# VIRTUAL VIEW SYNTHESIS USING VISUAL HULLS



Nicholas Frank Maunder

Dissertation submitted to the Department of Electrical Engineering at the University of Cape Town in fulfilment of the requirements for the degree of

Master of Science in Engineering

August 2005

# Acknowledgements

I would like to thank my supervisors Professor Gerhard de Jager and Dr Fred Nicolls for their guidance during this project. The technical support from Dr Nicolls is greatly appreciated. I would also like to thank the other members of the Digital Image Processing Research Group at UCT, including Keith Forbes, Bruno Merven, Mathew Price, and Phillip Milne for all their help and advice.

I am grateful to the De Beers Technology Group (GTS) for their generous financial support. The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at are those of the author and are not necessarily to be attributed to the NRF.

# Contents

**Abstract** i

**Acknowledgements** iii

**Contents** v

**List of Figures** vii

**List of Tables** ix

**1 Introduction** 1

1.1 Description of project . . . . . 2

1.2 Overview of Methodology . . . . . 3

1.2.1 Acquisition of the Reference Images . . . . . 5

1.2.2 Generating the Novel Views . . . . . 8

1.3 Thesis Outline . . . . . 9

**2 Theoretical Principles** 11

2.1 The Camera Model . . . . . 11

2.1.1 Intrinsic Parameters . . . . . 13

2.1.2 Extrinsic Parameters . . . . . 15

2.1.3 The Projection Matrix . . . . . 16

2.2 Multiview Geometry . . . . . 17

2.2.1 Epipolar Geometry . . . . . 17

2.2.2 Trifocal Geometry . . . . . 19

2.3 Visual Hulls . . . . . 20

**3 A Review of Virtual View Synthesis Techniques** **23**
3.1 Geometry-based Rendering Techniques . . . . . 23
3.2 Image-based Rendering Techniques . . . . . 27

**4 Geometry-based Rendering: Voxel Reconstruction and Texturing** **35**
4.1 Constructing Voxel Models using Octrees . . . . . 35
4.2 Adding Texture to the Voxel Models . . . . . 38

**5 Image-based Rendering: Image-based Visual Hulls** **41**
5.1 Computing the Visual Hull . . . . . 41
5.2 Texture mapping the Visual Hull . . . . . 45

**6 Results** **51**
6.1 Establishing a Measure of Performance . . . . . 52
6.2 Performance of the Geometry-based Rendering Technique . . . . . 54
6.3 Performance of the Image-based Rendering Technique . . . . . 68
6.4 Discussion of Results . . . . . 79

**7 Conclusions** **81**

**Bibliography** **84**

# List of Figures


1.1 Virtual view synthesis . . . . . 2
1.2 High-level flow diagrams for the implemented view synthesis methods . . . . . 4
1.3 Image acquisition using a turntable . . . . . 6
1.4 Calibration object for the VCAL calibration software . . . . . 6
1.5 The toy figurine data set . . . . . 8

2.1 The perspective projection camera model . . . . . 12
2.2 The extrinsic camera parameters . . . . . 15
2.3 Epipolar geometry . . . . . 18
2.4 The trifocal constraint . . . . . 21
2.5 Visual cone intersection defines an object's visual hull . . . . . 22

3.1 Voxel reconstruction using colour consistency . . . . . 25
3.2 The forward and backward mapping of image points . . . . . 28
3.3 Image transformation via homographies . . . . . 32

4.1 Octree representation of voxel models . . . . . 36
4.2 The resolution of a voxel model . . . . . 37
4.3 Rendering a voxel into an image via scan conversion . . . . . 39

5.1 Back projecting the two dimensional silhouette intersections into three dimensional space . . . . . 43
5.2 Determining where two visual rays meet by finding the closest approach . . . . . 43
5.3 The valid region of the epipolar line that must be searched . . . . . 45
5.4 The consequence of not determining the valid regions of the epipolar lines . . . . 46
5.5 Selecting the reference view that must be used for shading . . . . . 47
5.6 Testing the visibility of the surface points of the visual hull . . . . . 48

5.7 Visibility maps for the reference cameras . . . . . 49

6.1 Representing the position of a camera as an angular displacement . . . . . 54
6.2 Voxel reconstruction of a coffee cup . . . . . 55
6.3 Additional volume consistent with silhouettes . . . . . 56
6.4 Novel views of a coffee cup produced by the geometry-based approach . . . . . 58
6.5 Octree model of a ceramic cat . . . . . 59
6.6 Model of a ceramic cat showing added volume . . . . . 59
6.7 Novel views of a ceramic cat produced by the geometry-based approach . . . . . 61
6.8 Novel views of a toy figurine produced by the geometry-based approach . . . . . 63
6.9 The camera configurations when viewing the model radio . . . . . 64
6.10 Octree models of a radio viewed from above . . . . . 65
6.11 Dominant face of model radio changes as the angular displacement increases . . 66
6.12 Novel views of a radio produced by the geometry-based approach . . . . . 67
6.13 Visibility maps for the image-based approach . . . . . 69
6.14 A comparison of novel views rendered according to whether or not the visibility of surface points was taken into account . . . . . 70
6.15 Novel views of a coffee cup produced by the image-based approach . . . . . 71
6.16 Novel views of a ceramic cat produced by the image-based approach . . . . . 73
6.17 Novel views of a toy figurine produced by the image-based approach . . . . . 75
6.18 Error maps for the novel views of a toy figurine produced by the image-based approach . . . . . 76
6.19 Novel views of a radio produced by the image-based approach . . . . . 78

# List of Tables


6.1 Geometry-based rendering: Average error calculated for the ceramic cat data sets. 60
6.2 Geometry-based rendering: Average error calculated for the radio data sets. . . . 64
6.3 Image-based rendering: Average error calculated for the ceramic cat data set. . . 72
6.4 Image-based rendering: Average error calculated for the toy figurine data sets. . . 76
6.5 Image-based rendering: Average error calculated for the radio data sets. . . . . 79

# Chapter 1

# Introduction

Virtual or novel view synthesis refers to the process of generating a virtual view of a scene, or object, from a set of reference views. These reference views are the images obtained from a camera, or a number of separate cameras, positioned at different viewpoints around the scene. A virtual view of the scene represents what would be seen by a camera if it was positioned at a point not coinciding with the original reference cameras but having a common field of view (as shown in figure 1.1). Relevant information therefore needs to be extracted from the original reference views in order to render the image corresponding to the virtual viewpoint. This information can take the form of inferred geometrical cues or colour information for texturing.

Furthermore, video footage can be viewed as a sequence of still images. Therefore suitable view synthesis methods can be extended to generate virtual views of a dynamic event that varies over time.

There are a number of practical applications for virtual view synthesis. Synthesised views of a real scene or object can be used to enhance the experience of computer generated environments in the field of virtual reality. Similarly, in the field of augmented reality [2] being able to synthesise novel views from the images of a real object allows for the correct visualization of real-life objects that have been artificially placed in an observed scene.

Particular interest is being shown in the entertainment industry. One such area is that of film making, where synthetic views of scenes and objects can increase the realism of special effects and further extend the creative tools that producers and directors have at their disposal. Other

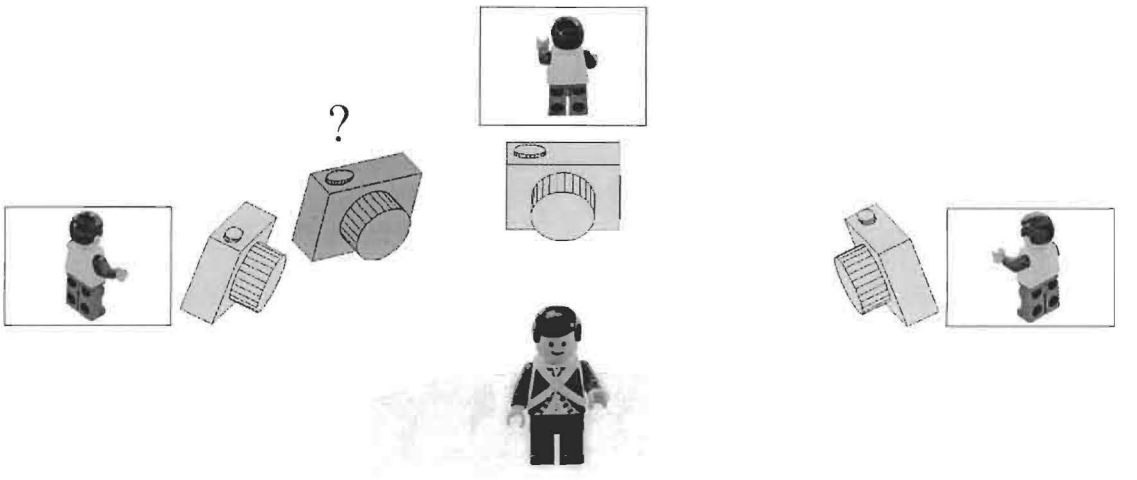



Figure 1.1: Virtual view synthesis. A virtual view represents what would be seen by a camera if it was positioned at a point not coinciding with the original reference cameras. The virtual camera is shown in blue.

areas include advertising on television and the video game industry [13].

Coverage of sporting events that take place in a stadium or arena could also be enhanced—synthetic views will allow for the smooth transition between the different cameras. Synthetic views can also aid in the visualization or even the virtual exploration of remote environments. A related idea is that of virtual teleconferencing [29].

## 1.1 Description of project

The aim of this project is to investigate the field of virtual view synthesis and demonstrate suitable methods for generating novel views of objects. Surveying the relevant literature reveals that the current techniques can be divided into two groups, namely those that first reconstruct a three dimensional geometric model of the observed object using the reference views and then render a novel view, and those that generate the new view directly from the reference views.

The methods belonging to the first group are referred to as *geometry-based* rendering systems, because the new image is formed by rendering the reconstructed geometric model. The methods

belonging to the latter group are termed *image-based* rendering systems, as the new view is rendered directly from the reference images [25]. In this work one technique from each of these groups is implemented and evaluated using a number of different data sets.

A third approach that generates novel views directly from the reference views was also studied. The approach is described in [16] and is based on image homographies and the parallax of warped image points. Due to time constraints it was only partially implemented and so no results are presented in this work. It is however introduced in chapter 3 during a review of the various view synthesis techniques.

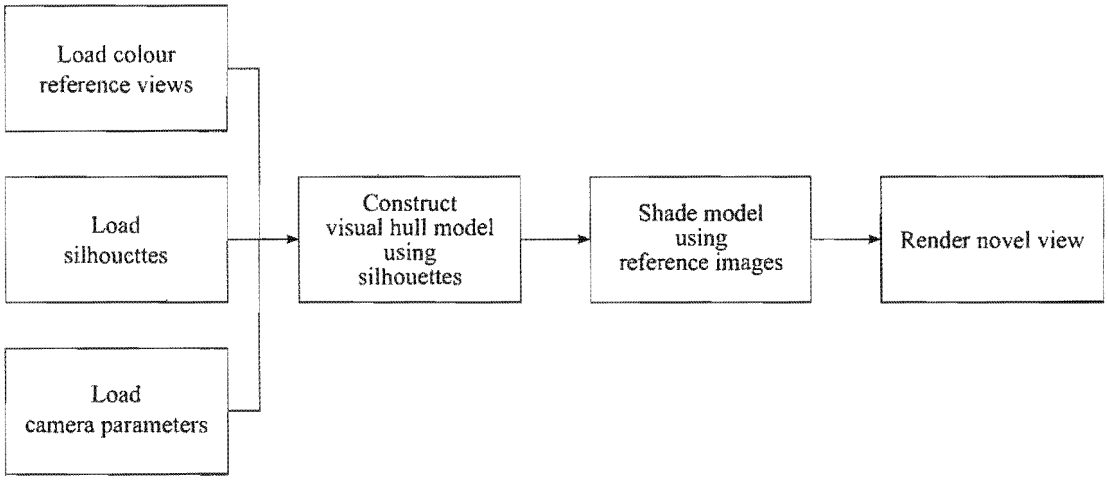
## 1.2 Overview of Methodology

The two techniques that were implemented are both built on the concept of the visual hull of an object. These methods are therefore suited to generating novel views of objects rather than of whole scenes. An approximation to the photographed object's visual hull can be computed from its silhouettes via the process known as volume intersection [19]. The colour reference images are then used to assign textures to the visual hull and with this information the novel view can be generated. A high-level flow diagram of the steps involved in each of the approaches is shown in figure 1.2.

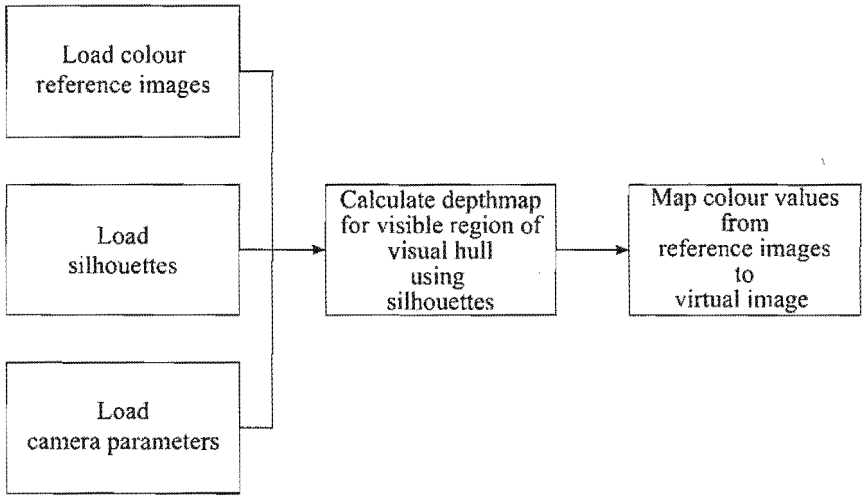
The implementations both take as input the reference views of the object, the silhouettes of the object, and the calibration parameters of the reference cameras and the virtual camera. The acquisition, segmentation and calibration of the reference images are thus treated separately from the actual view synthesis algorithms.

Once the required data has been loaded the geometry-based method uses the object's silhouettes to construct a *voxel* model of the visual hull. The mapping of texture onto the model is accomplished by assigning colour values to the voxels that lie on the surface of the model. These surface voxels are then rendered onto the image plane of the virtual camera, thereby generating the novel view.

The image-based solution that was implemented is based on the method proposed by Matusik et al. [24] entitled *Image-Based Visual Hulls*. This approach computes a viewpoint dependent rep-




(a) Steps involved in the geometry-based approach




(b) Steps involved in the image-based approach

Figure 1.2: High-level flow diagrams for the implemented view synthesis methods.

resentation of the object's visual hull which can be expressed as a depthmap relative to the virtual camera. The representation is viewpoint dependent because the algorithm is only concerned with the surface points of the visual hull that are visible from the virtual camera. The depthmap is then used to map colour values from the reference images to the virtual image. Since the visual hull computation is viewpoint dependent so is the novel view.

### 1.2.1 Acquisition of the Reference Images

The reference views consist of either images of real-life objects or images of artificial objects generated on a computer. The data sets consisting of real images were acquired using a single digital camera and, in some instances, a turntable. The computer generated images were created with three dimensional modelling software[1]. Precise calibration parameters could be obtained from the software and ground truth images of the novel views could be generated, which aids in the analysis of the output of the implemented view synthesis methods.

With all the data sets the viewpoints were positioned at approximately the same height above the ground plane and were arranged in a circular pattern around the object. The images were 640 by 480 pixels in size with the colour information being stored as 24-bit RGB values.

The silhouettes of the objects were obtained by manually segmenting the reference images. Automated segmentation proved problematic due to the background colour of the images that were captured using a turntable. In the case of the computer generated images the silhouettes were rendered separately through the manipulation of the modelling software.

#### The Coffee Cup Data Set

This data set consisted of five views of a coffee cup captured using an automated turntable. The turntable was driven by a stepper motor which was controlled by a computer. A single digital camera was placed at an elevated position in relation to the turntable, facing downwards at a slight angle as shown in figure 1.3.

The calibration of the camera was accomplished using the VCAL calibration software [11]. This

[1] Hash Animation Master 99 (http://www.hash.com)

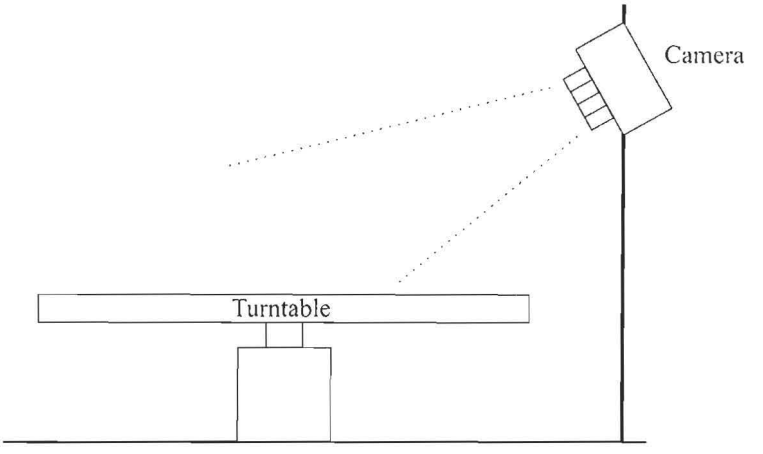



Figure 1.3: Image acquisition using a turntable and a single digital camera.

software employs a coded calibration object in order to determine the parameters for each of the five viewpoints—this object can be seen in figure 1.4. The object was placed at different locations on the turntable and, for each position, an image was captured from each of the viewpoints. Since the dimensions of the coded object are known the software can produce a metric calibration.

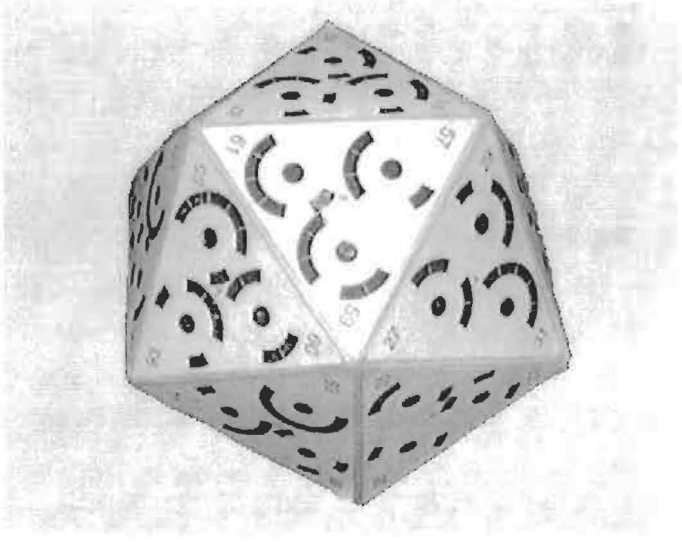


Figure 1.4: Calibration object for the VCAL calibration software.

Data sets consisting of more than five images of the cup were also obtained. Unfortunately, due to the turntable being rickety the calibration of these data sets was unsuccessful. As the turntable rotated the calibration object gradually moved out of position resulting in the incorrect estimation of the camera parameters.

### The Ceramic Cat Data Sets

The data sets of a small ceramic cat were also obtained using the automated turntable. These views, however, were calibrated using silhouette consistency constraints. Since the images were captured under circular motion an initial estimate of the camera poses can be easily established. This estimate is then optimised by minimising the reprojection errors between pairs of images thus enforcing the silhouette consistency constraint. The method is derived from the concepts discussed in [12]. A similar approach to camera calibration has been proposed by Wong and Cipolla [40].

This calibration technique could not be used on the coffee cup data sets because, due to the partial symmetry of the cup, a number of the silhouettes were very similar. These similarities resulted in the incorrect adjustment of the pose parameters of the viewpoints concerned.

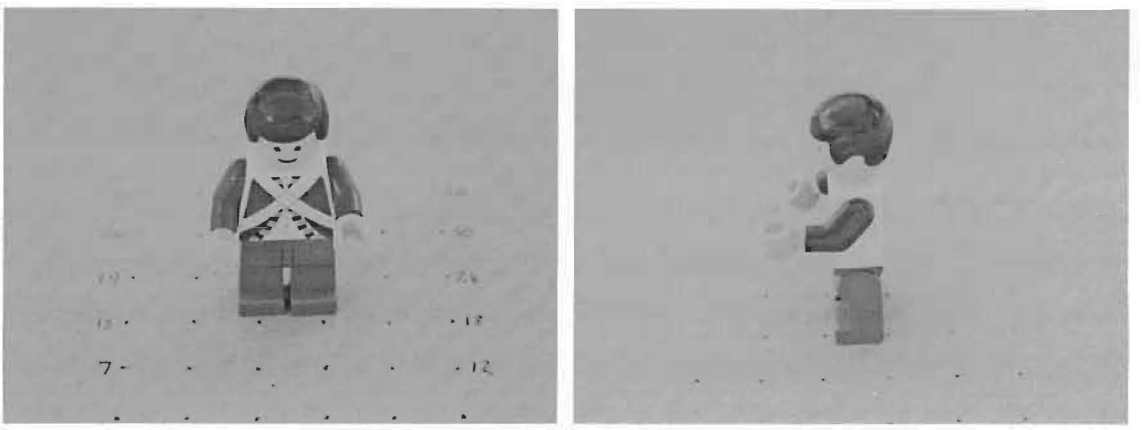
### The Toy Figurine Data Sets

The data sets of a toy figurine were acquired by positioning a digital camera at different viewpoints around the figurine. These views were calibrated using the technique proposed by Tsai [37]. Thirty-six points were arranged in a grid pattern at the base of the figurine on the ground plane. For each view fifteen image points corresponding to fifteen of the world points were selected. These corresponding points were then processed using an implementation of Tsai's method to obtain the calibration parameters. Figure 1.5 gives two examples of the images that were acquired of the toy figurine. The grid of points used to calibrate the views is visible in both images.

### The Radio Data Sets

This sequence of data sets was generated by three dimensional modelling software[2] on a computer. It consists of multiple viewpoints centred around a model of a radio. The calibration parameters for each of these viewpoints were obtained by transforming the respective camera parameters listed in the modelling software.

[2]Hash Animation Master 99 (http://www.hash.com)



(a) View one (b) View two

Figure 1.5: Two views from the toy figurine data set.

### 1.2.2 Generating the Novel Views

A novel view is specified by providing the full set of calibration parameters for the virtual viewpoint. During the evaluation of the view synthesis methods these parameters were obtained by calibrating additional views that were not used in the synthesis process. The output of the implementations was then compared to these extra views.

Alternatively, the new viewpoint could be specified by interpolating the calibration parameters of two adjacent reference viewpoints. The new parameters could further be manipulated using basic transforms in order to refine the view. This method is used to specify the virtual viewpoint when working with real data sets. However, since a reference image of the desired view will not be available this method of specification was not used during the evaluation stage.

In the case of the computer generated data sets the new viewpoints could also be specified by supplying the coordinates of the virtual camera and the coordinates of the point in the world at which it must aim. This form of specification is only made possible by the fact that the coordinates of the model radio are known.

## 1.3 Thesis Outline

Chapter 2 discusses the fundamental theory behind the two approaches to view synthesis that were implemented, as well as, many of the other techniques described in this work. Topics include a description of the *camera model* and *multiview geometry* followed by an introduction to the concept of *visual hulls*.

Chapter 3 presents a review of several virtual view synthesis techniques. Two basic groupings are identified into which the different approaches can be classified. Theoretical concepts unique to each grouping are also introduced in their respective sections. These include the idea of volumetric modelling and the differences between the forward and backward mapping of image points when rendering the novel view.

The focus then moves to the two techniques that were implemented, described in chapters 4 and 5. Each method is broken down into different stages which are individually discussed.

The results are discussed in chapter 6. The chapter begins by establishing a measure of performance and introduces the experiments that were conducted. The quantitative results obtained from the evaluation of the implemented approaches are then presented, along with some significant observations.

# Chapter 2

# Theoretical Principles

This chapter reviews the fundamental theory of computer vision upon which the investigated view synthesis techniques are built. It begins by describing the camera model that relates points in the observed world to their corresponding points in the image. This is followed by a discussion of multiview geometry, focussing on epipolar geometry in particular. Finally, the concept of visual hulls is discussed, which is the basis for both the volumetric reconstruction and shading technique explored in chapter 4, and the image-based rendering technique investigated in chapter 5.

## 2.1 The Camera Model

The camera model describes the relationship between the camera, the observed scene, and the image that is viewed. A model that closely approximates the imaging process of a real camera is the *perspective projection* model (also called the *pinhole* model) [13, 36]. A diagram depicting this model is shown in figure 2.1. A number of other projection models appear in the reference texts but they are not discussed here as they are not in the scope of this work.

The two primary components that make up the model are the *centre of projection* ($C$) and the image plane ($\pi$). The *optical axis* is defined as the line passing through the centre of projection that is perpendicular to the image plane (line $Cp$). The point at which the optical axis intersects

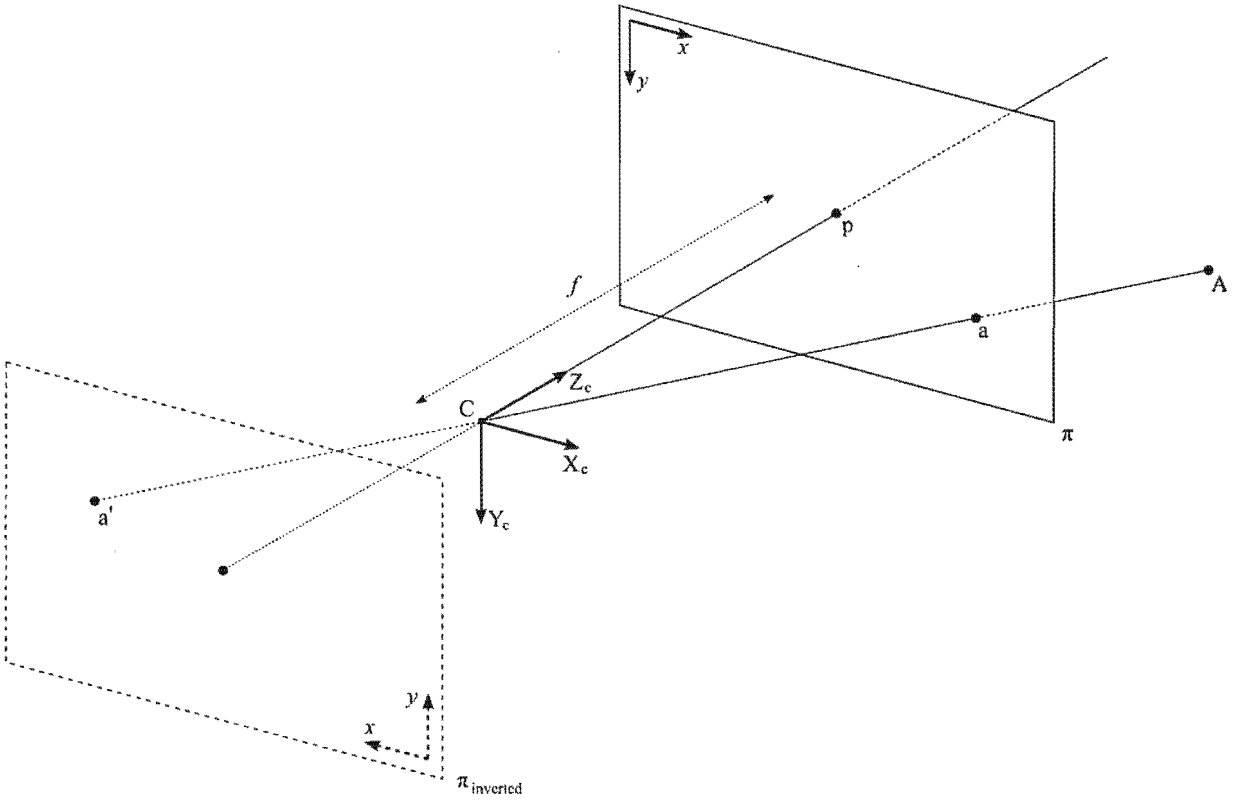



Figure 2.1: The perspective projection camera model. The images formed by perspective projection are inverted as the image plane ($\pi_{inverted}$) is found behind the centre of projection ($C$) in relation to the observed scene ($A$). An equivalent, more comprehensible representation is to show the image plane ($\pi$) in front of the centre of projection—the image in this instance will be the same as what is viewed by the camera.

the image plane is called the *principal point* ($p$), and the distance between the centre of projection and this point is referred to as the *focal length* ($f$) [13, 36, 9].

In order to derive equations a coordinate system needs to be assigned. The camera's reference frame is defined as having its origin at the centre of projection with its positive z-axis passing through the principal point of the image plane.

Using this model a mapping can be established that relates the observed scene points to their corresponding image points. This mapping is determined by the camera's calibration parameters which are divided into two groups, namely the *intrinsic calibration parameters* and the *extrinsic calibration parameters*.

### 2.1.1 Intrinsic Parameters

The intrinsic parameters are associated with the internal geometric properties of a camera [13, 36, 9]. The first parameter of note, the focal length ($f$), governs the perspective projection of the camera. Given a point $A$ in the camera's reference frame, the equations of projection can be formulated as follows:

$$\begin{aligned} x_p &= f\frac{X_c}{Z_c} \\ y_p &= f\frac{Y_c}{Z_c} \end{aligned} \tag{2.1}$$

where point $A = [\ X_c \quad Y_c \quad Z_c\ ]^\top$ and $[\ x_p \quad y_p \quad f\ ]^\top$ is the vector of coordinates of the corresponding point on the image plane (located at $z = f$).

The actual image captured by the camera corresponds to the image plane in the camera model. Since points in the actual image are expressed in pixel coordinates it can be viewed as having a reference frame of its own, separate from that of the camera's reference frame [13]. The transformation between the points of the image plane, expressed in camera coordinates, and the points of the actual image, expressed in pixel units, is determined by the size and shape of the camera pixels and the offset of the principal point from the origin of the image's reference frame. With this in mind the following can be written for mapping camera coordinates to image coordinates:

$$
\begin{aligned}
x &= \alpha\frac{X_c}{Z_c} - \alpha\cot\theta\frac{Y_c}{Z_c} + c_x \\
y &= \frac{\beta}{\sin\theta}\frac{Y_c}{Z_c} + c_y
\end{aligned}
\tag{2.2}
$$

which can further be expressed as:

$$
\begin{bmatrix} x \\ y \\ w \end{bmatrix} = K \begin{bmatrix} \frac{X_c}{Z_c} \\ \frac{Y_c}{Z_c} \\ 1 \end{bmatrix}, \qquad \text{where} \qquad K = \begin{bmatrix} \alpha & -\alpha\cot\theta & c_x \\ 0 & \frac{\beta}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.3}
$$

Here $[\ x \quad y \quad w\ ]^{\mathsf{T}}$ is the image point in homogeneous coordinates, $\alpha = \frac{f}{s_x}$, and $\beta = \frac{f}{s_y}$ with $s_x$ and $s_y$ being the horizontal and vertical size of a pixel (in millimetres) respectively [13]. The parameter $\theta$ specifies the skew of the pixels while $c_x$ and $c_y$ represent the pixel offset of the principal point. The $3 \times 3$ matrix $K$ is called the *intrinsic matrix.*

An optional parameter that can be modelled is that of *radial distortion* [36]. This type of image distortion is introduced by the optics of the camera, but can be ignored if a high degree of precision is not essential. Radial distortion results in the displacement of pixels from their actual position with the most noticeable effects being near the edge of the image. This displacement is modelled by the following equations:

$$
\begin{aligned}
x &= x_d(1 + k_1 r^2 + k_2 r^4) \\
y &= y_d(1 + k_1 r^2 + k_2 r^4)
\end{aligned}
$$

where

$$
r^2 = x_d^2 + y_d^2
$$

with $x_d$ and $y_d$ being the coordinates of the distorted points. The new intrinsic parameters are $k_1$ and $k_2$, which specify the amount of distortion. Since $k_2$ is always much smaller than $k_1$ it is usually ignored.

In summary, the intrinsic parameters of a camera are the focal length, the horizontal and vertical size of each pixel, the skew angle of each pixel, the offset of the principal point, and, optionally, the radial distortion coefficients.

### 2.1.2 Extrinsic Parameters

The extrinsic parameters describe the transformation of point coordinates between the world reference frame and the camera's reference frame [13, 36, 9]. This transformation is completely specified by a $3 \times 3$ orthogonal rotation matrix ($R$) and a three dimensional translation vector ($T$). The rotation matrix is determined by the camera's orientation relative to the world's reference frame while the translation vector is given by the camera's displacement from the world's origin (as shown in figure 2.2).

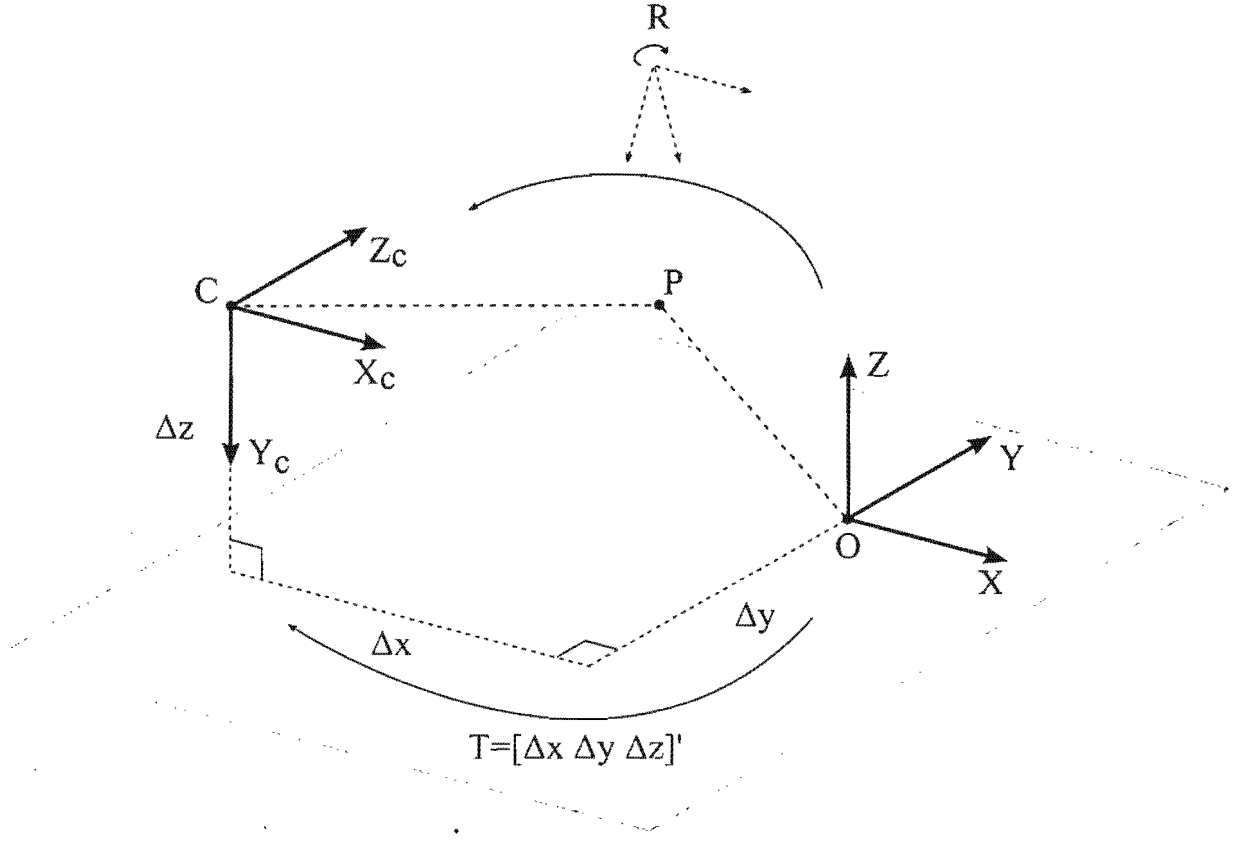



Figure 2.2: The extrinsic camera parameters. The camera's pose relative to the world's reference frame and origin ($O$) is specified by a rotation (R) and a translation vector (T).

Formulating the equations in terms of the *camera's* reference frame produces the following:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} R^\mathsf{T} & -R^\mathsf{T}T \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.4}$$

where $[\ X \quad Y \quad Z\ ]^\mathsf{T}$ is the point in the world's reference frame, and $[\ X_c \quad Y_c \quad Z_c\ ]^\mathsf{T}$ is the corresponding coordinate in the camera's reference frame. The $3 \times 4$ matrix $[\ R^\mathsf{T} \quad -R^\mathsf{T}T\ ]$ is referred to as the *extrinsic parameter matrix.*

### 2.1.3 The Projection Matrix

The *perspective projection matrix* maps three dimensional points in the world's reference frame to two dimensional pixel points in the image's reference frame [13]. It is formed by combining the intrinsic and extrinsic parameter matrices into a single expression. Using homogeneous coordinates the following can be formulated:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} \alpha & -\alpha\cot\theta & c_x \\ 0 & \frac{\beta}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R^\mathsf{T} & -R^\mathsf{T}T \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.5}$$

which can be written more compactly as

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.6}$$

where $P$ represents the $3 \times 4$ perspective projection matrix.

It is customary to fix the image plane at $z = 1$ in the camera's reference frame, thereby creating a normalized image plane [13]. Since the equations are formulated using homogeneous coordinates the actual coordinate of the image point is $[\ {}^{x}/_{w} \quad {}^{y}/_{w} \quad 1\ ]^\mathsf{T}$. A useful observation is that the term $w$ is in fact the z-coordinate, in the camera's reference frame, of the point in the world. In other words, if the coordinates of the world point were mapped to camera coordinates using

equation (2.4) then $Z_c$ would be equal to $w$. With this in mind equation (2.6) can be rewritten as:

$$\begin{bmatrix} x_i Z_c \\ y_i Z_c \\ Z_c \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.7}$$

where $[\ x_i \quad y_i\ ]^\mathsf{T}$ is the coordinates of the image point, $[\ X \quad Y \quad Z\ ]^\mathsf{T}$ is the world coordinates of the observed point, and $Z_c$ is the distance, perpendicular to the image plane, from the camera's centre of projection to the point in the world. This value could be used if a depth buffer or z-buffer needed to be implemented.

## 2.2 Multiview Geometry

The relationship that exists between multiple views of the same scene places geometric constraints on the inferred structure of the observed objects. This section discusses the formulation of these constraints and how they can be used to find point correspondences between images.

### 2.2.1 Epipolar Geometry

Epipolar geometry refers to the constraints that exist between *two* camera views of the same scene [36, 27]. If two cameras view the same three dimensional point then the corresponding image point in the second view is constrained to lie on a single line called the *epipolar line*. In order to find a single point correspondence in the second view only the epipolar line needs to be searched, as opposed to the entire image. This constraint, which maps an image point in the first view to a line in the second view, is known as the *epipolar constraint*.

The epipolar geometry between two cameras is shown in figure 2.3. The two camera centres ($C_1$, $C_2$) and the observed point ($P$) in the world form a plane called the *epipolar plane*. This plane intersects the image plane ($\pi_2$) of the second camera forming the epipolar line $l_2$. The projection of the first camera's optical centre ($C_1$) onto the second camera's image plane ($\pi_2$) is known as its *epipole* ($e_2$).

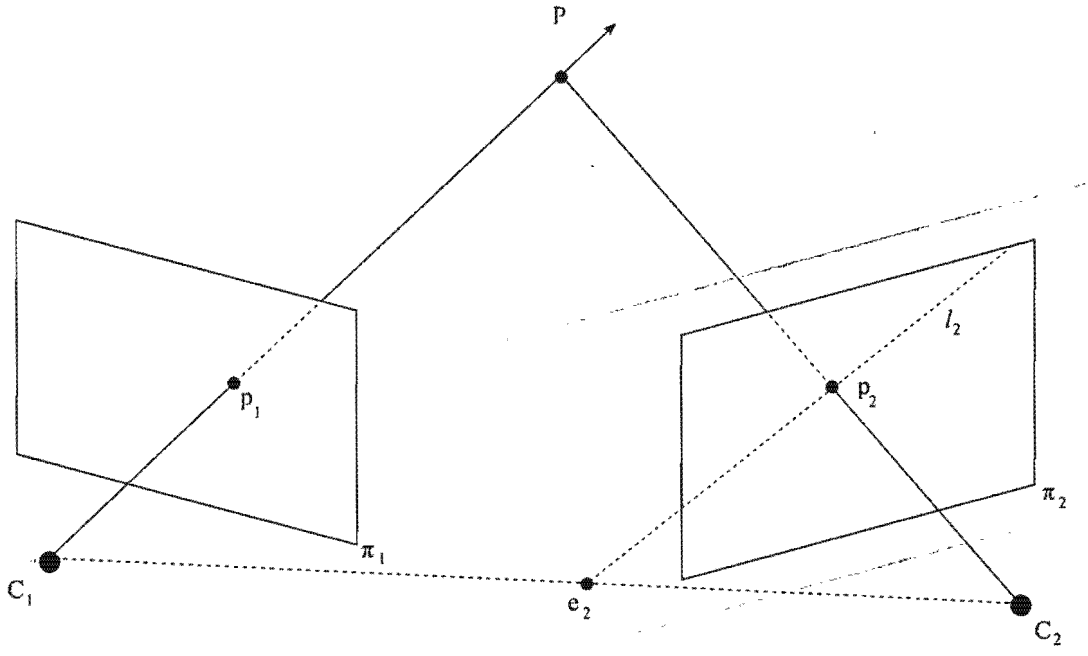



Figure 2.3: Epipolar geometry between two views. The camera centres $C_1$ and $C_2$ along with the observed point $P$ form a plane called the epipolar plane. This plane intersects the image plane ($\pi_2$) of the second camera forming the epipolar line $l_2$.

### The Essential Matrix

The essential matrix defines a mapping between points on the image plane of one camera and epipolar lines on the image plane of a second camera. This mapping is derived from the extrinsic parameters of the two cameras and thus the relative calibration of the system needs to be known [36]. This also means that the mapping is expressed in terms of the reference frames of the cameras and so in order to work with pixel coordinates the intrinsic parameters would need to be utilised. Mathematically the relationship between the two cameras is written as follows:

$$p_2^{\mathsf{T}} E p_1 = 0 \tag{2.8}$$

with

$$E = R \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \tag{2.9}$$

where $p_1$ is the projection of a scene point onto the image plane of the first camera, $p_2$ is the corresponding projection onto the image plane of the second camera, and $R$ and $T$ represent the

relative transformation between the reference frames of the two cameras. The matrix $E$ is called the *essential matrix.*

The term $Ep_1$ represents the epipolar line in the image plane of the second camera [36]. The resultant vector, $[\ a \quad b \quad c\ ]^\mathsf{T}$, gives the parameters of the line equation $ax + by + c = 0$.

#### The Fundamental Matrix

The fundamental matrix is similar to the essential matrix except that it is defined in terms of pixel coordinates as opposed to camera coordinates [36]. It can be estimated from a number of point correspondences between the two images without any knowledge of the intrinsic or extrinsic parameters of the cameras. The relationship between the fundamental matrix and the essential matrix is given by

$$F = K_2^{-\mathsf{T}} E K_1^{-1} \tag{2.10}$$

where $F$ is the fundamental matrix, $E$ is the essential matrix, and $K_1$ and $K_2$ are the intrinsic parameter matrices of the first and second cameras respectively. Substituting into equation (2.8) gives

$$\tilde{p}_2^\mathsf{T} F \tilde{p}_1 = 0 \tag{2.11}$$

where $\tilde{p}_1$ and $\tilde{p}_2$ are corresponding points, expressed in pixel coordinates, in the image of the first and second cameras respectively. As before, $F\tilde{p}_1$ represents the epipolar line in the image of the second camera.

### 2.2.2 Trifocal Geometry

Trifocal geometry refers to the geometric constraints that exist between three views of the same scene [9]. The epipolar geometry, as described in the previous section, between each pair of images can be used to predict the corresponding points in the remaining image. These points will actually be found at the intersection of their associated epipolar lines [9, 27]. This relationship, however, can be degenerate in certain instances—if the observed three dimensional point is located on the plane formed by the projection centres of the cameras (called the *trifocal plane*) then the correspondence will be undetermined.

This problem is overcome by the fact that having three views places additional constraints on the system. Whereas epipolar geometry is concerned with point correspondences, three views allows for the use of *line* correspondences between images [9]. This concept is illustrated in figure 2.4. A line ($l_1$) in the image plane of a camera defines a plane ($\pi_1$) in three dimensional space that passes through the camera centre ($C_1$). Similarly, a line ($l_2$) in the image plane of a second camera will define a second three dimensional plane ($\pi_2$). The intersection of these two planes will form a line ($L$) in three dimensional space. Projecting this line onto the image plane of a third camera will produce a two dimensional line ($l_3$) that is in correspondence with the two lines ($l_1$ and $l_2$) in the first two images [27, 9].

A point on the projected line ($l_3$) in the third image defines a second three dimensional line ($L'$) passing through that particular image point and the centre of the third camera. The two lines ($L$ and $L'$) in three dimensional space induce a constraint on the three cameras called the *trifocal constraint*. It is represented algebraically by a *trilinear tensor*. The properties of trifocal geometry and the formulation of the tensor are discussed in detail in [9, 13].

## 2.3 Visual Hulls

The closest geometric approximation of an object that can be reconstructed using only its silhouette images is referred to as its *visual hull* [19]. The visual hull can therefore be viewed as the largest shape (in terms of volume) that can be substituted for the original object while still producing the same silhouettes. Obtaining the visual hull is accomplished through the technique known as *volume intersection* [19, 33].

Given a number of views of an object, the silhouettes are usually obtained by segmenting the input images into binary images. A pixel marked as part of the silhouette indicates that its associated line of sight, or visual ray, from the camera centre meets the observed object [8]. All the intersecting visual rays for a particular image form a visual cone, and the intersection of the individual cones from all the input images gives the approximate visual hull (see figure 2.5).

It is only an approximation because the actual visual hull is described by Laurentini [19] to be the intersection of the cones corresponding to silhouettes obtained from *all* possible viewpoints

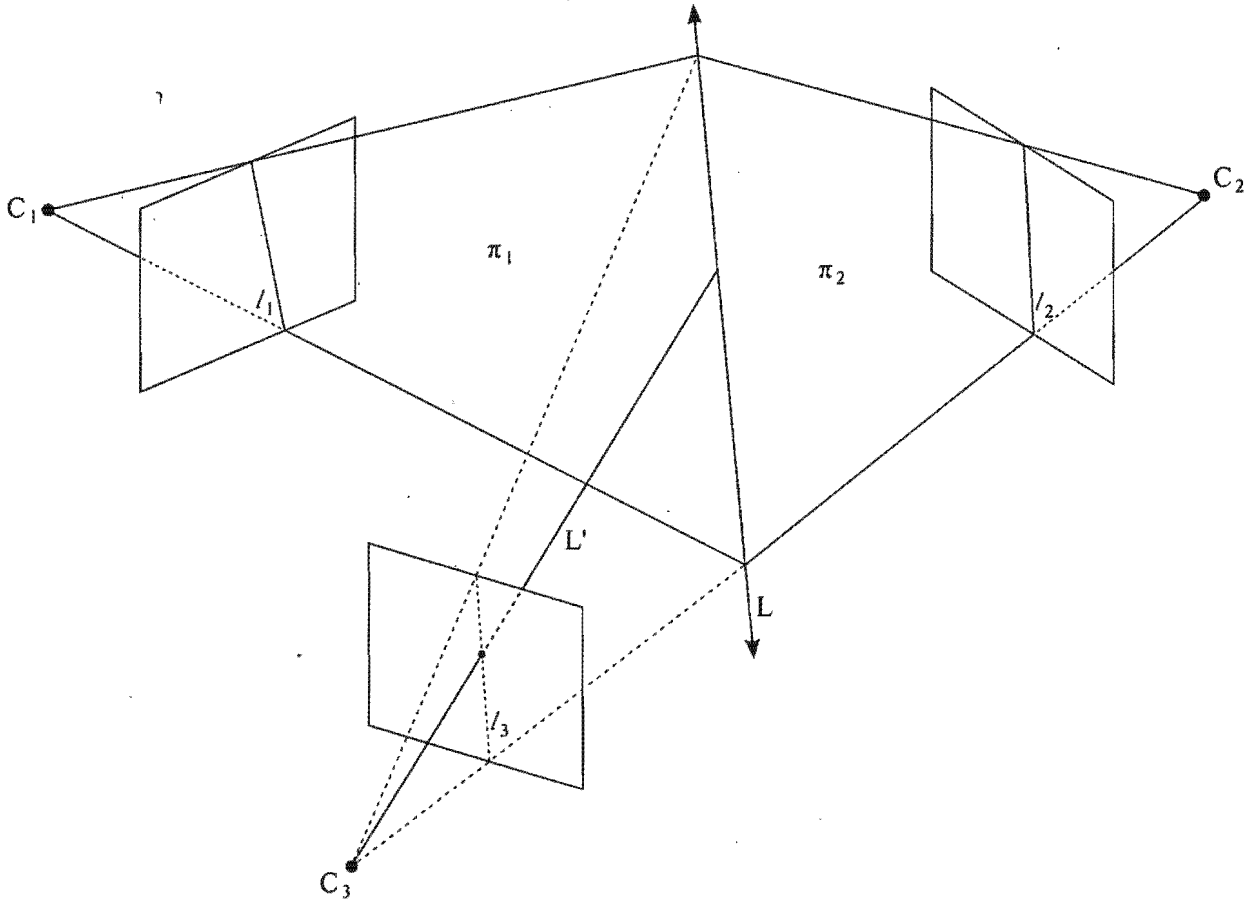



Figure 2.4: The trifocal constraint. Having three images allows for the use of line correspondences to get a constraint on the cameras.

exterior to the object's convex hull[1]. Increasing the number of input images will thus improve the accuracy of this approximation.

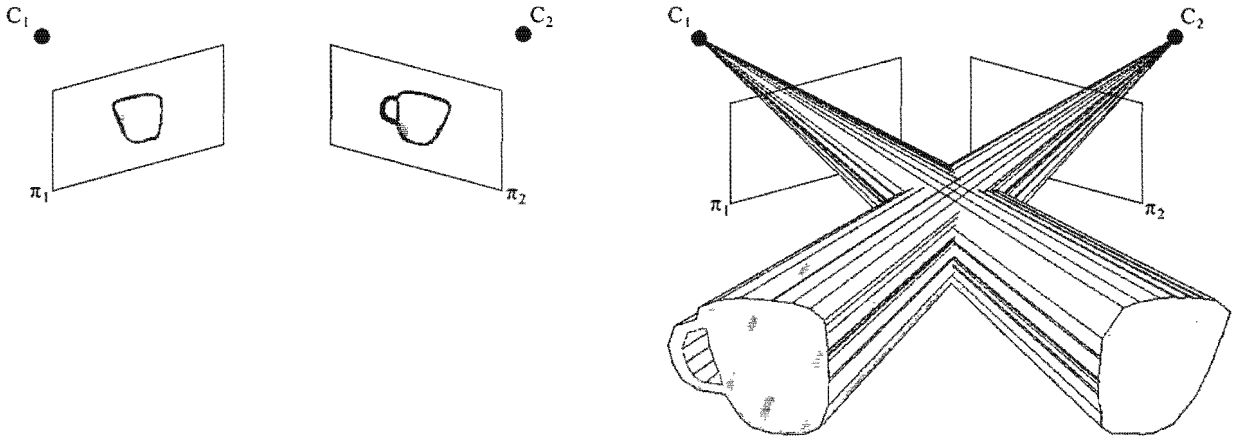



Figure 2.5: The intersection of the viewing cones defined by an object's silhouettes gives its approximate visual hull.

Since concavities are not visible in the object's silhouettes they are not modelled. The visual hull therefore encloses the true volume of the object.

[1]Laurentini also introduces the idea of an *internal* visual hull. He does, however, point out that the principal case is the approximation of the visual hull from viewpoints exterior to the object's convex hull.

# Chapter 3

# A Review of Virtual View Synthesis Techniques

Through the years a number of techniques have been suggested for the synthesis of a virtual view of a scene, or an object, from a set of reference views. These originate from at least three different fields of research including computer vision, photogrammetry, and computer graphics. The methods employed in rendering the new view fall into two general categories, namely *geometry-based rendering techniques* and *image-based rendering techniques* [25].

## 3.1 Geometry-based Rendering Techniques

Geometry-based systems make use of geometric descriptions of the surfaces of objects or volumetric data to model a scene and then render novel views [25]. Such systems first have to generate the geometric model using the reference images. The computational cost of producing the new view is thus *dependent* upon the complexity of the scene [18].

Constructing an approximate geometric model of a scene, or object, from a set of reference images can be done using image matching techniques. Correspondences between the images are computed and the three dimensional structure is then recovered via triangulation [13, 8]. An alternate approach is that of *volumetric scene modelling*. The basic principle behind this

approach is that volumes that are consistent with the given reference images are constructed in three dimensional space, thus reconstructing the scene. These volumes of space that are occupied by an object in the world can be represented as a regular tessellation of cubes, which are called *voxels* [8]. The scene space is thus divided into discrete units, with each unit being classified as either being part of the object or not. To facilitate rendering, the quantised model can be converted into a polygonal surface representation using the *Marching Cubes* algorithm [21].

Two common categories of voxel-based reconstruction algorithms can be identified [8, 33]. The first class includes those algorithms that make use of volume intersection to recover the approximate *visual hull* of the photographed object (as was discussed in section 2.3). In general, the calibration parameters need to be known, or estimated, in order to determine this volume of intersection. Deciding on which voxels form part of the visual hull can be done either in the scene space, by computing the three dimensional intersection of the silhouette cones, or in the image space, by analysing the two dimensional projections of the voxels in relation to the silhouettes [8]. To efficiently construct the voxel model an *octree* data structure can be used for computation and storage (see chapter 4 for more details).

In a recent paper [40], Wong and Cipolla construct a model of an object using uncalibrated images captured under circular motion. They estimate the calibration parameters of the cameras using the correspondences obtained from the epipolar tangents to the object's silhouettes, and the constraint on the camera motion. An initial model is then created using an octree reconstruction technique. This model is further refined by adding new silhouettes, captured from arbitrary viewpoints, to the original silhouettes. Certain parts of the model are not visible under the circular motion constraint, but by adding images captured from arbitrary viewpoints the hidden information can be included in the reconstruction process.

The second class of algorithms performs a *colour consistency* test to distinguish between voxels that are part of the scene objects and those that are not [8, 33]. When a point that lies on the surface of a photographed object is projected into the images from which it is visible, the corresponding projections will occupy areas similar in colour. Similarly, if the voxel projections in the reference images occupy areas of similar colour they can be considered as part of the surface model. This concept is illustrated in figure 3.1.

Generally these methods start with a tessellation of voxels, forming a large cube, that encloses the object with each voxel being initially labelled as part of the model. A colour consistency test

is then performed on the voxels—failing the test means that the voxel is not part of the model and must be removed, or labelled as transparent. The model is thus refined until no more voxels can be removed because all of its surface voxels are colour consistent.

If, however, the voxel being checked is occluded from some of the images the consistency test will fail even though the voxel should be part of the model. An important task for these algorithms is, therefore, to determine the visibility of the voxels from each of the reference images.

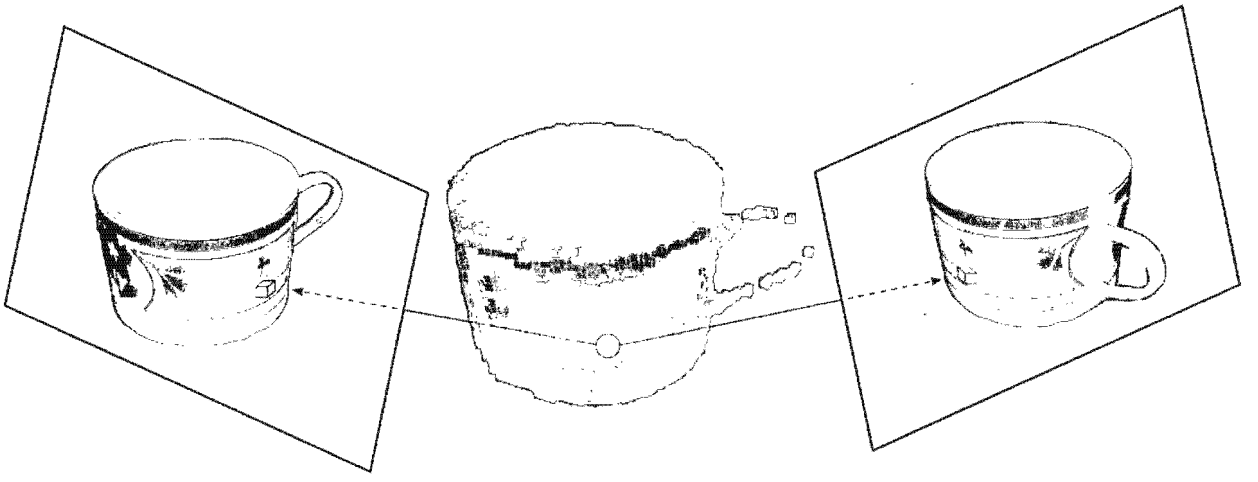


Figure 3.1: A model of an object can be constructed by searching for voxels that project to areas of similar colour in the reference images. Voxels that fail this consistency test are removed from the model.

Seitz and Dyer [30] solve this visibility problem, yet still process the voxels in a single pass, by placing a constraint on the location of the cameras. The voxels are initially divided into layers at increasing distance from the cameras. They are then processed layer by layer starting with the one closest to the cameras. Upon inspection of a particular voxel it is guaranteed that all voxels which might influence the visibility of that voxel have already been processed. It therefore does not have to be reprocessed if any of the remaining voxels change state, allowing for a single pass algorithm.

The *Generalised Voxel Colouring* algorithm developed by Culbertson and Malzbender [5] allows for the unrestricted placement of cameras while still accounting for the visibility of the voxels. They present two versions of their algorithm—both produce colour consistent models of the same quality with the differences being in the efficiency of computation and the amount of memory utilized.

Both versions begin by assigning each voxel a unique ID. The first version then renders each voxel into the reference images, but instead of storing colour information for each pixel that is part of the voxel's projections it stores the voxel's ID. As it renders it implements a form of *z-buffering*. The ID stored with a particular pixel thus indicates that the associated voxel is the closest surface voxel[1] that a visual ray through that particular pixel will intersect. The voxel is, therefore, visible from the camera.

Once the visibility of the voxels is known the colour consistency test can be performed. If a voxel is carved the visibility of certain other voxels might change and therefore the visibility information needs to be updated. Since recomputing the visibility is time consuming it is only done periodically. As a result, the carving of voxels is not efficient because voxels that would normally be carved if the visibility information were up to date are constantly rechecked for colour consistency until the next update.

The second version of their algorithm makes use of a *layered depth image* to overcome the inefficiency of the first. For each pixel the ID of every surface voxel that projects onto it is stored as opposed to only the ID of the closest voxel. These ID's are stored in depth order. If a surface voxel is carved then the adjacent interior voxels become surface voxels and the ID lists of the affected pixels are updated. The algorithm therefore knows which voxels have experienced a change in visibility and only these voxels need to be checked for colour consistency. Hence, this version does not perform consistency checks that are unnecessary and is thus more efficient. It does, however, use more memory for the associated data structures.

The second aspect to reconstructing voxel models using colour consistency is the measure of similarity. A common approach involves thresholding the standard deviation between the colours associated with the pixels of the voxel projections [33]. Other methods have also been suggested, including one that uses the intersection of histograms as a test for consistency [34].

[1] A *surface voxel* is a voxel that has at least one neighbouring voxel that is transparent. The algorithm maintains a separate list of all the surface voxels.

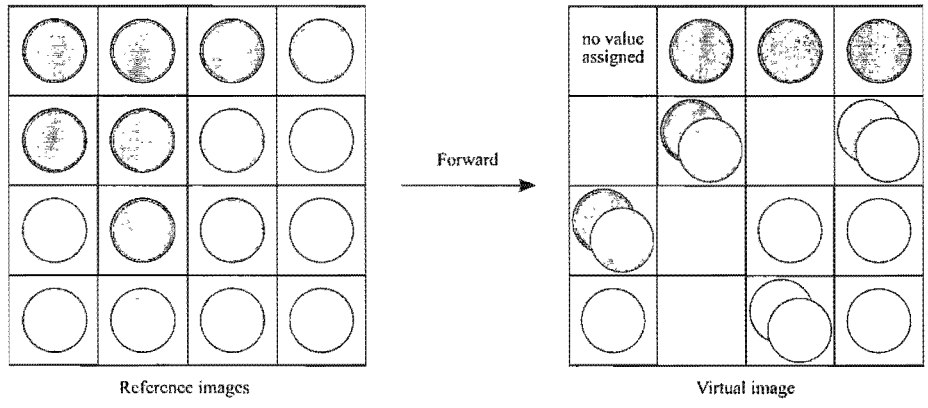
## 3.2 Image-based Rendering Techniques

Image-based rendering systems generate a virtual view of a scene directly from the photometric data contained in the reference views. This is accomplished through the use of techniques such as view interpolation or pixel-reprojection between the reference images and the virtual image. Since they do not reconstruct a geometric model of the scene, the computational cost of producing the novel view is *independent* of the scene's complexity [25, 18]. There is also the potential for photo-realistic renderings if images of real scenes or objects are used for the reference views.

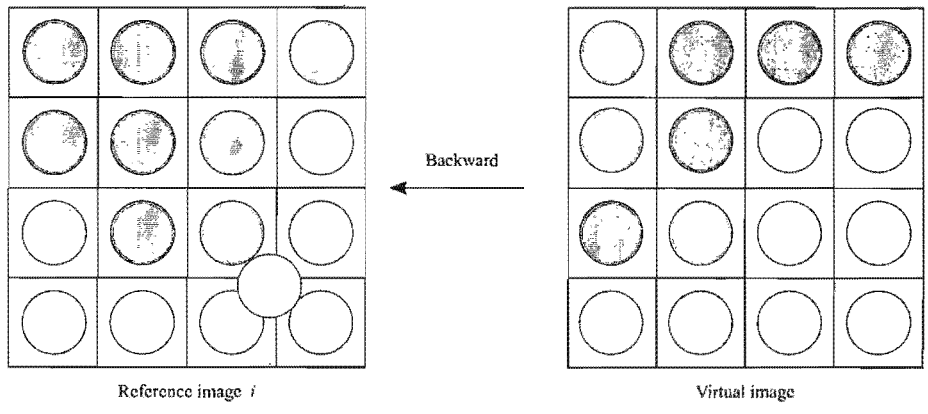
Classification of the various methods into distinct categories is not straightforward. A previous survey [32] prefers to view the different approaches as a "continuum" of image-based rendering techniques ranging from those that make use of *no* geometrical information to those that make use of *implicit* or *explicit* geometrical information.

Many image-based approaches, particularly those that make use of pixel reprojection, need to relate points in the reference images to points in the virtual image. This can be performed through either a *forward* or *backward* mapping [4]. In the case of a forward mapping points in the reference images are transformed to determine their respective points in the virtual image. The coordinates of these new points are then rounded to their nearest pixel coordinates. As a consequence, points may be mapped to the same pixel and holes can occur where pixels in the virtual image have not been addressed. This point is illustrated in figure 3.2(a). Due to these shortcomings it is preferable to use a backward mapping approach. Each pixel in the virtual image is mapped to corresponding points in the reference images. These new points will have subpixel coordinates and thus the new colour can be calculated by interpolating the colour values of the neighbouring pixels (figure 3.2(b)). Besides ensuring that each pixel in the virtual image is processed, the aliasing effects can also be minimised.

*Light Field rendering* [22] is an approach to image-based view synthesis that does not rely on the recovery of any geometric information. The idea is to model the *light field*, which Levoy and Hanrahan define as the radiant energy travelling along light rays through a point in three dimensional space from a specified direction. They accomplish this using a dense collection of views under the assumption that the viewing area is occlusion free. Due to the large number of reference images required (the authors mention hundreds or even thousands) the acquisition process will be time consuming and the storage requirements will be high. The paper does,




(a) Forward mapping




(b) Backward mapping

Figure 3.2: Mapping points between the reference views and the virtual view. Figure (a) depicts the forward mapping of points from the reference images to the virtual image. Figure (b) depicts the backward mapping of points from the virtual image to any of the reference images. The grid represents the pixels of the image and the circles represent the assigned colour value. Image examples of the effects of these mappings can be found in [4].

however, discuss a method to compress the captured data while still allowing for viewing in real-time.

View interpolation refers to the process of producing intermediate views of a scene from the images of two reference views [29]. It stems from the many image warping and morphing techniques that produce a sequence of images depicting the transition between a source and target image. The process usually involves establishing point correspondences between the images, followed by an interpolation of the displacement between, and colour values of, the related points.

Chen and Williams [3] use a depth map and the relative pose between cameras to easily find matching image points. Since they make use of synthetic images for their experiments this data is readily available. The first step is to determine the *dense* correspondence of the reference images obtained from two cameras with a narrow baseline. The displacement vectors between each pair of image points are then pre-computed, thereby improving the rate at which new views can be rendered. During image synthesis these displacement vectors are interpolated and the pixel values are merged to form the novel view. The process is essentially a forward mapping of pixels from the source image and, as a consequence, has to deal with overlapping pixels and the holes that form, due to pixels not being addressed, in the virtual image. The authors address both these issues and suggest ways to minimise their effects.

In general, image interpolation is not guaranteed to produce physically correct views of a scene [29, 31]. Chen and Williams, however, point out that if the transformation between the three cameras is restricted to a translation that is parallel to the image plane then the result of the interpolation will be perspectively correct. Seitz and Dyer [29], furthermore, demonstrate that by first rectifying the reference images a valid intermediate view can also be synthesised. Their approach is formulated under *orthographic* viewing conditions and does not require that the camera parameters be known. The algorithm begins by aligning the epipolar lines in the two reference images via image rectification. This requires that four corresponding points be identified in each of the images. The epipolar lines now run parallel to the x-axes of the images and a stereo matching method is used to locate corresponding "uniform intervals" of colour within each scanline[2].

In contrast to methods that try to reconstruct a scene, methods that synthesize new views through interpolation are not affected by the problems that arise from "uniform regions" of colour [29,

[2]A *scanline* refers to each row of pixels in a digital image, i.e. a line of pixels parallel to the x-axis of the image.

39]. Only the end-points of the "uniform intervals" in a particular scanline need to be matched as knowledge of the surface is not needed—errors that could arise due to ambiguous correspondences within such regions are therefore not an issue.

The points of the corresponding intervals, and their associated colour intensities, are then interpolated to produce the new view. The final step in the algorithm is the derectification of the resultant image which gives the synthesized view.

Seitz and Dyer point out that the output of the algorithm is dependent upon the *monotonicity constraint*. This constraint is related to the visibility of scene points within the two reference images. If matching points appear in the same order along corresponding epipolar lines in each of the reference images then they will be visible in all the intermediate views.

The geometric constraints (as discussed in section 2.2) that exist between multiple views of the same scene can be utilized to synthesize a new view. These techniques are similar to those used in the field of photogrammetry and are often referred to as *transfer methods* [13]. Corresponding points in the reference views are reprojected into the virtual view forming the new image.

Such an approach would be through the manipulation of the epipolar geometry that exists between pairs of images, as was investigated by Laveau and Faugeras [20]. The first step in their algorithm is to compute the dense correspondence between two reference images, and is accomplished using a stereo matching algorithm. Each point in a matching pair is represented in the virtual image by an epipolar line. The intersection of the two epipolar lines marks the reprojection of the matched points. The colour value of this new point in the virtual image can be determined by combining the colour values of the two original points in the reference images.

This form of reprojection represents a forward mapping of points and as such suffers from the aforementioned problems, namely overlapping pixels and holes. The authors, therefore, reformulate their approach to overcome these shortcomings. Essentially, a pixel in the virtual image is represented by an epipolar line in each of the reference views. A search must then be performed on these epipolar lines in order to find a correspondence [20, 4]. There are three possible outcomes to this search. The first is that no correspondence is found, meaning that the point could be occluded in one of the references images. The next possibility is that there is only one correspondence, in which case the colour values are transferred to the pixel in the virtual image. If multiple correspondences are found then the occlusion order of the matched points needs to

be determined. For a particular reference view the occlusion order is dependent upon the relative position of the virtual camera with respect to the reference camera's image plane.

The selected image point in the reference view is the projection of the surface point on the observed object which is closest to the virtual camera's centre of projection along the line of sight associated with the pixel in question.

Avidan and Shashua [1] make use of trilinear tensors to generate novel views of a scene from two or three reference images. Their method requires a dense correspondence between two reference images but they do not recover the full camera calibration parameters. The algorithm begins by calculating an initial trilinear tensor or "seed" tensor (as termed by the authors) for the reference images. When only two images are available the tensor is derived from the fundamental matrix—the third image can be seen as coinciding with one of the other two. Specification of the virtual view is accomplished through manipulation of this seed tensor thereby producing a change in the position and orientation of the third, or virtual, camera. The point correspondences between the original reference images and this new tensor are used to synthesize the new view. The actual reprojection of points is done using a backward mapping—rectangles in the reference images are first mapped to quadrilaterals in the virtual image from which the backward mapping is computed.

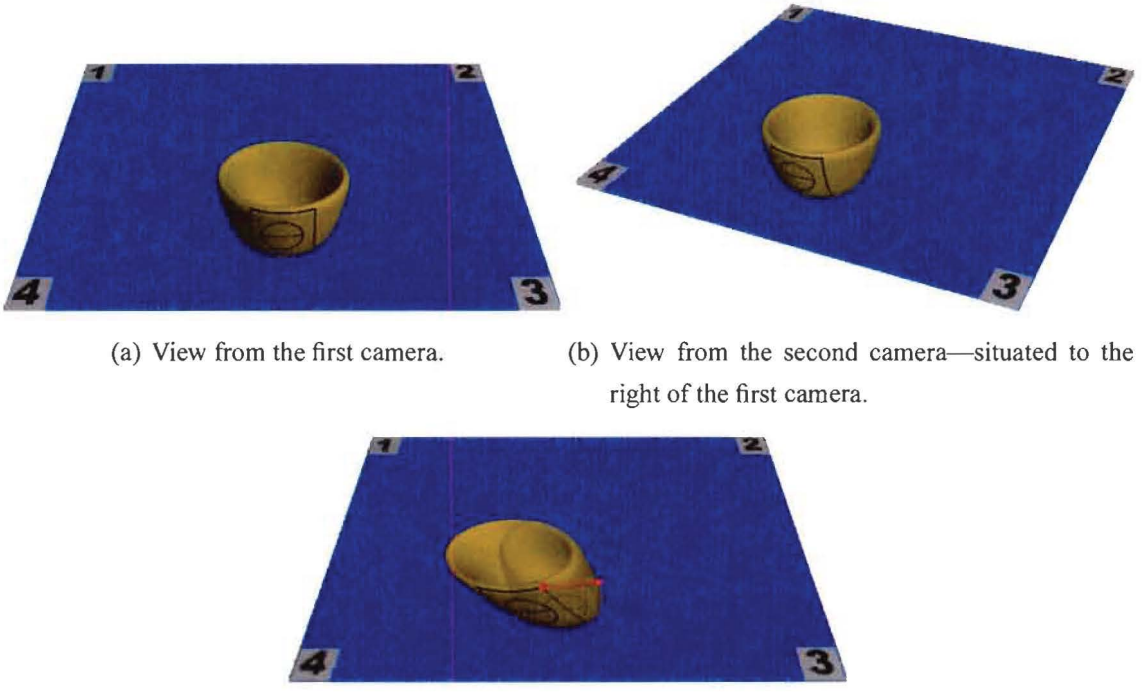
An algorithm developed by Matusik et al. [24] renders the image of a textured visual hull of an object without having to first reconstruct the geometric model—hence they call the approach *image-based visual hulls*. Avoiding the explicit construction of a geometric model is made possible by exploiting the epipolar geometry that exists between the virtual view and each of the reference views. The output will, therefore, be viewpoint-dependent as the image is generated directly.

The visual hull is computed using the silhouettes of the object and can be expressed as a depth map relative to the image plane of the virtual camera. This depth map is used to determine the appropriate texture mapping for the novel view. The technique for shading the visual hull is a *view dependent* mapping and will therefore capture lighting effects unique to any of the reference views.

The approach is essentially a backward mapping of pixels from the virtual image to the reference images and hence does not suffer from the problems associated with forward reprojections.

Further details on this method of synthesis can be found in chapter 5.

The transformation of points from one plane to another is described by a homography [9, 27]. A homography can therefore be formulated for the relationship between points on a plane in the real world and their projections on the image plane. A homography will also exist between multiple images that view the projections of points that lie on a specific plane. These projections in the first view will be correctly mapped via the homography to their corresponding projections in the second. Points that do not lie on the plane in the real world will not be mapped correctly—the displacement between the mapped point and the actual point in the second image is called the *parallax* [14, 16]. This concept is illustrated in figure 3.3.




(a) View from the first camera.

(b) View from the second camera—situated to the right of the first camera.

(c) Warped image of view from the second camera.

Figure 3.3: The image captured from the second camera (b) is transformed by the homography that exists between the image planes of the two cameras. The result is shown in (c) with the original view from the first camera overlaid—the parallax between a mapped point from the second image and the actual point in the first image is highlighted in red.

Irani, Hassner, and Anandan [17] manipulate the plane and parallax geometric formulations to render novel views that exhibit an extreme change in viewpoint between the real camera views and the virtual view. Their approach analyses the two dimensional projections of the three di-

mensional line of sight of each pixel in the virtual image. If an object intersects the line of sight of any particular pixel it will consistently appear in all the projections in the reference images. An assumption that is made is that all the cameras have an unoccluded view of the virtual line of sight.

The two dimensional projections need to first be transformed so that they share a single coordinate system. This is accomplished by warping the images using the homography that exists between themselves and the primary image (e.g. the first image in the set). This homography is induced by the reference plane that is visible in all the images. The projections in the warped images now form a *pencil of lines* with the axis point being the projection of the point where the line of sight meets the reference plane [17].

To relate the points lying on the projected line of sight in the primary image to points in one of the other images the following formula is used:

$$p_i \cong (v_i \times p_V) \times (e_i \times p_R) \tag{3.1}$$

where $p_i$ is the point in image $i$, $v_i$ is the epipole of the virtual camera in image $i$, $p_V$ is the projection of the point where the line of sight meets the reference plane, $e_i$ is the epipole of the camera corresponding to image $i$ in the primary image, and $p_R$ is the point on the projected line of sight in the primary image.

The projected line of sight in each image is now evaluated, and the first point that is consistent in colour among all of the images is what is seen along the three dimensional line of sight.

# Chapter 4

# Geometry-based Rendering: Voxel Reconstruction and Texturing

The approach to virtual view synthesis described in this chapter involves the reconstruction of the explicit three dimensional model of an observed object. This model is then rendered using scan-conversion algorithms giving the desired virtual view [10]. The approach is therefore classified as a geometry-based rendering technique.

Recovery of the object's geometrical structure is achieved via a technique known as volume intersection—intersecting the visual cones defined by the object's silhouettes gives an approximation to its visual hull as discussed in section 2.3. The volume of space occupied by the object is modelled using voxels. To facilitate the storage and creation of the voxel model a hierarchical data structure is utilized.

## 4.1 Constructing Voxel Models using Octrees

An octree is a hierarchical data structure that allows for the efficient computation and storage of voxel models [15, 10]. Instead of modelling the volume of space occupied by part of an object with eight separate voxels it is represented by a single larger voxel. An illustration depicting this concept is shown in figure 4.1. The occupied volume that is internal to the model is represented

by larger voxels replacing the eight smaller voxels that would otherwise have been used. Only the voxels near the surface of the model are smaller in order to match the shape of the object. The amount of information that needs to be stored is thus reduced as only the details of a single voxel needs to be stored as opposed to each of the smaller ones. It also improves the performance of the resulting algorithm since an initially coarse model is progressively refined, thereby reducing the number of voxel tests that need to be performed.

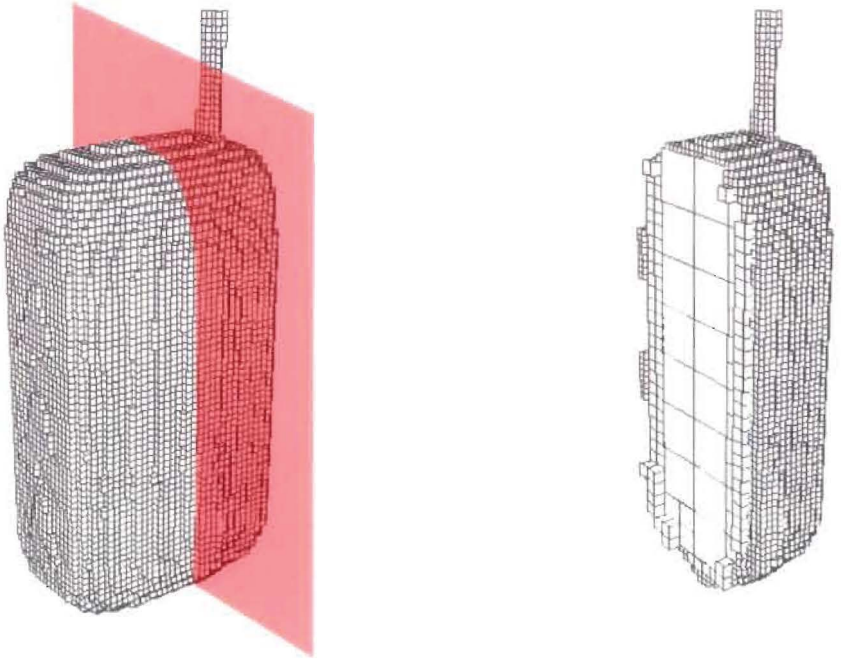


Figure 4.1: Octree representation of voxel models. The cross-section of a voxel model constructed using an octree data structure is shown. The interior voxels are larger as there is no detail to be modelled whereas the surface voxels are smaller to match the shape of the object. This reduces the storage requirements of the model and eliminates the redundant processing of interior voxels.

The volume occupied by the observed object is at first represented by a single all encompassing voxel. This voxel is then repeatedly subdivided until the model is formed. The subdivision happens as follows. The initial voxel, and every voxel that is subsequently processed, is projected into each of the reference camera views. Comparing these projections to the associated silhouette images can have one of the following outcomes [40]:

- The projections lie within the boundaries of every one of the silhouettes. The voxel is therefore recorded as being part of the model.

- One or more of the projections lie outside the boundaries of the silhouettes. The voxel is not part of the model and is therefore removed.

- The projections straddle the boundaries of the silhouettes. The voxel is sub-divided into eight smaller voxels and the process is repeated for each one.

The subdivision of voxels can continue until every voxel projects into the silhouettes. It is, however, more practical to limit the level of subdivision, thereby putting an upper bound on the resolution of the final model [40]. For example, if the limit was set to five levels of subdivision then the resolution of the output would be equivalent to using a tessellated cube of 32 by 32 by 32 voxels. This cube is synonymous with the initial voxel mentioned before. Since the object fills a finite volume of space the more voxels that are used during the reconstruction process the more accurate the model will be. Therefore, the greater the level of subdivision the more accurate the octree model will be (as shown in figure 4.2). Furthermore, if an octree were not used the modelling process would require 32768 ($32 \times 32 \times 32$) voxel projections and comparisons.

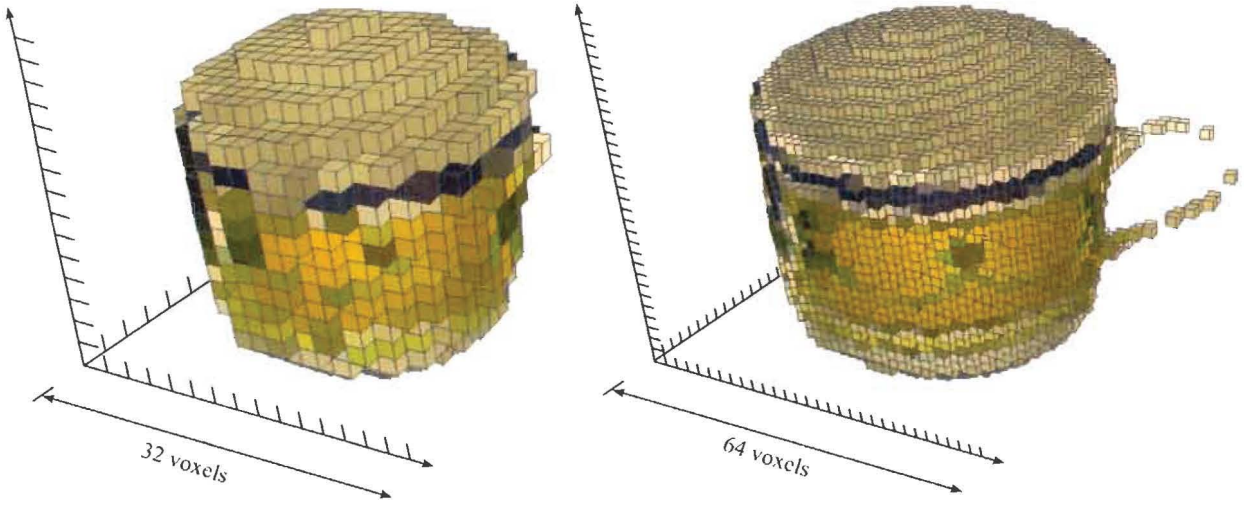



Figure 4.2: Increasing the level of voxel subdivision increases the accuracy of the octree model. Two octree reconstructions of a coffee cup are presented. The model on the left was created using five levels of subdivision which is equivalent to representing the occupied volume with a cube of 32 by 32 by 32 voxels. The model on the right was created using six levels of subdivision and is therefore more accurate. This is especially evident in the fact that the model on the left does not have a handle.

## 4.2 Adding Texture to the Voxel Models

The texture information for the voxel model is obtained from the original reference images. Depending on the observations, each surface voxel is assigned a colour. Since the interior voxels are never visible during the creation of a new view they can be ignored or even discarded—this will reduce the time needed to generate the new view. The simplest approach is to iterate through every surface voxel and, for a particular voxel, project its centre into each of the reference images, recording the colour value of the closest pixel to each projected point. These colour values can then be averaged and the resultant colour assigned to the voxel.

Averaging the colour values, as described above, is an example of *view independent* shading. Illumination effects that are unique to a particular camera view will not be reproduced in any of the synthesized virtual views. An alternative is to record the colour information obtained from each image separately. When rendering the new view only a subset of the colour values for a particular voxel is used for shading thus allowing for *view dependent* texture mapping [7, 6]. The colour that is assigned to a particular voxel is only determined at render time and is dependent upon the pose of the virtual camera in relation to the reference cameras. A similar approach proposed by Debevec [7] combines the texture information contained in the reference images according to a weighting function. The function effectively compares the viewing angle of the virtual camera to that of the reference cameras—the camera which has the closest viewing angle will have the most relevant texture information and will thus be given the highest weighting. When rendering the virtual image the colour value of a particular pixel is determined by merging the appropriate colour values from the original images according to the assigned weighting.

Shading the model by studying the projections of the voxel centres may give a good estimate, but incorporating more of the pixels neighbouring the projected point in the calculation should produce better results. An even more accurate approach is to render the entire voxel using a *scan-conversion* algorithm (as shown in figure 4.3) [10]. This leads to an exact mapping of pixels to a particular voxel allowing for a more complete colour assignment. A further advantage is that rendering can be performed using hardware, thereby reducing computation time.

Unfortunately, a problem arises with the straightforward application of the aforementioned techniques—a voxel can be occluded in a particular reference view by one or more other voxels in the model [33]. Including the projection of an occluded voxel in the colour analysis distorts

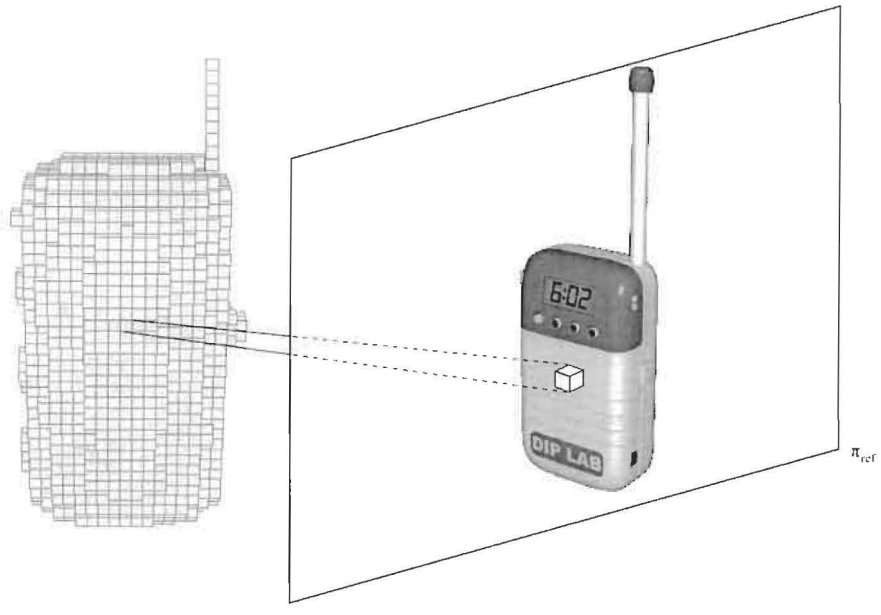



Figure 4.3: A voxel from the model is rendered onto the image plane of a reference camera using a scan conversion technique. This essentially forms a mask revealing which pixels in the image can be mapped to the voxel.

the assignment of values because the observed colours actually belong to the occluding voxels. Thus, before extracting the colour information from a particular reference view, the visibility of each voxel in that view needs to be established.

A common method for resolving the visibility issues related to model rendering is the implementation of a *z-buffer*, also known as a *depth-buffer* [10, 15]. Two separate buffers of equal size are therefore needed for the rendering process, namely the image buffer and the z-buffer. For a particular pixel in the image the corresponding entry in the z-buffer stores the distance to the observed polygon point responsible for the colour values. When a polygon is scan-converted the distance to the point being projected is compared to the z-buffer value of the current pixel in order to determine whether the new point is closer to the image plane of the camera (i.e. has a smaller z-coordinate). If this is the case then the colour and depth values are updated with the values of the new point. The polygons, and hence voxels, can thus be rendered in any order.

For a particular reference view, the visibility of the surface voxels can thus be established by first rendering them all with the aid of a z-buffer [5]. Instead of storing a colour value at each pixel the ID of the current voxel being processed is stored. The result is then an image map with the

value at each pixel identifying which surface voxel is visible along that particular line of sight.

During the colour computation for a surface voxel the algorithm first needs to scan the image map for all pixels with the correct voxel ID. The colour values of the corresponding pixels in the original reference image are then averaged to determine the contribution from the associated view. This process is repeated for each reference view with the colours being stored separately, or being combined, depending on whether or not view dependent texturing is desired.

# Chapter 5

# Image-based Rendering: Image-based Visual Hulls

A viewpoint-dependent representation of an object's visual hull can be computed without actually reconstructing an explicit geometric model. The result will take the form of a depth map relative to a particular viewpoint—each pixel in the image gives an indication of the distance to the surface point of the visual hull along that particular line of sight. These depth values can then be used to extract colour information from the reference images, thereby generating the novel view.

This chapter discusses an algorithm based on the approach to virtual view synthesis entitled *image-based visual hulls* [24].

## 5.1 Computing the Visual Hull

The computation of the observed object's visual hull is performed in the following manner: for each pixel in the virtual image the three dimensional point in the world where the pixel's line of sight meets the visual hull of the object must be calculated. The information necessary for this computation can be extracted directly from the silhouettes of the object by making use of the epipolar geometry that exists between the virtual view and each of the reference views [24].

The silhouettes are obtained by segmenting the reference images. The visual ray associated with a particular pixel in the virtual image is then projected into each of these silhouette images. For a given silhouette image the ray projection can be found using the fundamental matrix that relates points in one image to epipolar lines in another (as discussed in section 2.2.1). More precisely, the projection is given by $l = Fp$ where $l$ gives the coefficients of the line representing the ray projection in the image plane of the reference view, $F$ is the fundamental matrix, and $p$ is the homogenous pixel coordinate in the virtual image [36].

A search is now performed on the line in order to determine whether it overlaps the actual silhouette. Only the visible portion of the image plane, namely the silhouette image, is searched. Overlapping segments are projected back into three dimensional space giving the corresponding line segments along the visual ray in question. Figure 5.1 illustrates the back projection of the segments. This reprojection can be performed as follows. The line segments are each specified by two points and each point defines a three dimensional line, or visual ray, passing through that point and the reference camera's centre of projection. The intersection of these new visual rays with the original visual ray from the virtual camera is then found. Corresponding pairs of three dimensional points, marking the intersections, represent the back projections of the two dimensional line segments from the silhouette image [23].

The actual intersections are found by computing the two points for which the distance between the lines is a minimum. The line segment ($l$) linking these closest points, thus representing the minimum distance, will be perpendicular to both lines as shown in figure 5.2.

To compute the closest points the two lines are first expressed in parametric form as follows:

$$X(t) = C + t\left(P^{-1}x - C\right) \tag{5.1}$$

where $X(t)$ is a point on the line, $C$ is the camera's centre of projection, $P^{-1}$ is the inverse camera projection matrix, and $x$ is the homogeneous coordinate of the pixel in the image, respectively. The value of the parameter $t$ that will give the closest point for the line representing the ray from the virtual image can then be calculated using the following formula [35]:

$$t = \frac{(u \cdot v)(v \cdot w) - (v \cdot v)(u \cdot w)}{(u \cdot u)(v \cdot v) - (u \cdot v)^2} \tag{5.2}$$

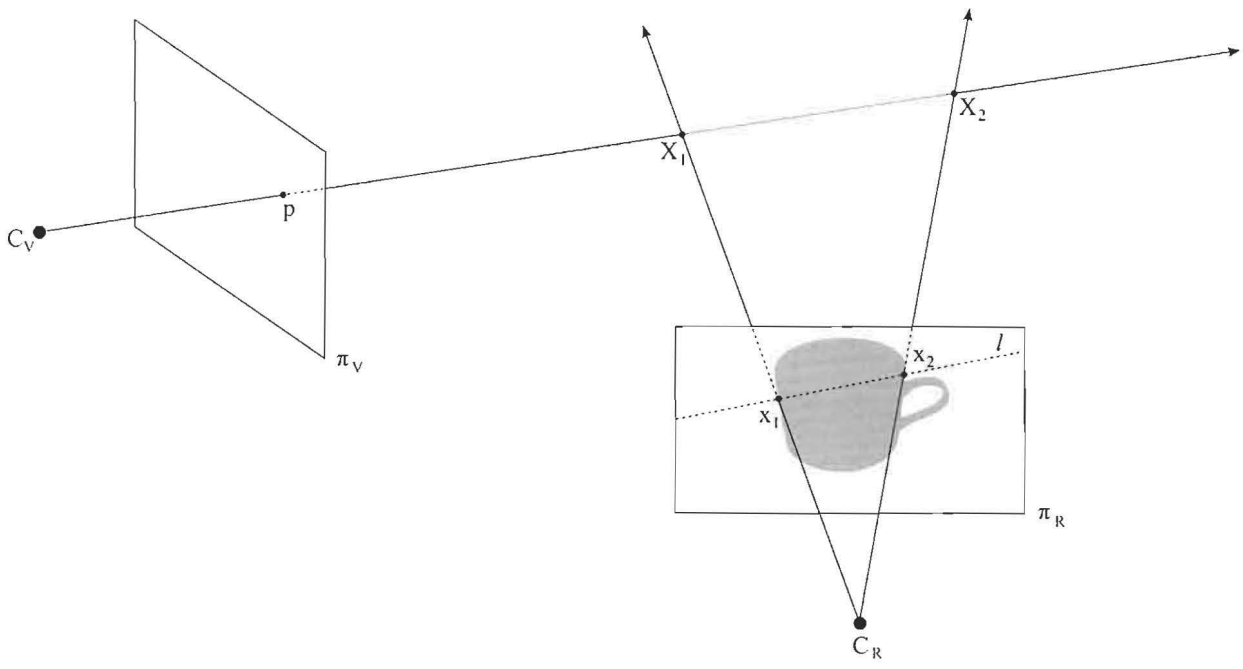



Figure 5.1: Back projecting the two dimensional silhouette intersections into three dimensional space. The points ($x_1$ and $x_2$) at which the epipolar line ($l$) intersect the silhouette image ($\pi_R$) are recorded, thereby specifying the line segment that overlaps the silhouette. These points define visual rays stemming from the reference camera ($C_R$). The three dimensional points ($X_1$ and $X_2$) where these rays meet the visual ray of the virtual camera ($C_V$) mark the back projection of the overlapping line segment ($\overrightarrow{x_1 x_2}$) in the image.

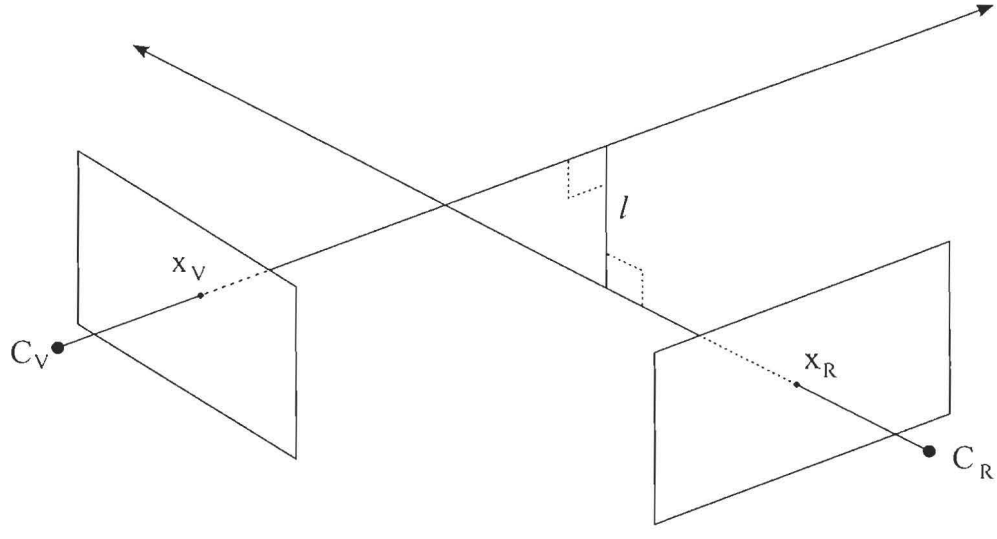



Figure 5.2: Determining where two visual rays meet by finding the closest approach.

with

$$\begin{aligned} u &= P_V^{-1} x_V - C_V \\ v &= P_R^{-1} x_R - C_R \\ w &= C_V - C_R \end{aligned}$$

where $P_V^{-1}$ and $P_R^{-1}$ are the inverse camera projection matrices of the virtual and reference camera, respectively, $x_V$ and $x_R$ are the coordinates of the image points in the virtual and reference camera, respectively, and $C_V$ and $C_R$ are the virtual and reference camera centres, respectively.

The process of finding the three dimensional line segments for a visual ray from the virtual image is repeated for each silhouette image. The intersection of each of these line segments is then calculated and the point that is closest to the image plane of the virtual camera marks the surface point of the visual hull that is visible [24].

Unfortunately, the pose of the virtual camera in relation to any one of the reference cameras can cause certain parts of the epipolar line in that reference view to be invalid. An example of such an instance is illustrated in figure 5.3. As mentioned before, only the epipolar line segment that is visible in the silhouette image needs to be searched for points overlapping the silhouette. This requirement, however, must be made even stricter in that only the visible epipolar line segment that represents the visual ray extending from the virtual camera's centre of projection should be searched [23]. For example, if a point to the left of $p_V$ on the epipolar line $l$ in figure 5.3 were back projected the calculated value for parameter $t$ will be negative, representing a world coordinate behind the virtual camera's image plane and centre of projection. The valid region of the epipolar line ($l$) that must be searched is the line segment $\overrightarrow{p_V a}$.

Determining the appropriate range of the visible epipolar line is dependent upon the position of the reference camera with respect to the virtual camera's image plane, and also on whether the line segment from the reference camera's centre of projection to the vanishing point of the visual ray from the virtual cameras (line segment $\overrightarrow{C_R P_V}$ in figure 5.3) intersects the reference camera's image plane. In this regard, approximately four separate cases can be identified—since they are only partially implemented in this work no formal discussion is presented here. More details on this topic can be found in [26] and [23]. The consequence of not taking these issues into account is shown in figure 5.4.

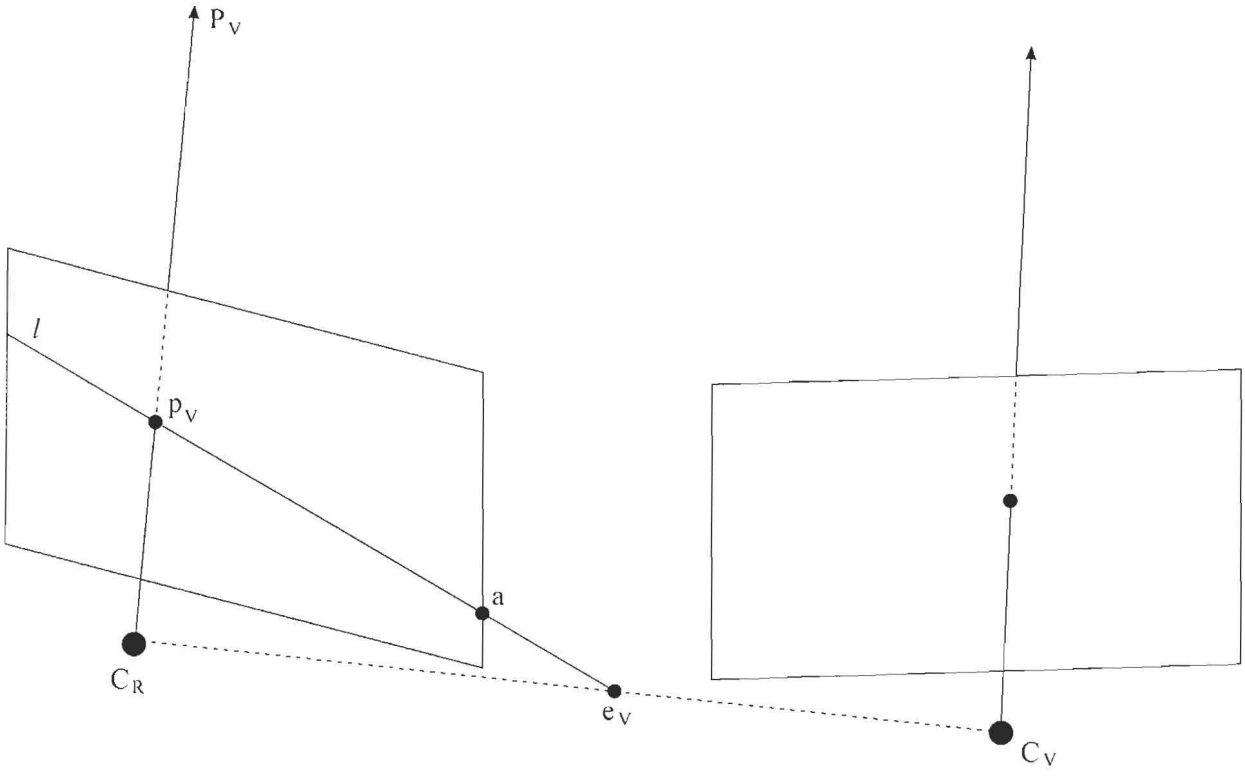


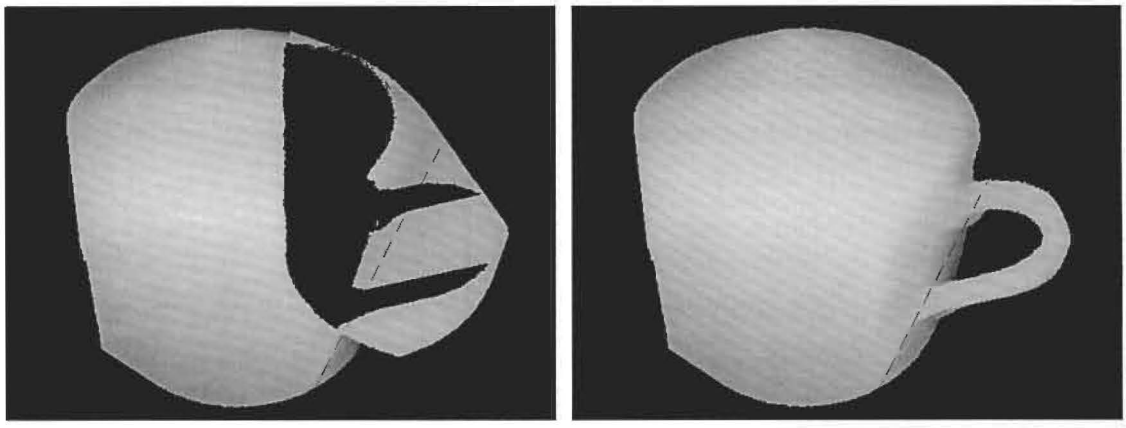

Figure 5.3: The valid region of the epipolar line ($l$) that must be searched is the line segment $\overrightarrow{p_V a}$.

## 5.2 Texture mapping the Visual Hull

A *view-dependent* texture mapping of the computed visual hull is performed by extracting colour information from the original reference images. A colour value needs to be assigned to each pixel in the depth map thus completing the synthesis of the novel view.

In practice, each pixel in the depth map is associated with a value for the parameter $t$, as discussed previously. It is therefore possible to compute the corresponding three dimensional point in the world, that is, the surface point of the object's visual hull, using equation (5.1). This world point can then be projected into the appropriate reference view using its camera projection matrix and the corresponding colour value can be read. Since the texture mapping is to be view-dependent the reference camera that should be selected is the one that has the closest viewpoint to that of the virtual camera [7, 6].

The most appropriate view can be determined by considering the angle between the vector linking the virtual camera's centre of projection to the surface point of the visual hull, and the vector



(a) No check implemented. (b) Partial check implemented.

Figure 5.4: The consequence of not determining the valid regions of the epipolar lines. Both images show a depthmap representing the visual hull of a coffee cup. Lighter pixels represent points that are closer to the virtual camera. The visual hull in (a) exhibits a large region of error when compared to the visual hull in (b).

linking the reference camera's centre of projection to the same surface point. The reference view associated with the smallest angle is the view that must be used. This texturing strategy is shown in figure 5.5.

A problem that arises is that although a camera may have a favourable viewing angle it may not have an unoccluded view of the surface point. Hence, to improve the quality of the texturing process the visibility of the surface points for each reference view must first be determined. Matusik, et al. [24] proposes an approach that compares points lying in the same epipolar plane. When testing the visibility of a surface point for a particular reference camera the only points that might occlude it from view will lie in the epipolar plane formed by itself, the virtual camera, and the actual reference camera.

For each reference view a visibility map or mask is created—this map is a binary image with each pixel specifying the visibility of the surface point observed by the corresponding pixel in the virtual image. The computation can thus be done as follows. In the virtual image iterate through each pixel of the observable epipolar line segment ranging from the epipole of the reference camera to a pixel on the image border. This is repeated for a limited number of border pixels depending on the location of the epipole in the image plane. The direction in which the line segment is scanned is determined by the location of the reference camera with respect to the

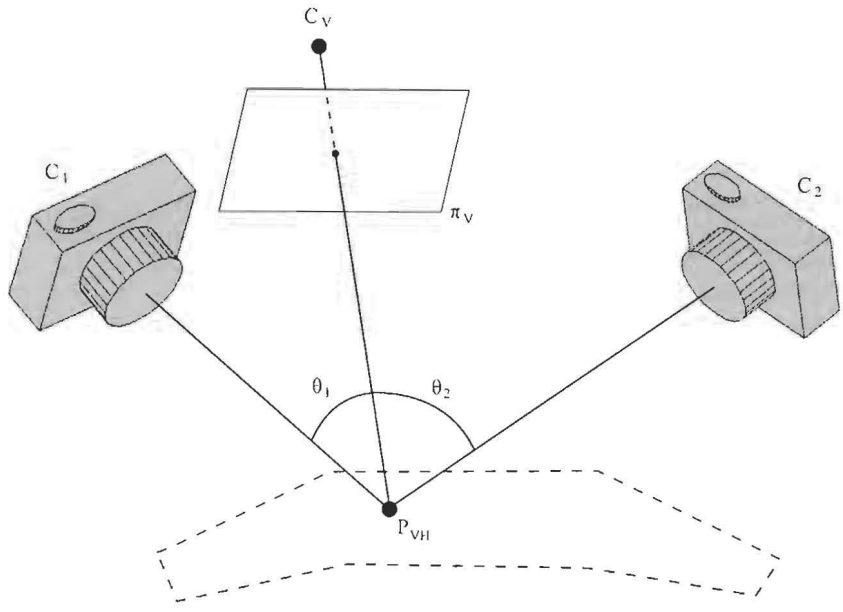



Figure 5.5: The view that has the closest viewpoint to that of the virtual camera is used for shading. $C_v$ represents the virtual camera's centre of projection while $C_1$ and $C_2$ represent two reference cameras. $P_V H$ is a point on the surface of the visual hull.

virtual camera's image plane. If the reference camera is in front then the appropriate direction is from the epipole to the image border. If the reference camera is located behind the image plane then the line segment must be scanned in the opposite direction, namely from the image border towards the epipole. This ordering is important as it determines the occlusion ordering of the sampled points of the visual hull.

Certain pixels along a particular epipolar line segment will image the surface of the visual hull. These pixels are each associated with a list of three dimensional line segments, coinciding with their respective visual rays, recorded during the computation of the visual hull. Projecting these line segments into the reference image will form an occlusion mask. If a surface point is subsequently projected to a pixel that is covered by the mask it will be classified as being *not visible* from the reference camera. Figure 5.6 depicts graphically the process of forming the occlusion mask.

Iterating through the epipolar line segment in the virtual image the current pixel is checked to see if it images a surface point. If it does, the surface point is projected into the reference image. This projection is compared to the occlusion mask and the reference image's visibility map is

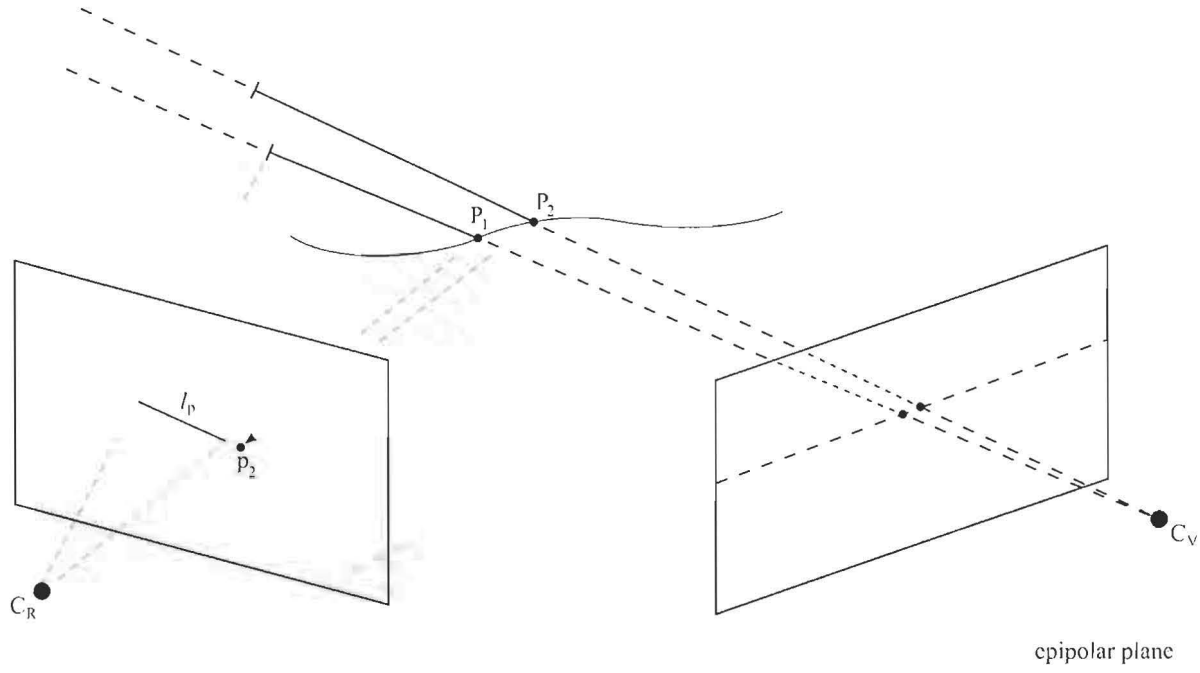
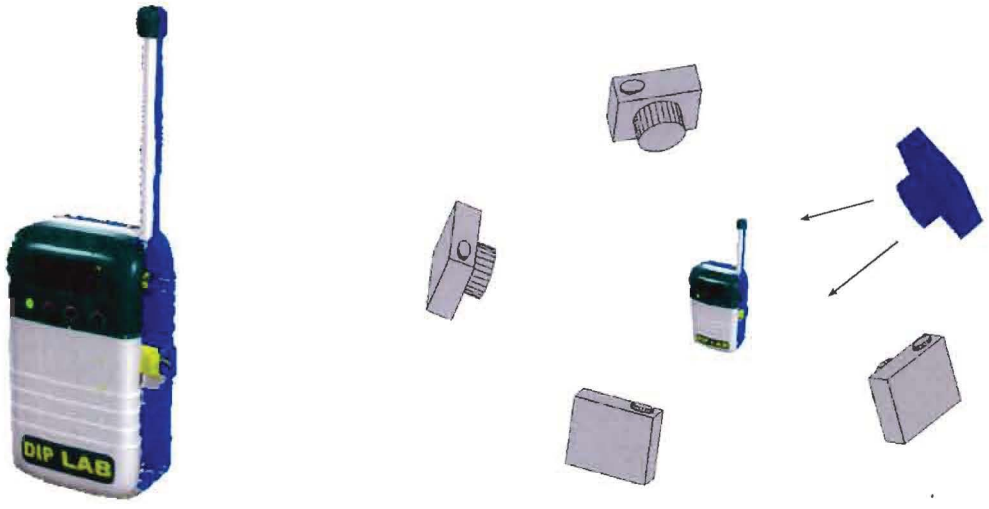



Figure 5.6: Testing the visibility of the surface points of the visual hull by analysing all points in an epipolar plane. $P_1$ and $P_2$ are points on the visual hull of the object. The image point $p_2$ (projection of $P_2$) is not covered by the occlusion mask ($l_p$) and is thus recorded as being visible from the reference camera $C_R$.

updated appropriately. The occlusion mask is then updated by adding the projections of the line segments that coincide with the pixel's visual ray.

Due to the discrete nature of an image a pixel in the virtual image may be visited more than once. In other words, more than one epipolar line may pass through that particular pixel. The visibility information from each visit must therefore be combined—the surface point is either labelled as being visible if it is visible on *every* visit, or it is labelled as being visible if it is visible on at least *one* visit. Furthermore, the algorithm will only produce an approximation to the actual visibility [23]. Filtering the visibility map can remove some of the noise caused by errors in calculation. An example of a visibility map overlaid on the computed visual hull is given in figure 5.7.

Once the visibility map for every reference view has been computed the correct colour values can be assigned following the procedure mentioned previously. The surface point is projected into the reference view with the most appropriate viewpoint, and from which it is also visible.




(a) Virtual image with the visibility map of the third camera overlaid.

(b) The position of the reference cameras with respect to the object.

Figure 5.7: The visibility map of the third reference camera (blue camera in (b)) is overlaid on the virtual image thus showing which points it can view.

The colour value is then calculated by performing a *bilinear interpolation* [28] on the four pixels surrounding the projected point.

# Chapter 6

# Results

This chapter discusses the results obtained during the testing of the virtual view synthesis techniques that were implemented. Through the analysis of the results the relative performance of the implementations can be gauged. The algorithms receive a set of colour images and a set of binary images as input, representing the reference views and the corresponding silhouettes of an observed object. The silhouettes were obtained by manually segmenting the reference images using image editing software[1]. Manual segmentation will, however, not be feasible when developing real-time applications or when the input data set is very large. Apart from the input images the algorithms also require the calibration parameters of the reference cameras, as well as the calibration parameters of the virtual camera for the desired view.

The output of the algorithms always includes a colour image representing the virtual viewpoint. Any additional output is unique to a particular implementation, and depending on the technique could consist of a geometric model of the object's visual hull, or a depth map, with the intensity values of the pixels giving the relative distance to the surface of the visual hull.

The input data sets consist of images captured from real cameras, as well as synthetically generated images acquired from three dimensional modelling software. The appeal of incorporating computer-generated views into the analysis is that the algorithms can be tested in an environment free from segmentation and calibration errors. The sensitivity of the algorithms to such errors can be investigated if results that have been recorded in absence of these errors are available. A

[1] GIMP 2.2 – The GNU Image Manipulation Program (http://www.gimp.org)

further advantage is that ground truth images can be easily obtained for any virtual viewpoint desired.

The first section of this chapter introduces the performance measure used to evaluate the implemented techniques. The results for the approach that makes use of geometric reconstruction are then presented, followed by the results for the image-based approach.

## 6.1 Establishing a Measure of Performance

The evaluation of the implementations is based exclusively on the visual quality of the synthesized view. The measure of performance thus involves a comparison between the newly rendered images and additional reference images of the object that were not used in the synthesis process. For the real data sets these "additional reference images" are actual images of the object acquired using a digital camera. In the case of the computer generated data sets these images are rendered using the calibration parameters of the virtual viewpoint.

Comparing any two images is done at a pixel level. Since a view of the object does not occupy every pixel in the image only a subset of the image pixels are processed, thereby limiting the number of background pixels included in the comparison. The region of interest is defined as the smallest rectangular area that will enclose all the foreground pixels in both the additional reference image and the virtual image. When evaluating a particular series of data sets, for instance the data sets of the ceramic cat, the region of interest is kept constant and is chosen so that it encloses all the foreground pixels of each virtual image that is processed.

To quantify the differences the distance in RGB colour space between the colour coordinate of a pixel in the additional reference image and the colour coordinate of the corresponding pixel in the virtual image is calculated [5]. This error measurement is computed for each pixel of interest using the following formula:

$$E = \sqrt{(R_v - R_{ref})^2 + (G_v - G_{ref})^2 + (B_v - B_{ref})^2} \tag{6.1}$$

where $E$ is the distance error in RGB colour space, $[\ R_v \quad G_v \quad B_v\ ]^\mathsf{T}$ are the red, green, and blue colour values, respectively, of the pixel in the virtual image, and $[\ R_{ref} \quad G_{ref} \quad B_{ref}\ ]^\mathsf{T}$ are the red, green, and blue colour values, respectively, of the corresponding pixel in the additional

reference image to which the virtual image is being compared. A better approach would be to consider the way humans perceive colour and hence make use of a colour space where the distance between colour coordinates is related to the difference in observed colour. These are referred to as perceptually *uniform* colour spaces [13]. Investigating such an error measure is, however, beyond the scope of this work.

The mean distance error for an image is calculated by averaging the error values for all the processed pixels, thereby giving an indication of the quality of the rendered output. This value is used to compare techniques and also investigate what factors have a significant influence on the quality of the synthesized images. One such factor that is investigated is the number of reference views used in the synthesis process. Whenever possible the new views of an object are generated using a varying number of input images to determine the effect on the quality of the output.

Furthermore, the error values are plotted as an intensity image to produce an *error map*. This map is used to determine whether the errors are localized or distributed uniformly across the image.

An experiment conducted for both the geometry-based and the image-based techniques is to determine how the quality of the virtual image is affected as the virtual camera is moved further away from the nearest reference camera. This is performed using the computer-generated data sets because the additional reference images for the novel viewpoints used in the comparison can easily be obtained—greater control is therefore had over the positioning of the virtual viewpoints. Since the original reference views were obtained under circular motion the distance of the virtual viewpoint from the nearest reference camera can be specified as the angular displacement between the vectors linking the two cameras to the centre of the circle. Figure 6.1 illustrates how the angular displacement influences the distance between the two cameras.

Part of the experiment also involves varying the number of reference views given as input. As the number of views increases the applicable range of the angular displacement decreases since the angular displacement between the reference cameras decreases. The applicable range is always half the angular displacement between any two of the reference cameras—further displacement from the one camera will mean the virtual viewpoint is closer to the second camera. In the data tables presented in the next two sections angles that are not applicable are indicated with "n/a".

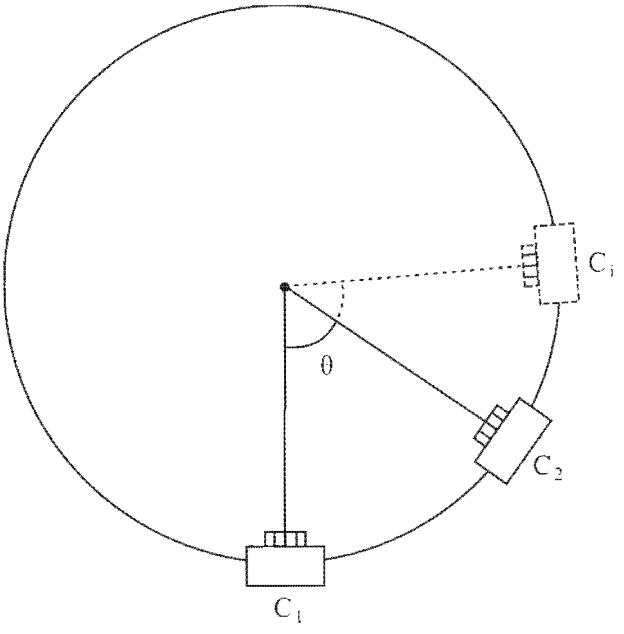



Figure 6.1: The position of a second camera relative to the first reference camera is represented as an angular displacement $\theta$.

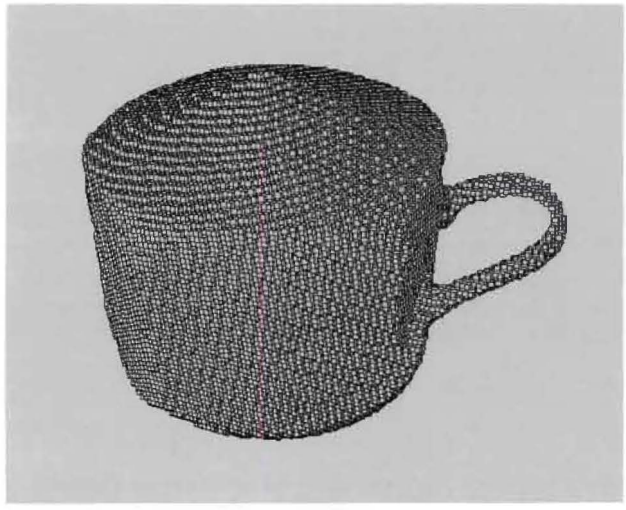
## 6.2 Performance of the Geometry-based Rendering Technique

### The Coffee Cup Data Set

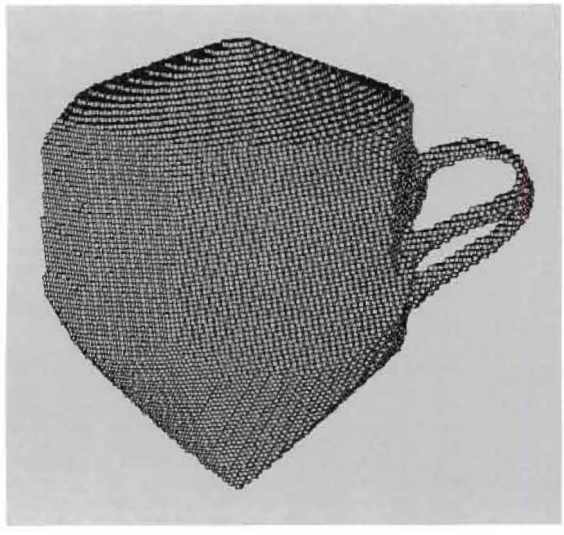
The first set of reference images given as input to the geometry-based rendering technique consisted of five real images of a colourful cup captured using a turntable. This was the only available data set of the coffee cup. There were also no additional views available for comparison so no quantitative evaluation could be performed.

Using the reference images the algorithm constructed a shaded voxel model of the visual hull of the cup and then rendered the novel view as seen from the specified virtual viewpoint. The untextured and textured versions of the voxel model were also output—the untextured model, as seen from a number of different viewpoints, is shown in figure 6.2.

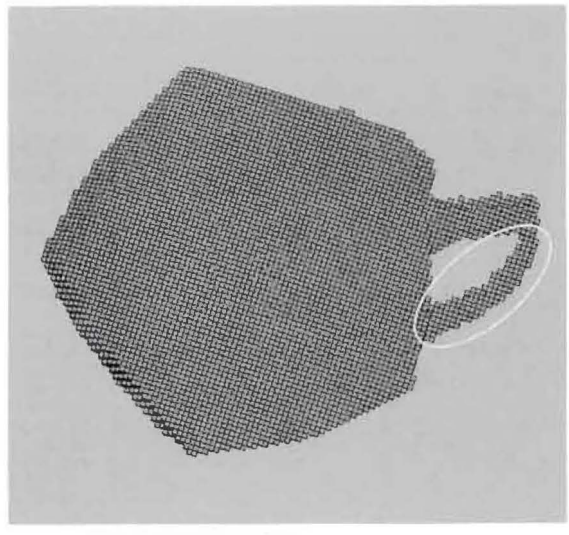
Inspection of the model reveals a number of obvious differences between its geometry and that of the real cup. Firstly, the concavity of the cup is not modelled. As discussed in section 2.3 the visual hull can not model the concave areas of an object since they are never visible in the object's silhouettes. However, since the goal is not model reconstruction, but the synthesis of



(a) Voxel model of coffee cup as seen from camera 1



(b) Voxel model viewed from the side



(c) Voxel model viewed from above

Figure 6.2: Voxel reconstruction of a coffee cup. This model was created from five silhouettes using seven levels of voxel subdivision. The extra volume that was not carved from the model can be seen in figure (b) forming the cone of voxels at the top and bottom of the cup. The additional voxels protruding from the handle are highlighted in (c) and are also visible in (b).

novel views, proper shading could compensate for the inaccurate geometry [23].

For the cup data set this will only be effective for viewpoints that are at the same height as the reference views. As is evident in figure 6.2(b), not only was the concavity not modelled, but extra volume was added to the top and the bottom of the visual hull creating the cone-like appearance. This extra volume is consistent with the silhouettes of the cup and was thus never carved away—this consistency is depicted graphically in figure 6.3. The volume is consistent with the silhouettes in that if the voxels belonging to the extra volume are projected into the reference views they will occupy areas of the image that are covered by the silhouettes of the cup.

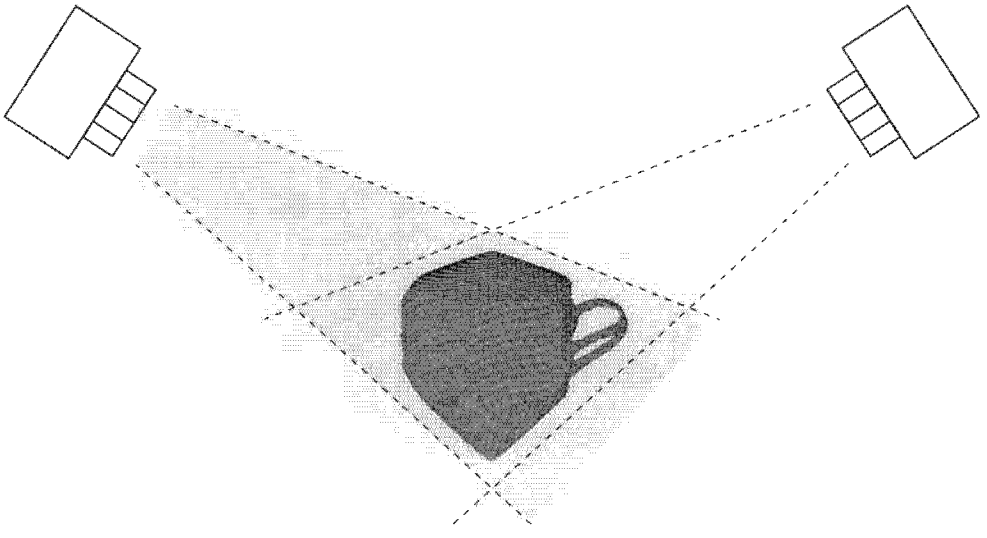


Figure 6.3: The volume occupied by the model is determined by the intersection of the visual cones of the cameras defined by the cup's silhouettes. This figure illustrates how, due to the elevated position of the cameras, the extra volume noted in figure 6.2(b) is included in the intersection of the visual cones and is thus consistent with the silhouettes.

This modelling error is a consequence of the reference viewpoints being at an elevated position in relation to the cup, viewing it at a downward angle. Figure 6.2(b) represents a viewpoint at a lower position (more or less the same height as the cup) than that of the reference views and thus shows how novel views synthesised from this position will be negatively affected by the extra volume. The model can be made more accurate by having at least one reference view at a position closer to the ground plane.

A further modelling error, apparent in figure 6.2(c), is the additional voxels protruding from the

handle of the cup. These voxels were also not carved away due to them being consistent with the input silhouettes. In the case of a turntable sequence this error can be minimised by increasing the number of reference views or, alternatively, overcome by adding views that are not part of the circular motion [40]. For instance, by supplementing the turntable sequence with a view from directly above the cup (from a viewpoint similar to figure 6.2(c)) the invalid voxels will be carved. Besides correcting the handle the overall shape of the model, as seen from above, will be more circular instead of resembling that of a pentagon, thus more accurately matching the actual shape of the cup.

Figure 6.4(b) shows a reprojection of the shaded model into the first reference view. The original reference view is also shown for comparison. Figure 6.4(c) and 6.4(d) show a novel views of the cup created using five reference images.

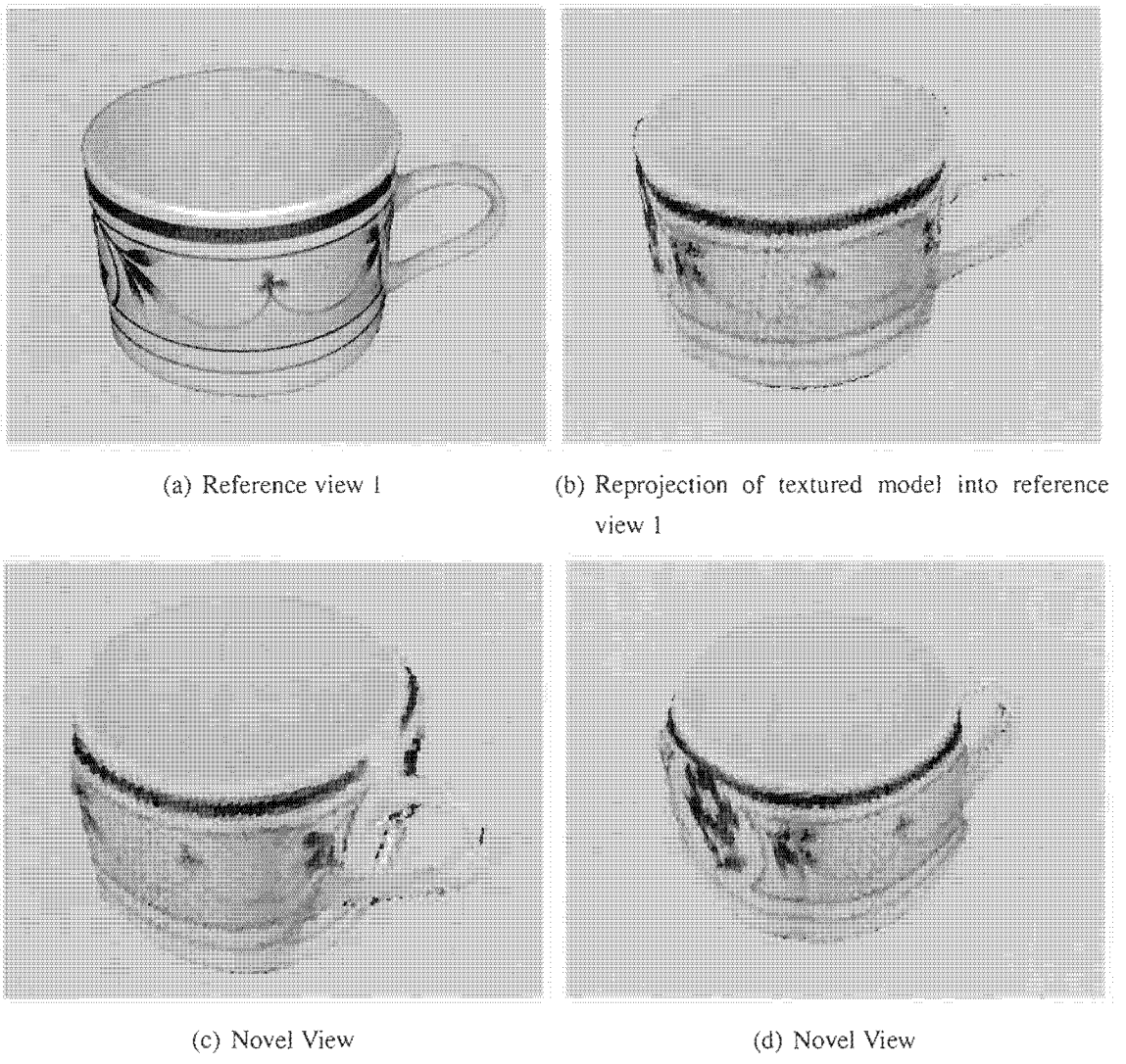
### The Ceramic Cat Data Sets

The second series of data sets provided as input to the algorithm consisted of images of a ceramic figurine of a cat. More calibrated images of the ceramic cat were available than that of the cup and therefore the number of images provided as input could be varied. The aim was to determine how the average error values are influenced as the number of reference views is increased.

The octree models of the cat were created using seven levels of subdivision—effectively a cube of $128 \times 128 \times 128$ voxels. Figure 6.5 shows two of these voxel models with the first being constructed using six levels of subdivision and the second using seven levels. By using voxels of a smaller size a more accurate approximation to the visual hull can be achieved. The texture mapped onto the model will also be more detailed.

The output of the algorithm after using five reference views exhibited similar characteristics to the output obtained using five views of the cup—figure 6.6(a) highlights the voxels that have been incorrectly added to the model. These additions are largely due to the limited number of reference views provided as input. However, with ten views the error is reduced to give a better approximation of the object's visual hull (this is shown in figure 6.6(b)).

Shading the models was accomplished using a view *independent* texture mapping strategy.



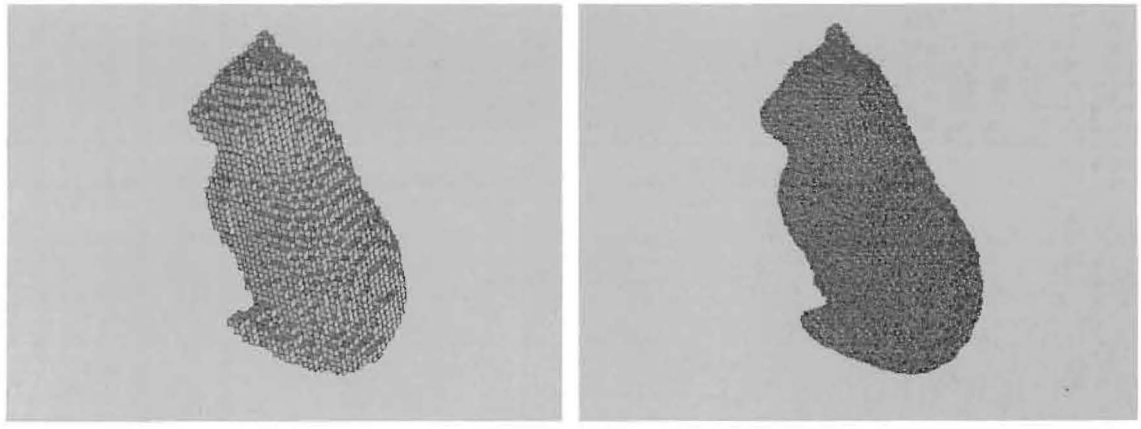
(a) Reference view 1

(b) Reprojection of textured model into reference view 1

(c) Novel View

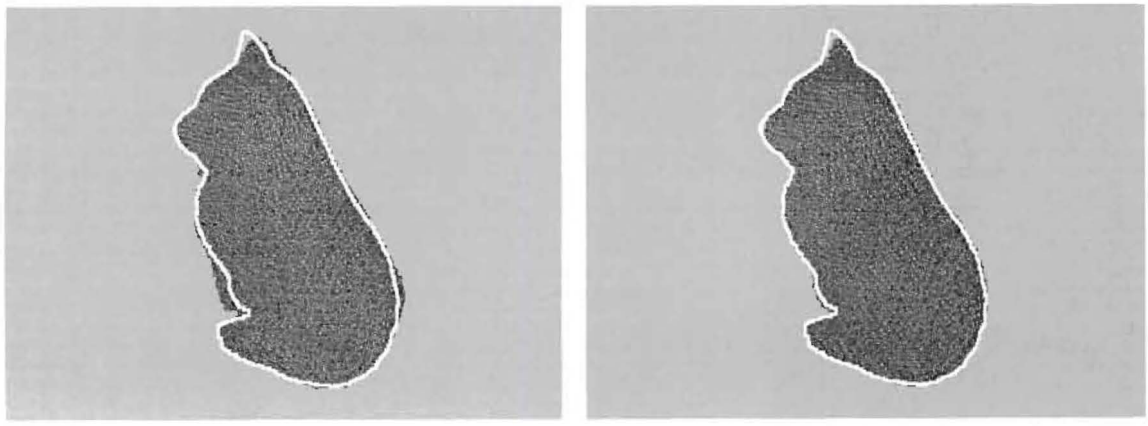
(d) Novel View

Figure 6.4: Novel views of a coffee cup produced by the geometry-based approach from five reference views. A textured voxel model was created using seven levels of subdivision and then rendered to produce the novel views. The virtual cameras are at approximately the same height as the reference cameras and therefore the additional volume on the top of the model is not visible. In fact, the appropriate textures give the illusion that the concavity of the cup is present just like in reference view one. Pixels highlighted in bright green indicate that a colour value could not be assigned—these correspond to voxels that were not visible in any of the reference views.



(a) Octree model of cat with 6 levels of subdivision (b) Octree model of cat with 7 levels of subdivision

Figure 6.5: Octree models of a ceramic cat created using sixteen reference views.



(a) Model of cat created using 5 reference views (b) Model of cat created using 10 reference views

Figure 6.6: Two models of the ceramic cat are shown from a novel viewpoint. The outline of the cat from the desired viewpoint is overlaid on both images. The voxels lying outside the outline in figure (a) represent the extra volume that was not carved during the reconstruction process.

Figure 6.7 shows novel views of the cat generated by rendering the textured voxel models created using a set of five and a set of ten reference views. The ground truth image is presented along with the novel views for comparison, as well as the associated error maps.

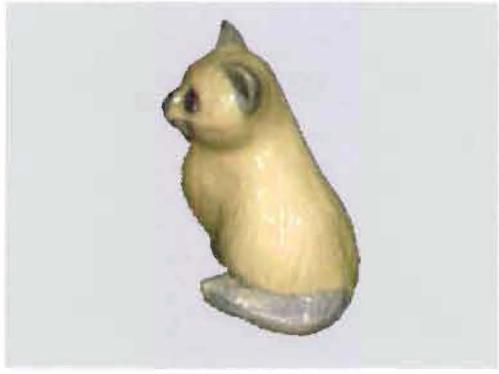
From the error maps it is evident that the most significant errors occur near the edges of the rendered foreground. This is a result of the limited resolution of the model due to the size of the voxels, and also the limitation on the accuracy of the approximation of the visual hull due to the finite number of input silhouettes used [19]. This is especially evident when using only five images. A further cause of significant error is the inability to adequately approximate the relative illumination of the photographed object such as *image highlights*—a result of the view independent texture mapping. This problem is made worse in that the sequence of views is captured using a turntable with a dominant light source. In such circumstances the illumination of the object is effectively changing between views since the object is moving. As can be seen in figure 6.7(a), the reference image of the ceramic cat exhibits a much greater contrast in illumination between the centre and the edges of the cat, which is not apparent in the rendered novel views shown in figures 6.7(b) and 6.7(c). This difference between the images is highlighted in the error maps of figures 6.7(d) and 6.7(e). It is a direct consequence of averaging the colour values obtained from the reference images. The consequences of having a dominant light source when capturing a turntable sequence are further discussed and illustrated in section 6.3.

Novel views of the ceramic cat were also generated using a set of eight and a set of sixteen reference images. The average error was then calculated for each virtual image generated, thereby associating each set of reference images with a measure of quality. The results are presented in table 6.1.

Table 6.1: **Geometry-based rendering:** Average error calculated for the ceramic cat data sets.

| Number of Views | Average Error |
|---|---|
| 5 | 33.49 |
| 8 | 27.54 |
| 10 | 28.27 |
| 16 | 28.16 |

The table reflects the observations mentioned previously—using more than five images reduces



(a) Desired novel view

(b) Novel view created using 5 reference views

(c) Novel view created using 10 reference views

(d) Error map – 5 reference views

(e) Error map – 10 reference views

Figure 6.7: Novel views of a ceramic cat produced by the geometry-based approach while varying the number of reference views used. Lighter pixels in the error maps represent larger errors. The most significant errors in (d) are caused by the addition voxels that were not carved near the edges of the cat. Further errors are due to the illumination of the cat as discussed in the text.

the average error of the new image. There is little difference between results obtained when using eight, ten or sixteen reference views as input. This could be because of the limited resolution of the model due to the size of the voxels. The more accurate approximation of the cat's visual hull afforded by the extra reference views can only be obtained if voxels with smaller dimensions are used. Alternatively, if the visual hull is already sufficiently accurate after using only eight views then the average error can only be reduced by improving the shading of the models.

### The Toy Figurine Data Sets

The next set of data sets consist of images of a toy figurine acquired using a digital still camera from multiple viewpoints around the object. No quantitative evaluation of the output from the geometry-based technique was performed when using these data sets as input. The octree models were created using seven levels of subdivision. The novel views rendered are shown in figure 6.8. Despite the limited resolution of the model some of the detail on the chest of the figurine can still be identified.

### The Radio Data Sets

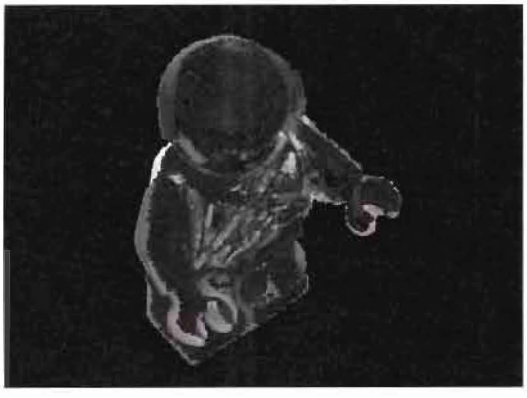
The final series of data sets consist of computer generated synthetic views of a radio. Since ground truth images could easily be generated an experiment was conducted to determine how the average error associated with the novel views varied as the distance of the virtual viewpoint from the nearest reference camera increased. Similarly, the number of reference views given as input was also varied. The models were again created using seven levels of subdivision and shaded using a view independent texture mapping. The results are given in table 6.2 on page 64.

As with the ceramic cat sequence, using more than five reference views decreases the average error value. There is, however, a significant increase in the average error when using ten reference views as opposed to eight. This increase, although much smaller, is also observed in table 6.5 which records the results of this same experiment for the image-based rendering technique. A possible explanation for this behaviour is that it is related to the relative placement of the cameras, which were spaced equally around the radio. Figure 6.9 shows the locations of the cameras with respect to the radio when acquiring eight and ten reference views, respectively.

(a) Desired novel view

(b) Novel view created using 5 reference views

(c) Novel view created using 16 reference views



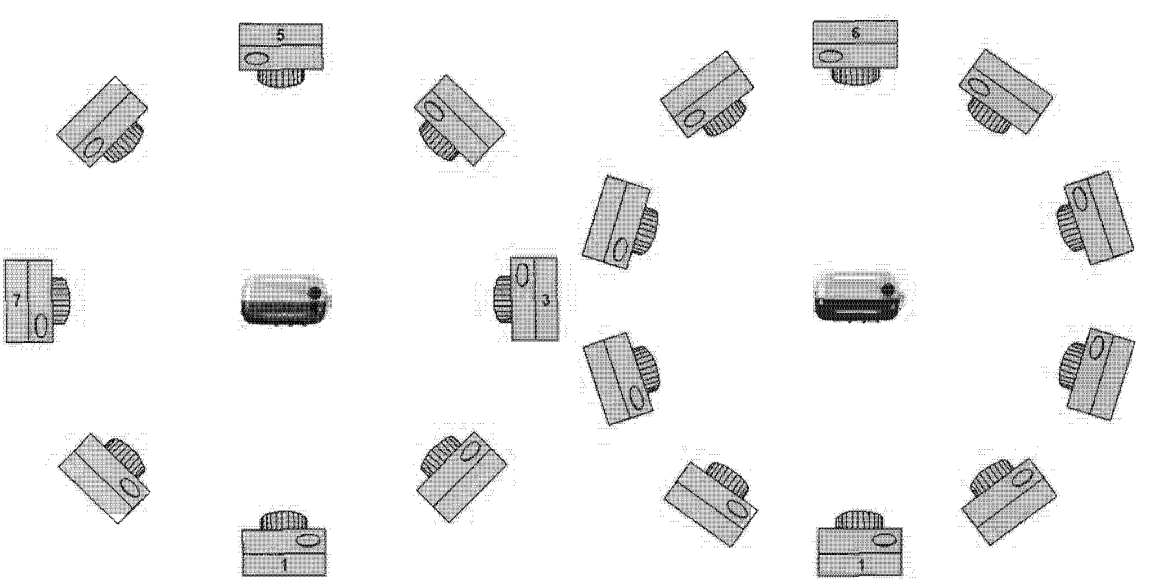
(d) Error map – 5 reference views

(e) Error map – 16 reference views

Figure 6.8: Novel views of a toy figurine produced by the geometry-based approach using a varying number of reference views. Bright green pixels represent points on the visual hull that were not assigned a colour. Lighter pixels in the error maps represent larger errors. In (c) the detail on the figurine's chest is clearer than in (b).

Table 6.2: **Geometry-based rendering:** Average error calculated for the radio data sets.

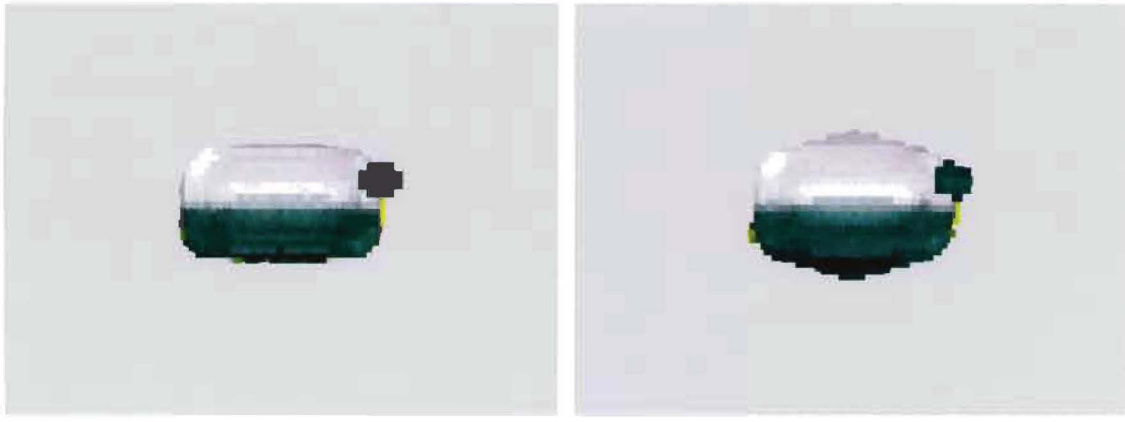
| **Number of Views** | **Angular Displacement** | | | | | |
|---|---|---|---|---|---|---|
| | 5° | 10° | 15° | 20° | 25° | 30° |
| 5 | 19.15 | 18.84 | 18.60 | 18.17 | 17.72 | 16.99 |
| 8 | 15.96 | 15.26 | 14.97 | 15.00 | n/a | n/a |
| 10 | 18.45 | 17.35 | 16.78 | n/a | n/a | n/a |
| 16 | 17.00 | 15.80 | n/a | n/a | n/a | n/a |



(a) 8 camera configuration (b) 10 camera configuration

Figure 6.9: The camera configurations when viewing the model radio.

The body of the radio is in the shape of a rectangular box with flat faces and slightly rounded edges. The difference between the camera configurations of the two data sets is that with eight cameras, four of the cameras (cameras 1, 3, 5, and 7 in figure 6.9(a)) are positioned parallel to the radio's faces while with ten cameras only two are parallel. As can be seen in figure 6.9(b), there are no cameras parallel to the front face of the radio. The result is that the front face of the model constructed using ten views is curved, and not flat as it should be, thus causing its shading to appear warped which increases the associated error. This curvature is clearly visible in figure 6.10.



(a) Model of radio created using 8 reference views (b) Model of radio created using 10 reference views

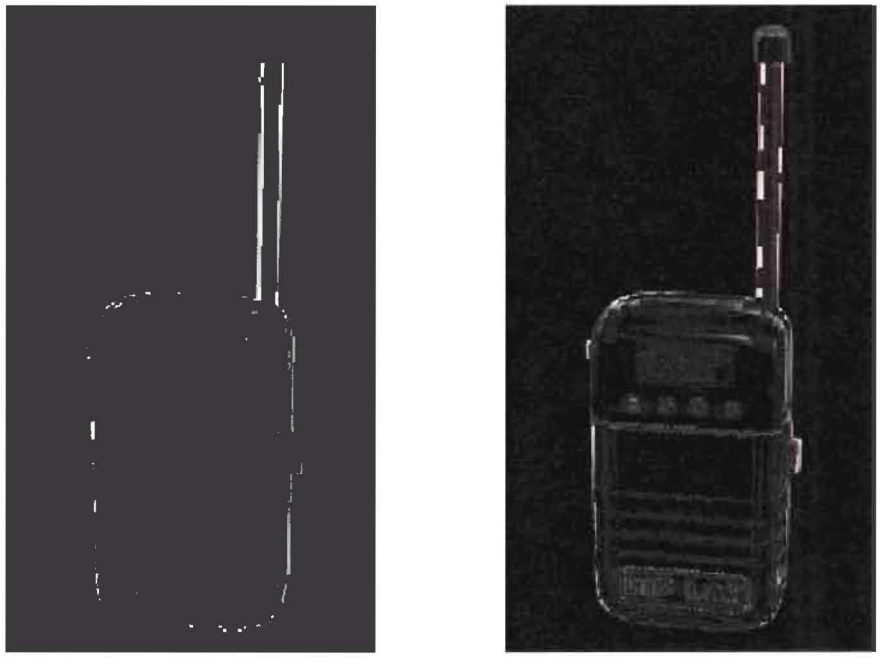
Figure 6.10: Octree models of a radio viewed from above. The model shown in (b) was created from ten reference views positioned as in figure 6.9(b). The front and back faces of the model are both curved, unlike the model in (a) which was created from eight reference views positioned as in figure 6.9(a).

A further unexpected trend in the results of table 6.2 was that the average error decreased as the virtual viewpoint moved away from the first reference camera. It is proposed that this anomaly is related to the resolution of the voxel models. The front face of the radio has detailed elements such as text and small buttons which cannot be accurately represented due to the size of the voxels. The side face of the radio, however, is less detailed and thus its texture can be more closely approximated. As the angular displacement of the virtual viewpoint from the first reference view is increased, so the area occupied by the projection of the front face onto the image plane decreases and the area occupied by the side face increases. This point is illustrated in figure 6.11. Since the approximation of the texture of the side face is more accurate the average error value will decrease. Two novel views of the radio are shown in figure 6.12.


(a) 10° angular displacement (b) 30° angular displacement (c) 45° angular displacement

Figure 6.11: As the angular displacement between the virtual camera and the first reference camera increases so the number of pixels occupied by the side face (2) increases while the number of pixels occupied by the front face (1) decreases.


(a) Desired novel view (b) Novel view created using 5 reference views (c) Novel view created using 16 reference views



(d) Error map – 5 reference views (e) Error map – 16 reference views

Figure 6.12: Novel views of a radio produced by the geometry-based approach. Besides the errors along the edges of the radio (particularly along the antennae) there are also regions of significant error on the front face. The detail of the radio could not be reproduced due to the limited resolution of the voxel model.

## 6.3 Performance of the Image-based Rendering Technique

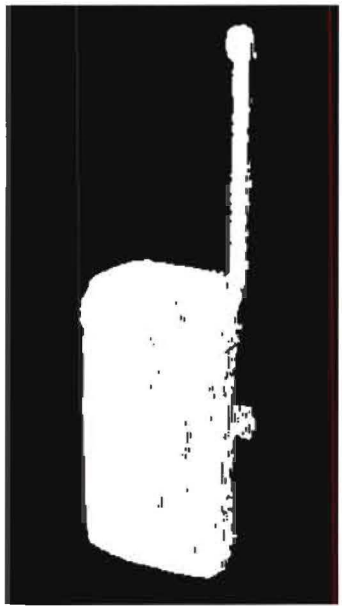
An important part of the image-based method is the creation of the visibility maps for each reference camera [24]. These maps resemble the shadow maps that would be created if the reference cameras were replaced by light sources, and for a particular light source, the scene points illuminated by its rays were determined [23]. Figure 6.13 gives two examples of the visibility maps along with one overlaid on the virtual image to show which points are visible from the reference camera in question.

The mapping of colours from the reference images to the virtual image would create large areas of error if the visibility of points on the visual hull were not taken into account. The consequences of neglecting the visibility calculations are highlighted in figure 6.14. In the images the errors take the form of streaking colours as the incorrect reference camera was chosen for the texture.

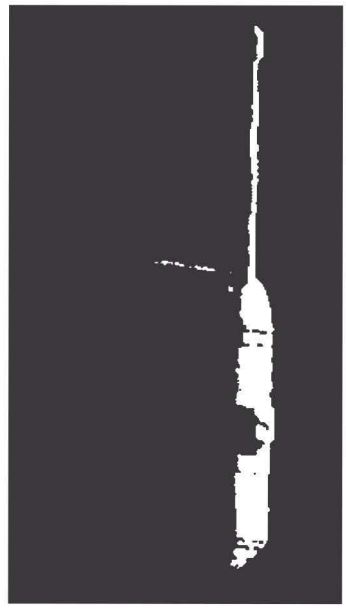
### The Coffee Cup Data Set

The five view data set of a coffee cup captured using a turntable was also processed using the image-based implementation. As with the geometry-based approach no quantitative evaluation was performed. The depth map representing the cup's visual hull showed the same volume additions as with the voxel reconstructions, namely the dome on the top and the protrusions from the handle. As discussed in section 5.2 the shading of the visual hull was accomplished using a view dependent texture mapping. A novel view of the cup is shown in figure 6.15, along with its corresponding depth map.

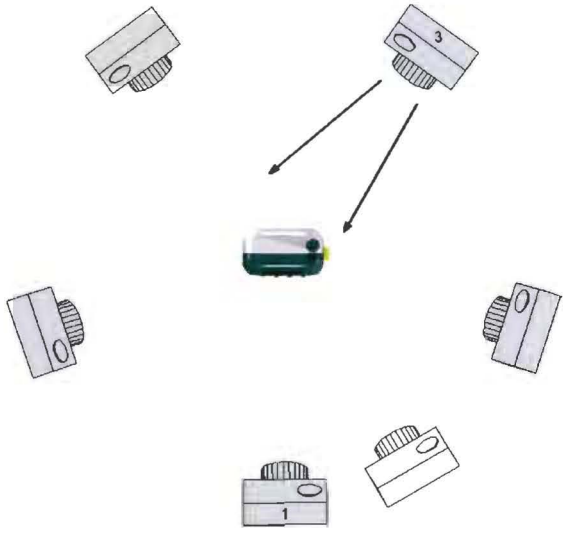
Inspection of the new view reveals neighbouring areas with a high variation in illumination, that is, an area of darker pixels bordering on an area of much lighter pixels. These two sets of pixels were mapped from two different reference images due to visibility constraints. This problem with the brightness can be attributed to the fact that the images were captured using a turntable with a dominant light source, and the object was therefore not subject to constant illumination. In other words, the same point observed in one reference view could be much lighter or darker than when observed in another reference view.



(a) Visibility map for reference view 1



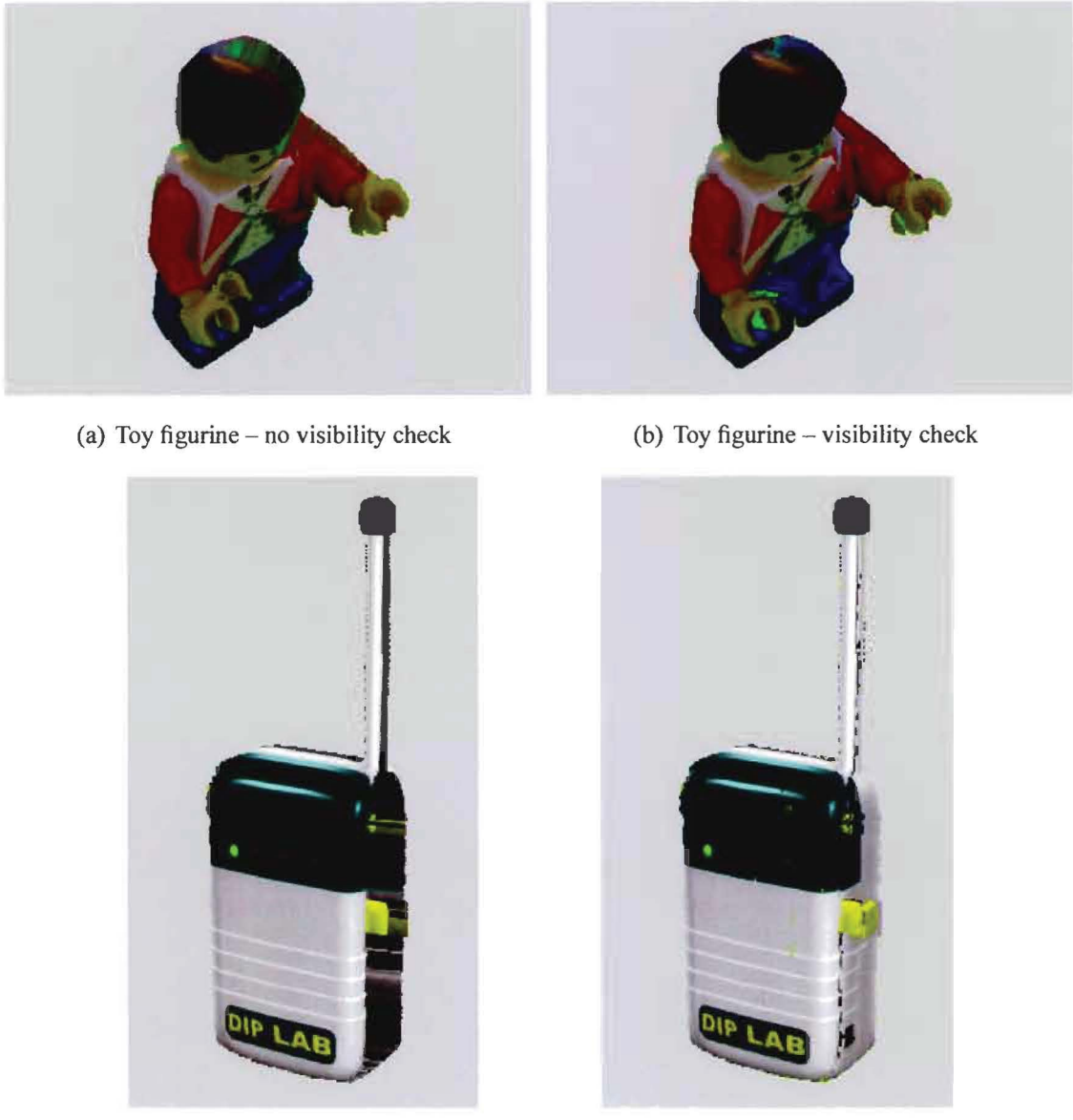
(b) Visibility map for reference view 3




(c) The camera configuration


(d) Novel view of radio with the visibility map of view 3 overlaid in blue

Figure 6.13: The visibility maps for reference views one and three are shown in (a) and (b), respectively—white pixels represent points of the visual hull that are visible. The camera outline in (c) shows the position of the virtual camera.



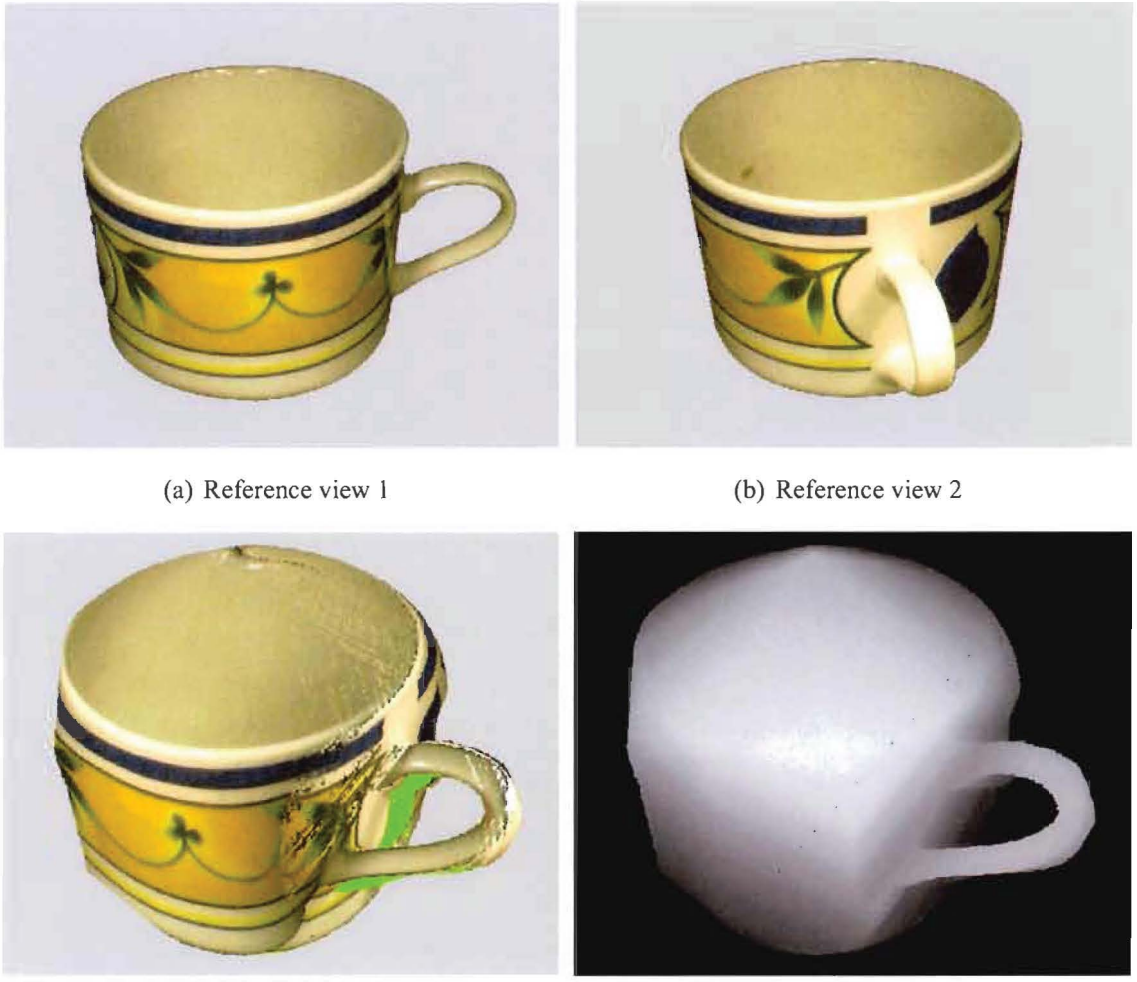

(a) Toy figurine – no visibility check

(b) Toy figurine – visibility check

(c) Radio – no visibility check

(d) Radio – visibility check

Figure 6.14: Figures (a) and (c) show the consequences of not taking the visibility of points into account when mapping colours from the reference views. The streaking colours along the side of the radio represent colour values that were mapped from the incorrect reference view.



(a) Reference view 1

(b) Reference view 2

(c) Novel view

(d) Depth map representing visual hull

Figure 6.15: Novel views of a coffee cup produced by the image-based approach using five reference views. Lighter pixels in the depth map represent points on the visual hull that are closer to the virtual camera. Bright green pixels represent points that were not assigned a colour.

### The Ceramic Cat Data Sets

A quantitative evaluation of the image-based approach was performed using the data sets of the ceramic cat. The aim was to determine how the average error values are influenced as the number of reference views is increased. Figure 6.16 shows the novel views rendered of the ceramic cat. These images also have neighbouring areas of high contrast since the reference views were acquired under the same conditions as the cup. An example of such an area can be seen running along the edge of the cat.

From the images it is evident that the view generated using ten reference views is a closer approximation to the actual view than the view generated using five reference views. This observation is reflected in table 6.3 which gives the average error for the novel views rendered while varying the number of reference views used as input. As the number of reference views increases so the average error calculated decreases. This is the expected behaviour because adding more views not only increases the amount of photometric information available but it also refines the approximation of the object's visual hull due to the increased number of silhouettes [19].

Table 6.3: **Image-based rendering:** Average error calculated for the ceramic cat data set.

| Number of Views | Average error |
|---|---|
| 5 | 31.66 |
| 8 | 27.59 |
| 10 | 27.14 |
| 16 | 26.45 |

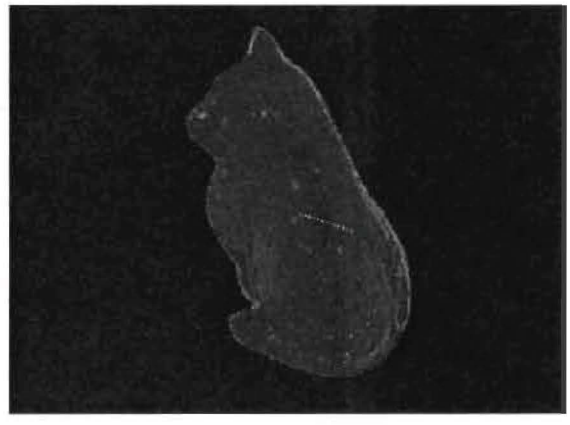
In the case of the geometry-based approach the level of detail that can be reproduced in the novel view is restricted by the resolution of the voxel model. Each voxel will generally map to more than one pixel in the new image. In contrast, the implemented image-based method establishes a separate mapping of colour between a *single* pixel in the virtual image and one of the reference images. The level of detail reproduced is therefore influenced by the image resolution. Both implementations were affected by the problem of inconsistent illumination, but because of the ability to reproduce greater levels of detail combined with the view dependent texture mapping the image-based approach achieved lower average error values than the geometry-based implementation.

(a) Desired novel view

(b) Novel view created using 5 reference views

(c) Novel view created using 10 reference views

(d) Error map – 5 reference views



(e) Error map – 10 reference views

Figure 6.16: Novel views of a ceramic cat produced by the image-based approach. Lighter pixels in the error maps represent larger errors. The most significant errors in (d) along the right edge of the cat are due to the illumination issues as discussed in the text.

### The Toy Figurine Data Sets

When evaluating the image-based implementation using the toy figurine data sets the average error values obtained for two novel viewpoints were recorded. Again the aim was to establish how the average error values are influenced as the number of reference views is varied.

In contrast to the cup and the ceramic cat the novel views of the toy figurine do not have neighbouring areas exhibiting an extreme change in brightness. The levels of illumination are more constant as the reference images were acquired from separate viewpoints around the object as opposed to rotating the object on a turntable. Two of the novel views that were rendered are shown in figure 6.17.

The novel view generated using sixteen views appears to have fewer errors than the one generated using only five views. In figure 6.17(b), produced using five views, the texture on the chest of the figurine is warped, unlike in figure 6.17(c) which was produced using sixteen views. When examining the associated depth map in figure 6.17(d) it is apparent that extra volume that was not carved from the chest of the figurine might be responsible for the warping. The head of the figurine in figure 6.17(e) is also more round than in figure 6.17(d) since the visual hull was computed using more views. The difference in apparent image quality is confirmed by table 6.4 on page 76, which records the results obtained while measuring the average error for the novel views rendered of the toy figurine. The results listed for virtual view one in this table correspond with the images shown in figure 6.17.

Furthermore, the results suggest that making use of eight or ten reference images to render new views will produce better images. The reason for the increase in the average error when utilizing sixteen views becomes apparent when comparing the rendered views, in conjunction with their error maps, to the actual view. These error maps can be seen in figure 6.18. Although the detail on the chest of the toy figurine is relatively error free it has been offset from the correct position—more so than when using ten images—thus increasing the calculated error values.

The results for virtual view two tend to follow the expected pattern. The causes for the aforementioned inconsistencies can have a number of sources, including problems with the visibility calculation, camera calibration parameters that are insufficiently accurate, or the relative positioning of viewpoints (cameras) around the object. The visibility calculation is only an approximation of the true visibility of points on the visual hull for a given reference view [24, 23]. This, along

(a) Desired novel view

(b) Novel view created using 5 reference views

(c) Novel view created using 16 reference views

(d) Depth map – 5 reference views

(e) Depth map – 16 reference views

Figure 6.17: Novel views of a toy figurine produced by the image-based approach. Lighter pixels in the depth maps represent points on the visual hull that are closer to the virtual camera. Bright green pixels represent points that were not assigned a colour.

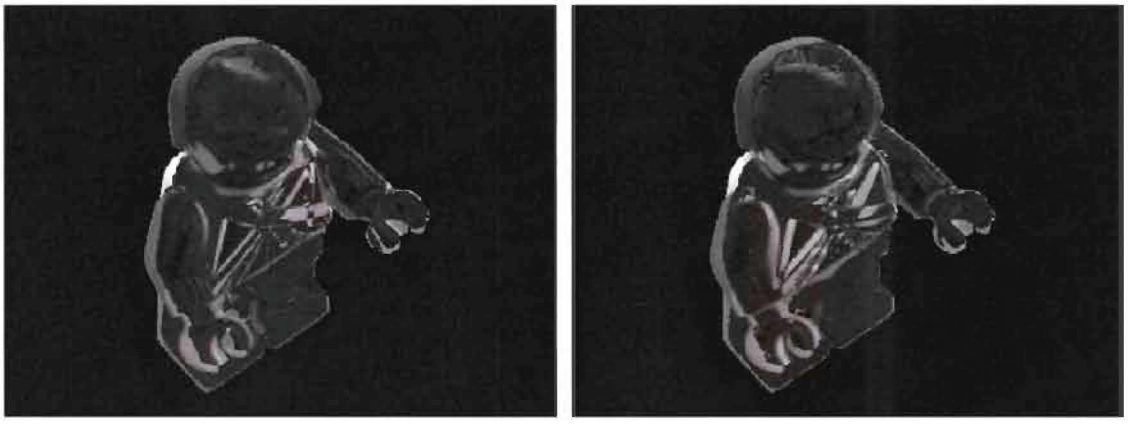
Table 6.4: **Image-based rendering:** Average error calculated for the toy figurine data sets.

| Number of Views | Novel View 1 | Novel View 2 |
|---|---|---|
| 5 | 36.36 | 35.29 |
| 8 | 29.85 | 23.92 |
| 10 | 29.89 | 23.43 |
| 16 | 32.19 | 23.29 |

with possible errors in implementation, will lead to the incorrect mapping of colours between the reference images and the virtual image.

Flawed camera calibration parameters will have a twofold effect on the accuracy of the novel views. Firstly, it can result in a deformed representation of the object's visual hull as volume is incorrectly carved or added. Secondly, the inaccurate projection of world points into the images will cause shading errors.

The relative positioning of the viewpoints or cameras around the object can influence the approximation of the visual hull and hence the quality of the novel view. This concept was discussed previously in section 6.2 and illustrated in figure 6.9 and 6.10.



(a) Error map – 10 reference views (b) Error map – 16 reference views

Figure 6.18: Error maps highlighting the increased error in the region of the chest of the figurine. Lighter pixels represent larger errors.

### The Radio Data Sets

The implementation of the image-based approach to virtual view synthesis was also tested using the computer-generated data sets of a radio. The aim was to investigate how the average error varied as the distance between the virtual viewpoint and the nearest reference camera increased. The results were also recorded while varying the number of reference views provided as input.

As with the toy figurine the novel views of the radio do not suffer from the problems of large variations in illumination. The radio's reference views were also acquired by moving the camera and not the object. However, the novel views generated do show discontinuities in the texturing. An example can be seen in figure 6.19, taking the form of dark line down the side of the radio. This discontinuity occurs on the border between two separate regions of pixels that have been mapped from two different reference views. Blending the colour values that are mapped from the different reference views to this region might reduce the error [24].

Table 6.5 on page 79 lists the results obtained while generating novel views with the radio data sets. The difference between the results achieved for this series of data sets using the geometry-based method and image-based method is even greater than with the ceramic cat. This is because the radio has more fine detail, such as text and buttons, for which the image-based method is more suited.

The results behave as expected, with the average error increasing as the angular displacement between the reference camera and the virtual camera increased. Increasing the number of reference views given as input to the algorithm causes a decrease in the average error except when making use of ten views instead of eight. The reason for this increase is related to the positioning of the viewpoints around the object, as was discussed previously in section 6.2. The error map in figure 6.19(e) highlights how the text in the novel view rendered with ten reference views (figure 6.19(c)) is warped due to the curvature of the computed visual hull.


(a) Desired novel view

(b) Novel view created using 8 reference views

(c) Novel view created using 10 reference views

(d) Error map – 8 reference views

(e) Error map – 10 reference views

Figure 6.19: Novel views of a radio produced by the image-based approach. The text in (c) is warped due to the curved approximation of the front face of the radio's visual hull. The letter "L" appears to be leaning backwards when compared to (a). This error caused by the warping is highlighted in (e). Lighter pixels in the error maps represent larger errors.

Table 6.5: **Image-based rendering:** Average error calculated for the radio data sets.

| Number of Views | Angular Displacement | | | | | |
|---|---|---|---|---|---|---|
| | 5° | 10° | 15° | 20° | 25° | 30° |
| 5 | 6.52 | 11.60 | 14.00 | 15.62 | 16.70 | 16.97 |
| 8 | 5.78 | 7.95 | 9.81 | 10.50 | n/a | n/a |
| 10 | 5.79 | 8.17 | 10.13 | n/a | n/a | n/a |
| 16 | 5.42 | 6.87 | n/a | n/a | n/a | n/a |

## 6.4 Discussion of Results

The visual comparison of the novel views rendered by the image-based and geometry-based approaches to that of the actual virtual view indicates that the image-based approach produces a closer approximation. Analysis of the recorded error values supports this observation. The error values calculated for the image-based approach are generally less than the values calculated for the geometry-based approach.

The reason for this is that the level of detail of the reference images that can be reproduced by the geometry-based method is restricted by the resolution of the voxel model. The image-based approach does not have this limitation. It produces a more accurate sampling of the object's visual hull which is determined by the image resolution of the virtual image [24]. The geometry-based approach, however, creates a quantized sampling of the visual hull related to the dimensions of the voxels. Hence with the image-based technique pixels in the virtual image are individually mapped to the reference images, whereas with the geometry-based method the voxels are generally mapped to more than one pixel.

Both implementations will be negatively affected by inaccurate camera calibration. A number of the unusual results noted in the tests (for the real data sets) could possibly be attributed to incorrect calibration parameters. Having just *one* camera with faulty calibration parameters can distort the approximation of an object's visual hull.

In general, the tests run with a greater number of reference views as input produced better results except in those instances where the relative positioning of the cameras around the object had a

significant influence. Such an example is when, for a consecutive number of runs, the cameras were spaced equally around the object. Thus when new cameras were added the configuration of the original cameras changed. However, adding more viewpoints without adjusting the existing configuration of the cameras will improve the approximation of the object's visual hull [19].

# Chapter 7

# Conclusions

The work presented in this thesis covers the implementation of two different approaches to virtual view synthesis. Both are based on the concept of the visual hull of an object and are thus more suited to synthesising novel views of objects as opposed to whole scenes.

The first approach constructs a voxel model of the approximate visual hull of the observed object. This approximation of the visual hull is determined by the object's silhouettes. The reference images are then used to apply textures to the model, which is then rendered using computer graphics techniques to generate the new view.

The second approach synthesises a novel view by mapping colour values directly from the reference images to the virtual image. The algorithm first finds the surface points of the visual hull that are visible in the virtual image. Each surface point is then projected into the reference view that has the closest viewpoint to that of the virtual camera, and from which the surface point is visible. The colour values that mapped back to the virtual image are calculated by interpolating the colour values of the four neighbouring pixels of the projected image point in the reference image.

The following conclusions can be drawn from the evaluation of the above methods:

- Comparing the results obtained for the two solutions reveals that the image-based approach achieves lower average error values than the geometry-based approach. This suggests that the image-based approach produces a more accurate approximation of the desired

virtual view. To create the virtual image the image-based approach performs a per-pixel sampling of the object's visual hull and thus generates a per-pixel mapping of colour values from the reference images. The geometry-based approach, however, produces a quantized approximation of the object's visual hull with each voxel generally mapping to more than one pixel in the virtual image, thus limiting the level of detail that can be reproduced.

- The relative positioning of the cameras around the observed object can influence the accuracy of the approximated visual hull and thus the quality of the synthesised views. As was noted in the results section (chapter 6), a flat surface will not be computed as being flat unless it is observed by a camera that is positioned at a viewpoint parallel to that surface.

- Synthesising novel views from reference images that were acquired using a turntable with a dominant light source will have a negative affect on the quality of the rendered view. The illumination of points on the object will vary between the reference views as a result of the dominant light source. In the case of the image-based method that made use of a view-dependent texture mapping the errors in the new view appeared as neighbouring regions of pixels with a high contrast in brightness. The view-independent geometry-based method does not exhibit areas of high contrast as it averages the relevant colour values.

- In general, increasing the number of reference views used to generate the novel view decreases the average error achieved. Possible reasons for the observed anomalies, other than the cases related to the relative positioning of the cameras, include camera calibration that is not sufficiently accurate or general implementation errors. Adding accurately calibrated views to an existing set of views will improve the approximation of the object's visual hull [19]. This will in turn lead to better novel views.

## Future work

Future work should include implementing the complete set of tests for determining the valid regions of the epipolar line segments when computing the object's visual hull in the image-based approach. This topic was discussed in section 5.1 and is necessary to ensure the error free calculation of the object's visual hull.

Further investigation into the visibility algorithm of the image-based technique should also be

conducted. Since the algorithm is only an approximation of the actual visibility [23] its limitations should be established. Methods to make the implementation of the algorithm more robust should also be explored.

The sensitivity of both the geometry-based and the image-based methods to segmentation and camera calibration errors should be determined. The reference images need to be segmented to obtain the silhouettes of the observed object and therefore inaccurate segmentation will lead to an inaccurate approximation of the object's visual hull. As mentioned before, faulty calibration parameters will result in errors in the novel view. Since the silhouettes and calibration parameters are provided as input to the algorithms these issues can be addressed separately. Alternatively, the algorithms can be made more robust to such problems.

# Bibliography


[1] S. Avidan and A. Shashua, "Novel View Synthesis by Cascading Trilinear Tensors," *IEEE Transactions on Visualisation and Computer Graphics*, Vol. 4, No. 4, October—December 1998.

[2] R. Azuma, "Recent Advances in Augmented Reality," *IEEE Computer Graphics and Applications*, November—December 2001.

[3] S. E. Chen and L. Williams, "View Interpolation for Image Synthesis," *Proceedings of the International Conference on Computer Graphics and Interactive Techniques*, pages 279–288, 1993.

[4] K. Connor and I. Reid, "Novel View Specification and Synthesis," *Proceedings of the British Machine Vision Conference*, pages 243–252, 2002.

[5] W. B. Culbertson and T. Malzbender, "Generalized Voxel Coloring," *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, Vol. 1883 of Lecture notes in Computer Science, pages 100–115, Springer-Verlag, 2000.

[6] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and Rendering Architecture from Photographs: a Hybrid Geometry- and Image-based Approach," *Proceedings of the International Conference on Computer Graphics and Interactive Techniques*, pages 11–20, 1996.

[7] P. Debevec, Y. Yu, and G. Borshukov, "Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping," *Proceedings of the Eurographics Rendering Workshop*, June 1998.

[8] C. R. Dyer, "Volumetric Scene Reconstruction from Multiple Views," In L.S. Davis, editor, *Foundations of Image Understanding*, pages 469–489. Kluwer, Boston, 2001.

[9] O. Faugeras, Q. Luong, *The Geometry of Multiple Images*, Cambridge, Massachusetts: The MIT Press, 2001.

[10] J. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley Publishing Company, Inc., 1990 with corrections 1997.

[11] K. Forbes, A. Voigt, and N. Bodika, "An inexpensive, automatic and accurate camera calibration method," *Proceedings of the Thirteenth Annual South African Workshop on Pattern Recognition*, PRASA, 2002.

[12] K. Forbes, A. Voigt, and N. Bodika, "Using Silhouette Consistency Constraints to Build 3D Models," *Proceedings of the Fourteenth Annual South African Workshop on Pattern Recognition*, PRASA, 2003.

[13] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Englewood Cliffs, N.J.: Prentice-Hall, August 2002.

[14] A. Fusiello, S. Caldrer, S. Ceglie, N. Mattern, and V. Murino, "View Synthesis from Uncalibrated Images using Parallax," *Proceedings of the 12th International Conference on Image Analysis and Processing*, pages 146–151, 2003.

[15] D. Hearn and M. P. Baker, *Computer Graphics*, Englewood Cliffs, N.J.: Prentice-Hall, 1986.

[16] M. Irani, P. Anandan, and D. Weinshall, "From Reference Frames to Reference Planes: Multi-View Parallax Geometry and Applications," *Proceedings of the European Conference on Computer Vision*, Vol. 2, pages 829–845, June 1998.

[17] M. Irani, T. Hassner, and P. Anandan, "What Does the Scene Look Like from a Scene Point?" *Proceedings of the European Conference on Computer Vision*, pages 883–897, 2002.

[18] S. B. Kang, "A Survey of Image-Based Rendering Techniques," *Technical Report CRL 97/4*, Cambridge Research Laboratory, August 1997.

[19] A. Laurentini, "The Visual Hull concept for Silhouette-Based Image Understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 2, February 1994.

[20] S. Laveau and O. Faugeras, "3-D Scene Representation as a collection of images," *Proceedings of the International Conference on Pattern Recognition*, Vol. 1, pages 689-691, 1994.

[21] W. E. Lorensen, and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics*, Vol. 21, No. 4, July 1987.

[22] M. Levoy and P. Hanrahan, "Light Field Rendering," *Proceedings of ACM SIGGRAPH '96*, 1996.

[23] W. Matusik, *Image-Based Visual Hulls*, Master of Science Dissertation, Massachusetts Institute of Technology, 2001.

[24] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan, "Image-Based Visual Hulls," *Proceedings of the International Conference on Computer Graphics and Interactive Techniques*, pages 369–374, 2000.

[25] L. McMillan and G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System," *Proceedings of SIGGRAPH'95*, August 1995.

[26] L. McMillan, *An Image-Based approach to Three-Dimensional Computer Graphics*, Ph.D. Thesis, University of North Carolina, 1997.

[27] M. Pollefeys, "Tutorial on 3D Modeling from Images," *Proceedings of the European Conference on Computer Vision*, June 2000.

[28] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: the Art of Scientific Computing*, Cambridge University Press, Cambridge, 1988.

[29] S. M. Seitz and C. R. Dyer, "Physically-Valid View Synthesis by Image Interpolation," *Proceedings of the IEEE Workshop on Representations of Visual Scenes*, pages 18–25, 1995.

[30] S. M. Seitz and C. R. Dyer, "Photorealistic Scene Reconstruction by Voxel Colouring," *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1067–1073, 1997.

[31] S. M. Seitz and C. R. Dyer, "View Morphing," *Proceedings of the International Conference on Computer Graphics and Interactive Techniques*, pages 21–30, 1996.

[32] H. Y. Shum and S. B. Kang, "A Review of Image-based Rendering Techniques," *IEEE/SPIE Visual Communications and Image Processing (VCIP) 2000*, pages 2–13, June 2000.

[33] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer, "A Survey of Methods for Volumetric Scene Reconstruction from Photographs," *International Workshop on Volume Graphics 2001*, June 2001.

[34] M. R. Stevens, B. Culbertson, and T. Malzbender, "A Histogram-based Color Consistency Test for Voxel Coloring," *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 4, pages 118–121, 2002.

[35] D. Sunday, *Distance between Lines and Segments with their Closest Point of Approach*, Available: http://softsurfer.com/Archive/algoritm_0106/algorithm_0106.htm, Accessed: 13 December 2004.

[36] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, N. J.: Prentice-Hall, 1998.

[37] R. Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 4, pages 323–344, 1987.

[38] S. Vedula, P. Rander, H. Saito, and T. Kanade, "Modeling, Combining, and Rendering Dynamic Real-World Events From Image Sequences," *Proceedings of the 4th Conference on Virtual Systems and MultiMedia*, Vol. 1, pages 326–332, 1998.

[39] T. Werner, R. D. Hersch, and V. Hlaváč, "Rendering Real-World Objects using View Interpolation," *Proceedings of the International Conference on Computer Vision*, 1995.

[40] K. K. Wong and R. Cipolla, "Reconstruction of sculpture from its profiles with unknown camera positions," *IEEE Transactions on Image Processing*, Vol. 13, No. 3, pages 381–389, 2004.