible interaction with virtual object. Enhance social communication.	Limited physical movement and tangible interaction. No attractive 3D graphics of virtual world.
AquaGauntlet	Provide 3D mixed reality experience, tangible interaction and nice 3D graphics of virtual characters. Enhance social communication.	Limited physical movement and tangible interaction.
Touch Space	Tangible interaction with virtual object, enhance social communication, nice graphical virtual characters in mixed reality world. Different levels of interaction in different environments: physical environment, augmented reality, and virtual reality.	Limited physical movement and number of players.
Pirates!	Provide physical movement and social interaction to great extent.	Limited tangible interaction and graphical virtual characters. No 3D mixed reality experience.
Human Pacman	Provide physical movement and social communication to large extent. Enhance 3D mixed reality experience and tangible interaction.	Physical movement is slightly limited due to wearable computer and HMD.
Magic Land	Provide varied tangible interaction with virtual objects, beautiful 3D mixed reality virtual scene and characters, and social interactions among players. *Players can be captured and become new characters encountering with other virtual characters in mixed reality world.*	Not fully provide physical movement like outdoor games such as Pirates! and Human Pacman.

6.6 User Study of Magic Land 3D-Live system

6.6.1 Aim of this User Study

We conducted this user study for our mixed reality Magic Land 3D-Live system in order to obtain the feedback from the users regarding their perception to this new technology system. For example, their feeling on interacting with the virtual objects, being captured in 3D in a special recording room, etc. This survey also helps to assess the performance of this system to check how much this system promotes social interaction and remote 3D collaboration. The improvements that may be continuously made in the future work are also expected to obtain from this user study.

6.6.2 Design and Procedures

Thirty subjects (13 Females and 17 Males) were invited to participate in this study. The age group of the subjects ranges from 15 years old to 54 years old, with the average age of 25.4 years old. All of them reported clear-vision and normal hearing abilities.

During the user study, each subject will go into the 3D-Live recording room first, and follow the system voice instructions to record herself. After the recording is finished, the system will ask her to leave the recording room and go to the Menu Table and wait for her 3D data to be transferred over. Once her captured 3D-Live data being sent to the Menu Table, the subject then can use the green button on

the Menu Table to find herself amongst the various recorded human characters. Once she has found herself, she can then use one of the empty cups to pick herself up, and put herself onto the main interaction table. She can then go and pick some more captured human 3D-Live characters and virtual 3D VRML characters and add to the interaction table, and try the interactions among them. Subjects were also encouraged to play this system together with their friends at the same time (social collaboration).

After the subjects tried all the functions of this system, they were asked to fill in a questionnaire paper with 13 questions as follows:

6.6.3 Results of this User Study

Question 1 and 2 are used to assess the overall feelings of the subjects to Magic Land. The two main features here in this system are merging the user into the virtual world, and interacting with other virtual objects. From the feedback, we found that 25 subjects out of 30 felt *Very Exciting* about the concept of merging themselves into the virtual world; and 20 subjects felt *Very Exciting* about the concept of interacting with virtual object. From this results, we can see that, this technology is indeed very attractive to the general public.

Question 3 is concerning about how much this technology can help to promote the social interaction. The feedback was quite positive. In total, there were 20 subjects feeling that this system can help in promoting social interaction, and 6 of them felt that it is *Very helpful*. Question 4, 5 and 6 are concerning about the 3D-

Table 6.2: Questions in the user study

Questions	**A**	**B**	**C**	**D**
1. Overall, how do you rank the Magic Land as a concept of merging yourself into the virtual world?	Very exciting.	Exciting.	Moderate.	Boring.
2. Overall, how do you rank the Magic Land as a concept of interacting with virtual objects?	Very exciting.	Exciting.	Moderate.	Boring.
3. In your view, how much does this technology promote social interaction?	Very helpful in promoting social interaction.	Some help in promoting social interaction.	Only little help in promoting social interaction.	No help at all in promoting social interaction.
4. How do you feel about being captured in 3D in the special recording room, and then shown in the virtual world on the menu and interactive table?	Comfortable and good	Moderate, same as taking normal photos.	Uncomfortable.	Nervous, and feel uneasy.
5. Would you like to have such 3D-Live system for remote 3D collaboration in the future? Here, collaboration means you can see someone remotely in 3D (different from traditional 2D video conference), and work and play with him/her together.	Yes, looking forward to trying it.	Yes, it might be a good idea.	I don't really care.	No, I don't think it will work well.
6. How collaborative is this system if we implement it for the remote 3D collaboration? Here, collaboration means you can see someone remotely in 3D (different from traditional 2D video conference), and work and play with him/her together.	Very collaborative. Everyone is working closely together to achieve the goal.	There is some level of collaboration here.	Only little collaboration here.	No collaboration at all, it's basically a single player system.
7. How entertaining is this system to you?	Very fun! I really enjoyed it.	It is a nice game. Good for playing occasionally.	It is about the same as the other games. Not much difference.	I don't like this game. It is not entertaining at all.

Table 6.3: Questions in the user study (cont.)

Questions	**A**	**B**	**C**	**D**
8. Compare to current 2D communications such as web cam, do you think this system is useful for communication in telepresence? Telepresence is to use computer technology to give the appearance of an individual being present at a location other than the actual location of that individual.	Very useful. It will make telepresence communication very exciting.	It may be useful for the telepresence communications.	Telepresence communication may not be very different from the current 2D communications.	I don't think this system can help in communication in telepresence.
9. Do you like the idea of using a physical cup to pick up the virtual objects or 3D-Live characters, comparing to using mouse and keyboard as in traditional computer games	It is good. The cup is easy to use for picking up objects.	I think there is no difference between using a cup and using normal mouse and keyboard for the control.	I prefer to use mouse and keyboard instead.	
10. How do you feel about the control of interaction between objects by moving the cups around on the interactive table?	Very interesting. I like this way of control of interactions.	It is ok. But the cup is not so easy to move around on the table.	This kind of control is fine. But I also like to use traditional mouse and keyboard to control.	I don't like to cups.
11. In your view, how much do the physical cups promote the social collaboration to make interactions on the table, comparing to the traditional way of using keyboard and mouse?	Very helpful in promoting social interaction.	Some help in promoting social interaction.	Only little help in promoting social interaction.	No help at all in promoting social interaction.
12. How do you feel about the deleting of the objects in the cup by using a virtual trash can?	Very easy to use. It is a good idea.	It is fine.	I don't feel special, neither do I like it.	I don't like this way of deleting.
13. Would you like to try this kind of system again in the future?	Yes. I am looking forward to it.	Yes, maybe I will try.	I don't want to try any more.	

Live recording room. 18 subjects felt that 3D-Live recording process is *Comfortable and good*, and 9 felt *Moderate*. Only 3 subjects felt that the recording processing is *Uncomfortable* or make them *Nervous and feel uneasy*. 73.4% of the testing subjects felt this system can be used for remote 3D collaboration in the future, and 63.4% of the testing subjects believed such system will be collaborative. It shows that, this 3D-Live capturing process can be accepted by most of the population. The feedback of question 8 shows that, nearly two third of the testing subjects think this system is useful in tele-presence comparing to the current 2D video teleconferences.

Another important part of this Magic Land system is that, we are using physical cups to pick up and move the virtual objects or 3D-Live characters. From the answer to question 9, 10, and 11, we can see that most testing subjects like the way of using physical cups comparing to using mouse and keyboard as in traditional computer games. Comparing to mouse and keyboard, 17 subjects felt the cups were easier for picking up and virtual objects, and 18 subjects felt the cups were easy to move the objects around. Also there were 18 subjects feeling that using cups is helpful in promoting social interaction.

As a multimedia system, we also value how entertaining this system is through question 7. As a result, 10 subjects enjoyed the game a lot, and 11 said *It is a nice game. Good for playing occasionally.* This result is quite encouraging to apply this technology in further digital entertainment development. To check how friendly the user interface is, we put question 12 to see how the users will feel about the

way of deleting virtual object from the cup. It showed that 70% of the subjects like our idea of using a virtual trash can. And from question 13, we can see that more than 90% of the subjects liked to try this kind of system again in the future.

6.6.4 Conclusion of the User Study

Overall, from the user study we can conclude that our Mixed Reality Magic Land 3D-Live system is testified to have produced a tangible, natural and novel interaction interface to the users.

Most of the testing subjects claimed that this system is very attractive to them, and they were excited to see themselves being captured in 3D, and then being put into the interaction table together with the other 3D objects. Although few people complained that the 3D-Live capturing process makes them feel uncomfortable or nervous, most testing subjects felt comfortable or natural with the system. So, we can say that this 3D-Live system is acceptable by the general public, and maybe minor modifications can be made to make it more user friendly.

From the results, we can see that most testing subjects felt the mixed reality technology helps to promote the social interaction among the participants. More than half of the participants think this technology will be useful for the remote 3D collaboration system in the future. But still a few of the testing subjects think there is a little collaboration in this system or no collaboration at all. The reason for this should be that all the 3D-Live characters we used now are captured separately, no relationship among them. But when the technology be used in the remote 3D

collaboration in the future, the captured characters must be related, user should have different feeling.

Using the physical cups instead of using the traditional mouse and keyboard is also proved to be a more natural way of controlling the virtual objects from the results of the user study. Most participants felt it easy to use, and helpful in promoting the social interaction. Also we can see that most of the users think it is a good idea to use virtual trash can to delete the objects. This result shows that the mixed reality technology provided a natural user interface.

Additionally, further improvements to this system may be made in increasing the gaming complexity and hardware refinement. There were still 30% of the testing subjects feeling that, this system is not so entertaining. We can improve that by adding more meaningful interactions, 3D sound effects, better computer graphics, etc.

The diagram of the results of all these 13 questions can be viewed from Figure 6.6.

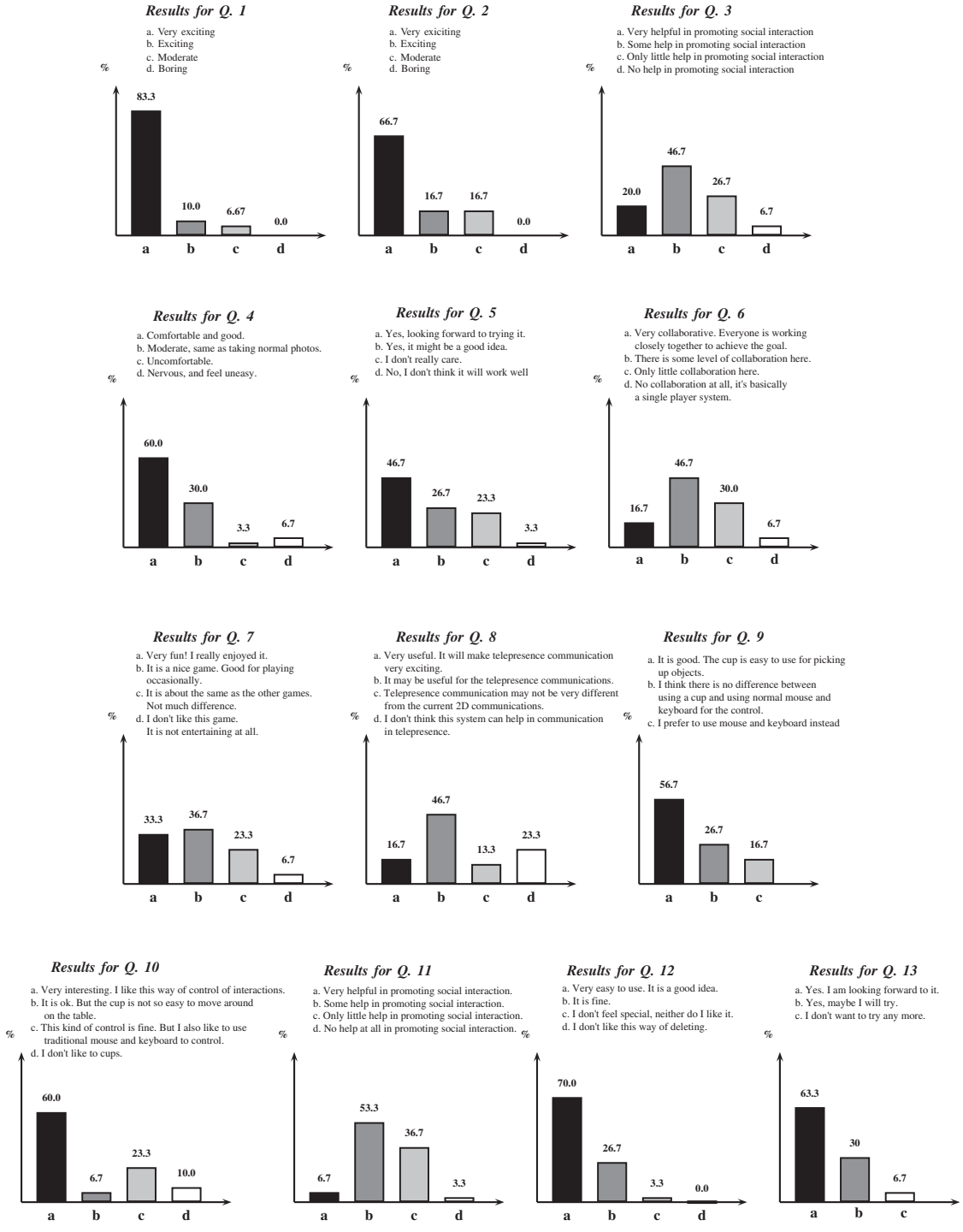


Figure 6.6: Graph results for multiple choice questions

Chapter 7

Conclusion

7.1 Summary

This thesis has introduced a complete system for capturing and rendering humans and objects in full 3D. The ultimate goal of this project is to achieve real-time 3D communication, that is closer in spirit to the kind of perfect tele-presence made popular by the Star Wars movies. This allows humans to communicate with each other unrestricted as if the other person was really standing in front of him/her. It would be a more complete experience because various body gestures and other nonverbal cues that were suppresses by other communication media could then be fully expressed.

The whole 3D-Live system has been presented in details in Chapter 3. This is a complete and robust real time and live human 3D recording system, from capturing images, processing background subtraction, to rendering for novel view

points. Many issues in designing and implementing this system have been described and addressed in this chapter.

In chapter 4, the thesis has gone through different methods to improve the image-based novel view generation algorithm. It introduces new ways to compute visibility and blend color in generating images for novel viewpoints. These contributions have significantly improved quality and performance of the system, and are very useful for mixed reality researchers.

After that, in chapter 5, the early stages of the development of a model based novel view generation approach has shown a lot of promise. The feasibility and potential advantages of this method has now been revealed, and future work on this area could take 3D-Live closer to achieving real time 3D communication.

Going beyond communication, Magic Land has demonstrated the potential of 3D-Live technology in interactive art and entertainment. The unique combination of mixed reality, tangible interaction and digital art creates an unparalleled novel experience that has been shown through many different conferences and exhibitions over the world and is now a permanent exhibit at the Singapore Science Center. Results of the survey on Magic Land's users reveal some important issues and emphasize the effectiveness of 3D-Live, mixed reality, and tangible interaction on Human Computer Interaction.

7.2 Future Developments

3D-Live will provide a framework for many breakthrough and pioneering human computer communication and interaction technologies in the future. In the future, the following enhancements are foreseeable.

The image-based algorithm described in this thesis has not utilize the color information from images captured from camera. Thus, the generated result is only a visual hull, not a photo-hull. In the future, we will check the color consistency among captured images to acquire better rendering results.

The model-based algorithm presented in chapter 5 is based on the assumption "Nearby 3D points project to nearby image points". This assumption could be violated by objects with large depth discontinuities, or by self occlusion. One possible solution for this problem is checking the differences in depths of vertices. If two nearby vertices are too far from each other, they will not be connected. This solution will be approached in the future.

Moveover, currently, the described algorithm computes and throws away a different mesh for each frame of video. For some applications, it might be useful to derive the mesh of the next frame as a transformation of the mesh in the original frame and to store the original mesh plus the transformation function. Temporal processing such as this would also enable us to accumulate the texture (radiance) of the model as it is seen from different viewpoints. Such accumulated texture information could be used to fill in parts that are invisible in one frame with information

from other frames. This will significantly increase the speed of the algorithm.

For Magic Land, the current limit of this system is that human 3D avatars do not really interact with virtual objects. It is because we are using pre-recorded data and we lack of active feedbacks from captured persons. So, in the next step, we will implement a real-time capture system where players at the table can interact with the real-time avatar of the player being captured inside the recording room. And at the same time, the captured person would receive feedback from the system about her/his location in Magic Land, while other users move him around with the cups. What we intend to do is replacing the green wall by high frequency screens. These screens will frequently switch between displaying only a green screen and showing the virtual environment where her/his avatar is placing. By this way, we will provide the captured person the real-time ego-centric view and she/he will be totally immersive in the virtual world (VR) and will be able to feel all interactions in the realest way. For example, when the cup is placed in front of a dragon, the person inside the room will see this dragon standing right in front of her/him and maybe blowing fire toward her/him also.

Furthermore, the capture user could also affect the VR by reacting appropriately to her/his current status in the virtual environment. She/he could actually interact with the user and other subjects through body position and movement. For example, the position and movement of the captured person will decide how virtual objects interact on the table.

7.3 Conference and Exhibition Experience

Up to now, Magic Land has been shown to both academic research community and public at different conferences and exhibitions. In Singapore, it was first shown to public during the Planet Game exhibition at Singapore Science Center from September 2004 to February 2005. Currently, Magic Land is a permanent exhibition at this science center. It has also been shown at the Interactivity Chamber of SIGCHI 2005, organized at Portland, USA, in April 2005. Most recently, in June 2005, Magic Land was demonstrated for around 30,000 attendees during the WIRED NextFest Exhibition at Chicago, USA. This is a huge exhibition of around 120 projects which has been selected through a worldwide search for cutting-edge prototypes, installations, proof-of-concepts and other emerging technologies.

Figure 7.1: Exhibition at Singapore Science Center

Figure 7.2: Demonstration at SIGCHI 2005

Figure 7.3: Demonstration at Wired NextFest 2005

Bibliography

[1] E. Trucco and A. Verri. *Introductory Techniques for 3D Computer Vision.* Prentice Hall, 1998.

[2] G. Slabaugh, W. B. Culbertson, T. Malzbender, and R. Schafer. A survey of volumetric scene reconstruction methods from photographs. In *Proc. International Workshop on Volume Graphics*, 2001.

[3] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. In *International Journal of Computer Vision, Vol.38, No. 3*, 2000.

[4] G. G. Slabaugh, W. B. Culbertson, T. Malzbender, M. R. Stevens, and R. W. Schafer. Methods for volumetric reconstruction of visual scenes. In *The International Journal of Computer Vision*, 2004.

[5] Y. H. Fang, H. L. Chou, and Z. Chen. 3d shape recovery of complex objects from multiple silhouette images. *Pattern Recognition Letters*, 24:1279 – 1293, 2003.

[6] W. Matusik, C. Buehler, and L. McMillan. Polyhedral visual hulls for real-time rendering. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, 2001.

[7] R. Azuma. A survey of augmented reality. In *Presence*, 1997.

[8] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. In *IEEE Computer Graphics and Applications*, Nov./Dec.2001.

[9] G. Markus, W. Stephan, N. Martin, L. Edouard, S. Christian, K. Andreas, K. M. Esther, S. Tomas, V. G. Luc, L. Silke, S. Kai, V. M. Andrew, and S. Oliver. blue-c: A spatially immersive display and 3d video portal for telepresence. In *Proceedings of ACM SIGGRAPH*, 2003.

[10] J. M. Hasenfratz, M. Lapierre, and F. Sillion. A real-time system for full body interaction. *Virtual Environments*, pages 147–156, 2004.

[11] S. J. D. Prince, A. D. Cheok, F. Farbiz, T. Williamson, N. Johnson, M. Billinghurst, and H. Kato. 3d live: real time captured content for mixed reality. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2002.

[12] M. Billinghurst and H. Kato. Real world teleconferencing. In *Proc. of the conference on HFCS (CHI 99)*, 1999.

[13] http://www.hitl.washington.edu/artoolkit/.

[14] M. Billinghurst, A.D. Cheok, S.J.D. Prince, and H. Kato. Projects in VR - Real World Teleconferencing. In *IEEE Computer Graphics and Applications, Volume 22*, 2003.

[15] MXRToolkit. [Online]. Available at: http://sourceforge.net/projects/mxrtoolkit/.

[16] P. Fua. From multiple stereo views to multiple 3-d surfaces. *International Journal of Computer Vision*, 24(1):19–35, 1997.

[17] T. S. Huang and A. N. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE*, 82(2):252–268, 1994.

[18] C. R. Dyer. Volumetric scene reconstruction from multiple views. In *Foundations of Image Understanding*, 2001.

[19] H. H. Chen and T. S. Huang. A survey of construction and manipulation of octrees. *Computer Vision, Graphics, and Image Processing*, 43(3):409–431, 1988.

[20] C. H. Chien and J. K. Aggarwal. Volume / surface octrees for the representation of three-dimensional objects. *Computer Vision, Graphics, and Image Processing*, 36(1):100–113, 1986.

[21] S. Srivastava and N. Ahuja. Octree generation from object silhouettes in perspective views. *Computer Vision, Graphics and Image Processing*, 49(1):68–84, Jan. 1990.

[22] M. Potmesil. Generating octree models of 3d objects from their silhouettes in a sequence of images. *Computer Vision, Graphics andand Image Processing*, 40(1):1–29, Oct. 1987.

[23] R. Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics and Image Processing: Image Understanding*, 58(1):23–32, July 1993.

[24] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1997.

[25] W. Martin and J. K. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, 1983.

[26] W Matusik, C Buehler, R Raskar, S J Gortler, and L McMillan. Image-based visual hulls. *Proc. SIGGRAPH*, pages 369–374, 2000.

[27] Point Grey Research Inc. [Online]. Available at: http://www.ptgrey.com.

[28] ARToolKit. [Online]. Available at: http://www.hitl.washington.edu/artoolkit/.

[29] OpenCV. [Online]. Available at: http://sourceforge.net/projects/opencvlibrary/.

[30] S. J. D. Prince, A. D. Cheok, F. Farbiz, T. Williamson, N. Johnson, M. Billinghurst, and H. Kato. Live 3-dimensional content for augmented reality. In *IEEE Transactions on Multimedia (submitted)*.

[31] T. Horprasert et al. A statistical approach for robust background subtraction and shadow detection. In *Proc. IEEE ICCV'99 Frame Rate Workshop, Greece*, 1999.

[32] M. Seki, H. Fujiwara, and K. Sumi. A robust background subtraction method for changing background. In *Proceedings of the Fifth IEEE International Workshop on Applications of Computer Vision*, 2000.

[33] R. Mester, T. Aach, and L. Dmbgen. Illumination-invariant change detection using statistical colinearity criterion. In *DAGM2001, number 2191 in LNCS. Springer*, pages 170–177.

[34] RGB "Bayer" Color and MicroLenses. [Online]. http://www.siliconimaging.com/RGB Bayer.htm.

[35] G. Slabaugh, R. Schafer, and M. Hans. Image-based photo hulls. In *Proceedings of the 1st International Symposium on 3D Data Processing, Visualization, and Transmission*, 2002.

[36] R. Szeliski. Video mosaics for virtual environments. In *IEEE Computer Graphics and Applications*, March 1996.

[37] Remondino Fabio. From point cloud to surface:the modeling and visualization problem. In *International Workshop on Visualization and Animation of Reality-based 3D Models*, volume XXXIV-5, Tarasp-Vulpera, Switzerland, February 2003. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences.

[38] Peter Savadjiev, Frank P. Ferrie, and Kaleem Siddiqi. Surface recovery from 3d point data using a combined parametric and geometric flow approach. Technical report, Centre for Intelligent Machines, McGill University, 3480 University Street, Montral, Qubec H3A 2A7, Canada.

[39] H. Kato, K. Tachibana, M. Tanabe, T. Nakajima, and Y. Fukuda. Magiccup: A tangible interface for virtual objects manipulation in table-top augmented reality. *Proceedings of Augmented Reality Toolkit Workshop (ART03)*, pages 85–86, 2003.

[40] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Rtp: A transport protocol for real-time applications. *Internet Engineering Task Force, Audio-Video Transport Working Group*, 1996.

[41] S. P. Lee, F. Farbiz, and A. D. Cheok. Touchy internet: A cybernetic system for human-pet interaction through the internet. *SIGGRAPH 2003, Sketches and Application*, 2003.

[42] A. Amory, K. Naicker, J. Vincent, and C. Adams. The use of computer games as an educational tool: identification of appropriate game types and elements. 30(4):311–321, 1999.

[43] T. W. Malone. Toward a theory of intrinsically motivating instruction. 5:333–369, 1981.

[44] CAVE Quake II. [Online]. Available at: http://brighton.ncsa.uiuc.edu/ prajlich/caveQuake/.

[45] T. Oshima, K. Satoh, H. Yamamoto, and H. Tamura. Ar2 hockey system: A collaboration mixed reality system. 3(2):55–60, 1998.

[46] H. Tamura, H. Yamamoto, and A. Katayama. Mixed reality: Future dreams seen at the border between real and virtual worlds. 21(6):64–70, 2001.

[47] A. D. Cheok, X. Yang, Z. Zhou, M. Billinghurst, and H. Kato. Touch-space: Mixed reality game space based on ubiquitous, tangible, and social computing. *Journal of Personal and Ubiquitous Computing*, 6(5/6):430–442, 2002.

[48] S. Bjork, J. Falk, R. Hansson, and P. Ljungstrand. Pirates! - using the physical world as a game board. In *Interact 2001, IFIP TC. 13 Conference on Human- Computer Interaction*, Tokyo, Japan, 2001.

[49] A. D. Cheok, S. W. Fong, K. H. Goh, X. Yang, W. Liu, and F. Farzbiz. Human pacman: A sensing-based mobile entertainment system with ubiquitous computing and tangible interaction. 2003.

[50] R. L. Mandryk and K. M. Inkpen. Supporting free play in ubiquitous computer games. 2001.