

IMAGE-BASED RENDERING TECHNIQUES FOR  
APPLICATION IN VIRTUAL ENVIRONMENTS

Xiaoyong Sun

A Thesis submitted to the Faculty of Graduate and Postdoctoral  
Studies in partial fulfillment of the requirements for the degree of  
Master of Applied Science, Electrical Engineering

July 2002

Ottawa-Carleton Institute for Electrical and Computer Engineering  
School of Information Technology and Engineering  
University of Ottawa  
Ottawa, Ontario, Canada

© Xiaoyong Sun, 2002

*To Jilian ...*

# Contents

<b>List of Figures</b>	<b>vi</b>
<b>Abstract</b>	<b>x</b>
<b>Acknowledgements</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Real-Image-Based Virtual Reality . . . . .	2
1.2 Related Work . . . . .	2
1.3 Thesis Orientation . . . . .	4
1.4 Thesis Outline . . . . .	5
<b>2 The Scene Representation Model: Plenoptic Function</b>	<b>7</b>
2.1 Panoramic Views and View Interpolation . . . . .	8
2.2 Light Field Modelling . . . . .	9
2.3 Stereoscopic View Rendering . . . . .	10
<b>3 Panoramic Views and View Interpolation</b>	<b>11</b>
3.1 Panoramic Views . . . . .	13
3.1.1 Image warping . . . . .	13
3.1.2 Image alignment and registration . . . . .	17
3.1.3 Stitching two images together . . . . .	18
3.1.4 Clipping the first and the last images . . . . .	18

3.2	View Interpolation . . . . .	21
3.2.1	Pre-warping: rectification and scaling . . . . .	22
3.2.2	Interpolation along the row direction . . . . .	25
3.2.3	Post-warping . . . . .	25
3.2.4	Simulation results . . . . .	26
3.3	Conclusions . . . . .	26
<b>4</b>	<b>Light Field Modelling</b>	<b>32</b>
4.1	The Light Field Rendering and the Concentric Mosaics Technique . .	33
4.1.1	The Light Field Rendering technique . . . . .	33
4.1.2	The Concentric Mosaics technique . . . . .	37
4.1.3	Comparison of the two techniques . . . . .	41
4.2	The Data Structure in the Concentric Mosaics Technique . . . . .	45
4.3	Rendering with the Concentric Mosaics Technique . . . . .	48
4.3.1	Determining a view: border conditions . . . . .	49
4.3.2	Determining an arbitrary light ray . . . . .	51
4.3.3	Interpolation methods . . . . .	52
4.3.4	Simulation results and observations . . . . .	59
4.4	Design Considerations in the Capture of Concentric Mosaics Data . .	62
4.4.1	Considerations from sampling rate . . . . .	64
4.4.2	Considerations from the number of samples . . . . .	65
4.4.3	The frequency domain interpretation for our analysis . . . . .	66
4.4.4	Simulations . . . . .	67
<b>5</b>	<b>Rendering of Stereo Views in the Concentric Mosaics Technique</b>	<b>71</b>
5.1	The Visualization of Stereo Views on a Monitor . . . . .	73
5.1.1	Visualization of the input stereo pair . . . . .	73
5.1.2	The viewing and rendering of stereo views . . . . .	76
5.2	The Anaglyph Technique . . . . .	78
5.2.1	Visualization of an anaglyph image . . . . .	78

5.2.2	Optimization problem with projection solution . . . . .	81
5.2.3	Simulation of color recovery and intensity disparity of the left and right views . . . . .	84
5.2.4	Simulation results on generating anaglyph images . . . . .	89
5.3	The Fast Rendering of Anaglyph Views in the Concentric Mosaics Technique . . . . .	90
5.3.1	The distance changing between left light rays and right light rays . . . . .	94
5.3.2	Simulation results and conclusions . . . . .	96
<b>6</b>	<b>Conclusions and Future Work</b>	<b>99</b>
6.1	Summary of the Thesis . . . . .	99
6.2	Thesis Contributions . . . . .	101
6.3	Future Work . . . . .	101
	<b>Bibliography</b>	<b>103</b>

# List of Figures

2.1	A light ray in free space . . . . .	9
3.1	Nodes distributed in the navigation area. The irregular navigation path is approximated by the straight lines between nodes. . . . .	12
3.2	The algorithm for generation of panoramic views . . . . .	14
3.3	The coordinate relationship for warping an image onto a cylindrical surface . . . . .	15
3.4	The images before warping (left) and after warping onto a cylindrical surface (right) . . . . .	16
3.5	The illustration of Lucas-Kanade registration algorithm in one dimension	17
3.6	Illustration of the stitching algorithm . . . . .	19
3.7	Panoramic view of the VIVA lab . . . . .	20
3.8	Flow chart of the proposed algorithm for view interpolation . . . . .	21
3.9	One original image before rectification (the white lines are epipolar lines, which are drawn by a program from Etienne Vincent) . . . . .	27
3.10	The image in Figure 3.9 after rectification (the white lines are epipolar lines that are nearly horizontal) . . . . .	28
3.11	The other original image before rectification . . . . .	28
3.12	The image in Figure 3.11 after rectification and scaling . . . . .	29
3.13	The intermediate view interpolated from the images in Figure 3.10 and 3.12 . . . . .	29
3.14	Intermediate view between Figure 3.9 and 3.11 after post-warping . .	30

4.1	2-plane parameterization of light rays for Light Field Rendering . . .	34
4.2	The Light Field Rendering camera gantry (Stanford University) [1] .	35
4.3	An example of the pre-captured image samples for Light Field Rendering [1] . . . . .	36
4.4	Some light rays for an arbitrary position $P$ . . . . .	37
4.5	The illustration of capture procedure for Concentric MosaiCs technique	38
4.6	The Concentric MosaiCs capture device (Microsoft Research)[2] . . . .	39
4.7	The capturing and rendering procedure of the Concentric MosaiCs technique . . . . .	40
4.8	The illustration of depth distortion . . . . .	41
4.9	A rendered image using the Light Field Rendering technique . . . . .	43
4.10	A rendered image using the Concentric MosaiCs technique . . . . .	44
4.11	A sampled ray (condensed light ray) in two-dimensions . . . . .	46
4.12	The panoramic view with (a) $\beta \approx 0$ and (b) $\beta = \frac{\delta_c}{2}$ . . . . .	47
4.13	Illustration of the non-uniform sampling in the angular $\beta$ direction . .	48
4.14	The nonlinear relationship between $\beta$ and $\Delta\beta$ . . . . .	49
4.15	Geometric considerations for rendering with Concentric MosaiCs (Note that the angles are exaggerated for the purpose of the illustration. $\rho$ is the distance from $O$ to $P$ , $\theta$ is the angle between $OP$ and $X$ axis) .	50
4.16	The interpolation in the rendering algorithm . . . . .	52
4.17	Nearest Point Approximation and Infinite Depth Assumption Interpolation (Note that the angles are exaggerated for the purpose of the illustration) . . . . .	54
4.18	Linear interpolation with constant depth assumption (Note that the angles are exaggerated for the purpose of the illustration) . . . . .	58
4.19	Rendering with nearest sampled rays method . . . . .	60
4.20	Rendering through linear interpolation with infinite depth assumption	60
4.21	Rendering through linear interpolation with constant depth assumption	61
4.22	The sampling model of Concentric MosaiCs . . . . .	62

4.23	The frequency domain interpolation . . . . .	66
4.24	The minimum number $N_R$ of image samples at different relative lengths $R/R_{\text{MIN}}$ of the rotation beam . . . . .	67
4.25	Down-sampling factor $t=1$ (from original data set) . . . . .	68
4.26	Down-sampling factor $t=2$ . . . . .	69
4.27	Down-sampling factor $t=3$ . . . . .	69
5.1	Visualization of the stereo images . . . . .	73
5.2	Visualization of the anaglyph images . . . . .	78
5.3	Transmission of a pair of commercial anaglyph glasses as a function of wavelength . . . . .	85
5.4	The study of color recovery of the anaglyph images (The solid line represents the $x$ chromaticity coordinates, the dashed line represents the $y$ chromaticity coordinates of the generated anaglyph images in the XYZ colorimetric system and the dotted line represents the white coordinates with $x, y = 0.3333$ for reference white.) . . . . .	86
5.5	The study of intensity disparity of the left and right views (solid line represents the luminance of left views and dashed line represents the luminance of right views) . . . . .	87
5.6	The study of color recovery of the final views (The solid line repre- sents the $x$ chromaticity coordinates, the dashed line represents the $y$ chromaticity coordinates of the finally perceived views in the XYZ col- orimetric system and the dotted line represents the white coordinates with $x, y = 0.3333$ for reference white.) . . . . .	88
5.7	The left view (reduced size) . . . . .	89
5.8	The right view (reduced size) . . . . .	90
5.9	The anaglyph image with $d=1$ . . . . .	91
5.10	The anaglyph image with $d=0.88$ . . . . .	92
5.11	The anaglyph image with $d=0.76$ . . . . .	93



5.12	The analysis on the viewing distance between two eyes (The distance between two eye $E_R$ and $E_L$ is d.) . . . . .	95
5.13	A rendered anaglyph image with the proposed fast algorithm . . . . .	97
5.14	A rendered anaglyph image with the usual approach . . . . .	98

# Abstract

In this thesis, the methods of Image-Based Rendering for creating virtual environment applications are studied in order to construct a real image-based virtual environment with the principle of representing an environment through a set of pre-captured images. These pre-captured images are used to synthesize arbitrary views in a virtual environment. Currently developed techniques include view synthesis through interpolation, Light Field Rendering and Concentric Mosaics. These methods are presented and compared in the thesis and we conclude that the Concentric Mosaics technique is more suitable for practical applications. The stereoscopic view synthesis through the Concentric Mosaics rendering technique is also addressed. The application of the anaglyph technique can make stereo views of a virtual environment available to any personal computer user with an inexpensive pair of colored glasses.

# Acknowledgements

First, I would like to thank my supervisor Dr. Eric Dubois for giving me this chance to work with him. I did benefit not only from the helpful directions at all times whenever I needed them, but am still benefiting from his character which leads me on to become an excellent researcher like him.

I also want to thank my wife and my parents. Their support over the years and their encouragement has made me self-confident to face life's challenges as they come along. Also thanks to the help from all my friends, some of who are located in areas with maximal time difference from mine. I also appreciate the help from my colleagues in the VIVA lab. Among them, the talks on the epipolar geometry with Etienne have been very helpful to me.

Also I must thank Microsoft Research for the generosity of providing me the Concentric Mosaics data to carry out this research work and the National Capital Institute of Telecommunications, Canada which provided the funding for this research work.

# Chapter 1

## Introduction

Virtual reality techniques are becoming more and more important as increasing computing power and network bandwidth allow the ordinary personal computer user to navigate in a virtual environment, even remotely. It can provide the user a better experience in applications such as e-commerce, teleconference, virtual museum visiting, new worker training, etc.

Traditionally, virtual environments are constructed from 3D geometric entities. An arbitrary view can be rendered by projecting 3D geometric entities toward a specified viewing direction with the help of special purpose 3D rendering engines. However, at least two intrinsic problems exist in this traditional geometry-based technique for image synthesis:

(i) The creation of the 3D geometric entities is a laborious manual process and it is very difficult or even impossible to model some complex objects or environments using regular elementary geometric entities.

(ii) In order to run in real time, the rendering engine has to place a limit on the scene complexity and rendering quality. Even so, special purpose 3D rendering accelerators are usually required to speed up the rendering procedure, which are by no means standard equipment that is widely available for personal computer users.

## 1.1 Real-Image-Based Virtual Reality

Since there is really no upper bound on scene complexity, it is very difficult and human-intensive to model the real world with high fidelity. A currently developed technique which is a powerful alternative, namely Image-Based Rendering [3], has attracted much attention for image synthesis. Unlike the computer graphics based methods, a collection of pre-captured sample images is used to render a novel view, instead of projecting from geometric models. The advantages of Image-Based Rendering methods over computer graphics based methods are that real images of a scene are used and the method is the same and standard for any scenes, whether complex or not.

## 1.2 Related Work

Depending on whether geometric information is used and what kind of geometric information is required, the Image-Based Rendering techniques are classified into three categories: rendering with explicit geometry, rendering with implicit geometry and rendering without geometric information [4].

Transfer methods [4], a term used in the photogrammetric community, involve the general idea of rendering with geometric information, either implicit or explicit. A novel view is rendered by reprojecting image pixels appropriately at a given virtual camera viewpoint from a relatively small number of pre-captured reference images using some geometric constraints, such as depth values at each pixel, epipolar constraints between image pairs, or the trilinear tensors between triplets of images.

The explicit geometry is the depth information of an environment or an object. With the knowledge of depth distribution, a spatial geometric model of a scene or object can be constructed, based on which novel views can be projected through texture mapping. The approach has evolved from computer graphics, the idea of which is similar to the procedures of reconstruction and re-projection through the camera pose calibration. The methods rely on accurate geometric models, in the

form of a set of pre-captured images associated with the depth maps of the scenes. The methods developed in this category include view-dependent texture mapping [5], 3D warping [6], layered-depth images (LDI) [7] and LDI tree [8], etc.

In the methods of rendering with implicit geometry, feature correspondence information between images is required and can be obtained through computer vision techniques. The methods include view interpolation, view morphing etc. The geometric constraints are represented using the trilinear tensor [9] or the fundamental matrix [10], depending on the number of reference images.

The trilinear tensor is computed from the feature correspondences across three reference images, while in the case where only two reference images are available, the third image can be regarded as identical with any one of the two reference images. With the trilinear tensor and camera intrinsic parameters, a new trilinear tensor can be obtained when the pose of a virtual camera changes. Then the new view can be generated by mapping point correspondences in the reference images to their new positions through the trilinear tensor [9].

With two reference images, the geometric constraints can also be represented by the fundamental matrix, which is used in view interpolation and view morphing. If the dense optical flow is known, arbitrary views can be constructed from two input images by the view interpolation methods proposed by [11].

Although the flow field can be obtained from the depth values which are known for synthetic images, it is difficult or even impossible to establish flow fields for real images. The view-morphing technique performs view interpolation by extracting the camera pose information through the fundamental matrix. A scan-line-based view interpolation approach has been proposed by [12], which simplifies the two-dimensional problem to one-dimension using image rectification.

In the methods of rendering without geometric information, a set of images is pre-captured and the rendering is the procedure of reconstruction of pixels (or other points among the pixels) from the pre-captured images. Intensity and colors are interpolated if the position does not correspond exactly to a pixel in the pre-captured image. The

methods include the Light Field Rendering technique [13] and the Concentric Mosaics technique [14].

For example, a video camera is mounted on one end of a rotation beam in the Concentric Mosaics technique. As the beam rotates around the pivot point, the video camera which is pointing outwards along the beam direction takes pictures. A set of Concentric Mosaics data is captured after the beam rotates one complete circle. Then any arbitrary view within a certain navigation area can be rendered based on this set of data.

### 1.3 Thesis Orientation

Although a range finder can provide the depth information in an environment, the depth information map is neither convenient to obtain (as opposite to taking a picture using a camera), nor precise if the depth variation is large, such as in a complex environment. Thus the applications of the Image-Based Rendering methods with explicit geometric information will be limited by the required precise 3D geometric models and we will not study those methods here. Our work will focus on the Image-Based Rendering methods both with implicit geometric information and without geometric information.

In the view interpolation method, one of the key issues is the pre-warping, or image rectification. Previous work for rectification is either based on a simplification assuming orthogonal projections, which is not a good approximation in practice, or the procedure is complex and based on the assumption that the views are singular views, i.e., the epipoles are not within the field of view. A simple and efficient method based on epipolar geometry will be used in our algorithm [15].

The Concentric Mosaics technique is a practically useful one among the Image-Based Rendering methods without geometric information. Given a specified scene, how to determine the length of the rotation beam and the rotation velocity is an important issue in the design considerations. However, previous work on the Concentric

Mosaics technique did not address this aspect. The problem is studied in our work [16] based on the scene sampling theory [17].

Stereoscopic views may be more attractive than monoscopic ones, especially in the application of navigating in a virtual environment. The previous work [18] on the stereo rendering of the Concentric Mosaics is based on the shutter glasses and a screen division technique to separate the left and right views. In our implementation for the stereo rendering algorithm, both the shutter-glasses method without screen division and the color-glasses method (anaglyph) are used. In particular, a fast rendering algorithm for the Concentric Mosaics technique based on the anaglyph technique is proposed [19] which provides an opportunity for the personal-computer user to navigate in a stereoscopic virtual environment.

## 1.4 Thesis Outline

The thesis begins by introducing the mathematic model for the scene representation, or plenoptic function [20] in Chapter 2.

Chapter 3 presents one scheme to construct a virtual environment using Image-Based Rendering with implicit geometric information. The method is the combination of the panorama technique (image mosaics) and the view interpolation technique, the idea behind QuickTime VR (virtual reality) [3].

Chapter 4 discusses the methods of Image-Based Rendering without geometric information, which is light field modelling. Both the Light Field Rendering technique and the Concentric Mosaics will be introduced and the comparison will be made. The design considerations of the Concentric Mosaics technique will be emphasized.

Chapter 5 considers the problem of rendering stereoscopic views with the Image-Based Rendering technique, which is related to the methods for viewing stereoscopic views on a monitor. The most advanced and convenient technique via a pair of shutter glasses, which requires an expensive system, and the cheapest method via a pair of colored glasses, or anaglyph technique [21], have both been described. A fast stereo



rendering algorithm based on the combination of the anaglyph technique and the Concentric Mosaics technique is also proposed.

Our conclusions and the future work follow in Chapter 6.

## Chapter 2

# The Scene Representation Model: Plenoptic Function

At a specific time and location, an idealized eye sees a two-dimensional picture. This two-dimensional picture is constructed from all light rays entering the eye, which are passing through the center of the pupil at every possible angle  $(\theta, \phi)$ . The entire set of light rays that can be perceived at every possible location  $(V_x, V_y, V_z)$  and every time  $t$  can be represented by a seven-dimensional function, if each light ray is decomposed into different wavelengths  $\lambda$ , as

$$P = P(\theta, \phi, \lambda, t, V_x, V_y, V_z). \quad (2.1)$$

The seven-dimensional plenoptic function can be reduced to six dimensions by ignoring the time variable, which is appropriate for static environments. The plenoptic function can further be reduced to five dimensions by eliminating the wavelength variable. However, each light ray will consist three components for RGB representation of a color view. Thus, we use vector  $\mathbf{P}$  to replace  $P$  for RGB color representation of light rays.

$$\mathbf{P} = \mathbf{P}(\theta, \phi, V_x, V_y, V_z). \quad (2.2)$$

Although it is difficult and even impossible to capture all the light rays within a certain spatial area, the plenoptic function does provide a mathematic model of the

scene to be represented.

## 2.1 Panoramic Views and View Interpolation

The panoramic view is the collection of all light rays toward one specified position  $\mathbf{V}_0(V_{0x}, V_{0y}, V_{0z})$ , or

$$\Psi_0 = \{\mathbf{P} = \mathbf{P}(\theta, \phi, V_x, V_y, V_z) | \theta \in \Theta, \phi \in \Phi, V_x = V_{0x}, V_y = V_{0y}, V_z = V_{0z}\} \quad (2.3)$$

where parameters  $\Theta$  and  $\Phi$  determine the range of viewing directions. For the spherical and cylindrical panoramas, different viewing direction ranges are used.

The in-between view  $\Psi_i$  can be synthesized by two adjacent views  $\Psi_1$  and  $\Psi_2$  as,

$$\Psi_i = \{\mathbf{P}_i = ((i-1) \otimes \mathbf{P}_1) \oplus ((2-i) \otimes \mathbf{P}_2) | \mathbf{P}_1 \in \Psi_1, \mathbf{P}_2 \in \Psi_2\} \quad (2.4)$$

where the  $\oplus$  operation denotes the interpolation procedure and the  $\otimes$  operation denotes applying different weights to different views. Here,

$$\Psi_1 = \{\mathbf{P}_1 = \mathbf{P}(\theta, \phi, V_x, V_y, V_z) | \theta \in \Theta_1, \phi \in \Phi_1, V_x = V_{1x}, V_y = V_{1y}, V_z = V_{1z}\} \quad (2.5)$$

$$\Psi_2 = \{\mathbf{P}_2 = \mathbf{P}(\theta, \phi, V_x, V_y, V_z) | \theta \in \Theta_2, \phi \in \Phi_2, V_x = V_{2x}, V_y = V_{2y}, V_z = V_{2z}\} \quad (2.6)$$

with  $1 < i < 2$  and

$$V_{ix} \in [V_{1x}, V_{2x}] = \begin{cases} \{V | V_{1x} < V < V_{2x}\} & \text{if } V_{1x} < V_{2x} \\ \{V | V_{1x} > V > V_{2x}\} & \text{if } V_{1x} \geq V_{2x} \end{cases} \quad (2.7)$$

$$V_{iy} \in [V_{1y}, V_{2y}] = \begin{cases} \{V | V_{1y} < V < V_{2y}\} & \text{if } V_{1y} < V_{2y} \\ \{V | V_{1y} > V > V_{2y}\} & \text{if } V_{1y} \geq V_{2y} \end{cases} \quad (2.8)$$

$$V_{iz} \in [V_{1z}, V_{2z}] = \begin{cases} \{V | V_{1z} < V < V_{2z}\} & \text{if } V_{1z} < V_{2z} \\ \{V | V_{1z} > V > V_{2z}\} & \text{if } V_{1z} \geq V_{2z} \end{cases} \quad (2.9)$$

Thus, fewer images are required due to the use of interpolation. Further details will be presented in Chapter 3.

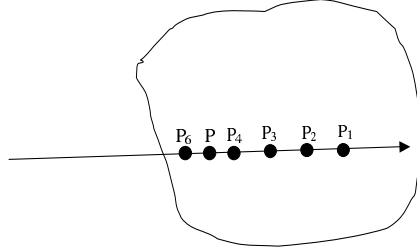


Figure 2.1: A light ray in free space

## 2.2 Light Field Modelling

It is impossible to pre-capture all light rays in the plenoptic function and it is also not necessary to do so. This is the idea behind the methods of rendering without geometric information. Based on the assumption that the light rays do not change along their locus of propagation, the light field modelling techniques aim at using fewer but sufficient light rays to represent all the light rays in the plenoptic function. As shown in Fig. 2.1, one light ray passes through  $P_1$ ,  $P_2$ ,  $P_3$ ,  $P_4$ ,  $P_5$ , and  $P_6$ . Thus, instead of using six light rays, one light ray is enough to represent all six light rays toward these six positions and certainly more along its propagation trace. Thus the techniques of light field modelling use a set of pre-captured images as the representative light rays. The rendering of any arbitrary view is the procedure of recombining the properly selected light rays for a specific location and view direction. The techniques include Light Field Rendering [13], Lumigraph [22], Concentric Mosaics [14], Panoramas [23], etc.

The key problem in the light field modelling techniques is how to record the representatives of all possible light rays by means of pre-captured images and how to efficiently index each light ray. In the point of view of the plenoptic function, a well-indexed subset of light rays  $\Pi_1$  needs to be found to represent the set of all light rays  $\Pi$  as,

$$\Pi_1 \subseteq \Pi \tag{2.10}$$

with

$$\mathbf{\Pi} = \{P = P(\theta, \phi, V_x, V_y, V_z) | \theta \in \Theta, \phi \in \Phi, (V_x, V_y, V_z) \in \mathbf{V}_{xyz}\} \quad (2.11)$$

where parameter  $\Theta$  and  $\Phi$  determine the range of viewing directions, and  $\mathbf{V}_{xyz}$  denotes the spatial navigation area.

The Light Field Rendering technique and the Concentric Mosaics technique are two methods to obtain different subsets of the plenoptic function as the representative light rays to render any arbitrary view.

## 2.3 Stereoscopic View Rendering

Generating a stereoscopic view is convenient in the Image-Based Rendering. The views of the two eyes can be generated based on the same plenoptic function using the same algorithm. The stereoscopic view  $\mathbf{\Psi}_s$  is constructed by two views: the left view  $\mathbf{\Psi}_l$  and the right view  $\mathbf{\Psi}_r$ , with

$$\mathbf{\Psi}_s = \{\mathbf{\Psi}_l, \mathbf{\Psi}_r\} \quad (2.12)$$

where subscript  $s$  denotes ‘stereo’,  $l$  denotes ‘left’ and  $r$  denotes ‘right’. Here,

$$\mathbf{\Psi}_s = \{\mathbf{P}(\theta, \phi, V_{rx}, V_{ry}, V_{rz}), \mathbf{P}(\theta, \phi, V_{lx}, V_{ly}, V_{lz}) | \theta \in \Theta, \phi \in \Phi, \sqrt{(V_{lx} - V_{rx})^2 + (V_{ly} - V_{ry})^2 + (V_{lz} - V_{rz})^2} = d\} \quad (2.13)$$

where,  $d$  is the distance between the two eyes.

# Chapter 3

## Panoramic Views and View Interpolation

The first attempt for the application of navigating in a virtual environment was proposed as the technique of QuickTime VR [3]. Regular nodes are distributed in the navigation area, as shown in Fig. 3.1. The navigation path can be approximated by the straight lines connecting a set of nearby nodes, as in the example shown in the figure. With panoramic images constructed at each node, arbitrary views in any viewing direction can be generated. The same philosophy has also been proposed by McMillan and Bishop [20]: the representation of a 5D light field as a set of panoramic images (2D) at different 3D locations for navigating in the 3D space.

It is a technical challenge to capture images precisely at a very dense set of locations within a certain navigation area. On the other hand, the images at some locations can be synthesized from the adjacent images by the various techniques of view synthesis. Thus two techniques are essential for the construction of the above mentioned virtual environment:

- i) the generation of panoramic views;
- ii) the view synthesis.

In this chapter, we will first study the algorithm for generating panoramic views by stitching together a set of overlapping images of the environment from different

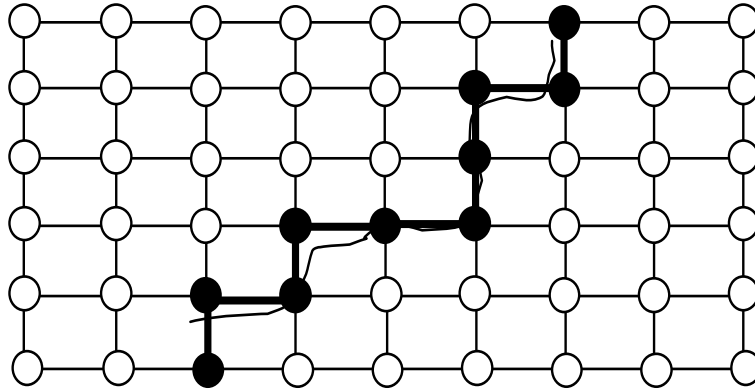


Figure 3.1: Nodes distributed in the navigation area. The irregular navigation path is approximated by the straight lines between nodes.

viewing directions at a fixed location and then we will focus on the study of view synthesis, or view interpolation.

Compared with the generation of panoramas, the algorithms for view synthesis are complex and based on different principles [11] [24]. In this chapter, we will focus on the in-between view interpolation using two captured images. This requires establishing a correspondence relationship between almost every pixel in the two images for the interpolation, which is a complex two-dimensional problem that is impossible to solve precisely at present.

The problem can be simplified by transforming the interpolation into one dimension [12]. We first perform a pre-warping operation to the images by applying image rectification using epipolar geometry in order to transform the two-dimensional interpolation problem into a one-dimensional interpolation along the scan-line direction. After interpolation, the interpolated images are converted back to the normal imaging condition through post-warping.

Previous methods for image rectification are either based on the camera calibration, with affine transformation following the rotations of image in depth, or assuming the projections are orthogonal for simplification, which is not a good approximation

in practice. A simple and efficient method based on epipolar geometry will be studied in this chapter [15]. The method for synthesis of intermediate views includes the construction of the fundamental matrix based on corresponding points (or correspondences in computer vision) in the two existing views, computation of the pre-warping and the post-warping matrices, and scaling of the images based on the common features in the two images.

For image rectification, there are eight degrees of freedom with two constraints. Proper choice of the remaining six constraints, or three points in the image pairs, is essential. The post-warping matrix is specified through the movement of the epipole.

The method makes computation of the transform matrix simple and stable. Simulation results are presented to illustrate the performance of the proposed algorithm.

## 3.1 Panoramic Views

A number of techniques, such as recording an image onto a long film using a panoramic camera, or a lens with very large field of view, mirrored pyramids and parabolic mirrors, have been traditionally developed to capture panoramic images.

The image mosaic technique [25] [23] is a new and less hardware-intensive method for constructing full-view panoramas by first capturing a set of regular overlapped photographic or video images from different view directions. Images are projected into a common space first before aligning captured images together because of the non-uniform sampling property of the camera. This is the essential idea for image warping, which will be further discussed later. All these images are then aligned and stitched into panoramas. The overall procedure can be seen in Fig. 3.2.

### 3.1.1 Image warping

In order to stitch the images obtained from different views, they must be projected onto a common surface first. This can be implemented by the technique of image warping [26], or essentially texture mapping [27].



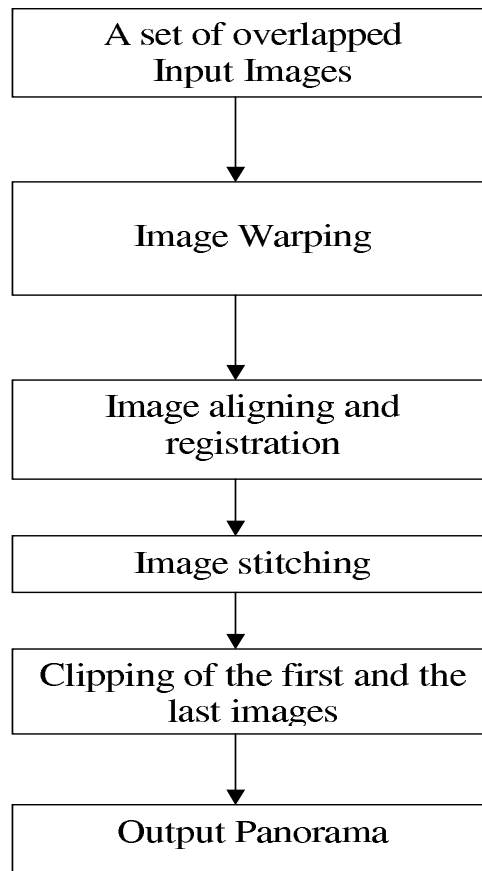


Figure 3.2: The algorithm for generation of panoramic views

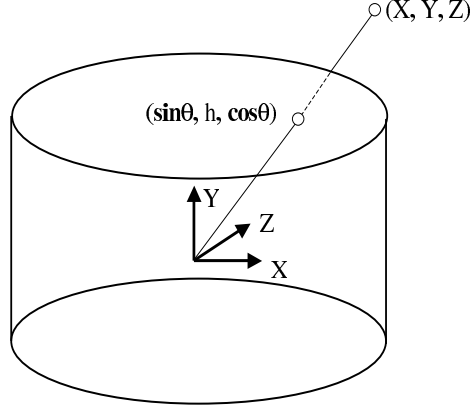


Figure 3.3: The coordinate relationship for warping an image onto a cylindrical surface

As opposed to image filtering, which changes the intensity of the image as shown in Equation 3.1, image warping changes the domain of the image as shown in Equation 3.2.

$$g_f(x) = h_1(f(x)) \quad (3.1)$$

$$g_w(x) = f(h_2(x)) \quad (3.2)$$

where  $f(x)$ ,  $g_f(x)$ , and  $g_w(x)$  are the original input image, the image after filtering and the image after warping.  $h_1(x)$  and  $h_2(x)$  denote two different functions. Thus projecting the image onto different kinds of surfaces by changing its original sampling lattice is the essential idea of image warping, or texture mapping.

The traditional texture mapping has been studied in computer graphics, which includes translation, rotation, similarity, affine and perspective in homogeneous coordinate system, etc. [27]. A cylindrical surface is used in our application. The images are taken by a camera mounted on a levelled tripod. Under this imaging condition, no rotation around the camera optical axis is involved. A pure translation model largely reduces our work in the image warping and image registration.

Assume that the original image is  $I_O(x_O, y_O)$  and that  $I_W(x_W, y_W)$  represents the resulting image after warping  $I_O(x_O, y_O)$  onto a cylindrical surface. Then the relationship between pixel positions in  $I_O(x_O, y_O)$  and  $I_W(x_W, y_W)$  is (see Fig. 3.3)



Figure 3.4: The images before warping (left) and after warping onto a cylindrical surface (right)

[2],

$$\theta = \frac{x_O - x_{OC}}{f} \quad (3.3)$$

$$h = \frac{y_O - y_{OC}}{f} \quad (3.4)$$

$$\hat{x} = \sin(\theta) \quad (3.5)$$

$$\hat{y} = h \quad (3.6)$$

$$\hat{z} = \cos(\theta) \quad (3.7)$$

$$x_W = f \cdot \frac{\hat{x}}{\hat{z}} + x_{OC} \quad (3.8)$$

$$y_W = f \cdot \frac{\hat{y}}{\hat{z}} + y_{OC} \quad (3.9)$$

where  $(x_{OC}, y_{OC})$  is the center position of image  $I_O(x_O, y_O)$ . The focal length  $f$  of the camera is in the units of number of pixels for the calculation. One of the warping results is illustrated in Fig. 3.4.

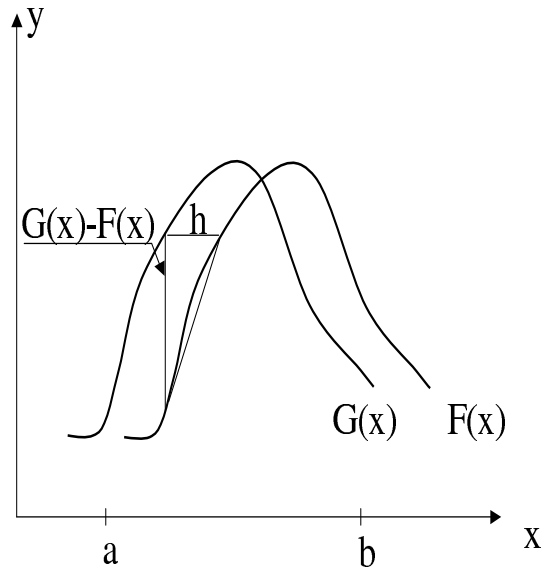


Figure 3.5: The illustration of Lucas-Kanade registration algorithm in one dimension

### 3.1.2 Image alignment and registration

Image alignment and registration is the process of aligning the adjacent images with overlapping area, and determining the related displacements with each other by the process of image registration. Image registration has been studied in many image-processing applications. The basic idea is to search for the common parts between two images. There are many algorithms proposed, such as Iterative Image Registration [28], Hierarchical Spline-Based Image Registration [25], and others.

The Iterative Image Registration technique proposed by Lucas and Kanade [28] is used here. The philosophy of this scheme is illustrated in Fig. 3.5 in the one-dimensional case for simplification. The extension to a two-dimensional iterative formula for the image processing application can easily be determined. The horizontal disparity between two curves  $F(x)$  and  $G(x) = F(x + h)$  can be calculated in an iterative way [28]

$$h_0 = 0, \tag{3.10}$$

$$h_{k+1} = h_k + \frac{\sum_x w(x)F'(x + h_k)[G(x) - F(x + h_k)]}{\sum_x w(x)F'(x + h_k)^2} \quad (3.11)$$

where

$$w(x) = \frac{1}{G'(x) - F'(x)}. \quad (3.12)$$

A hierarchical structure for the image registration at different layers is also used in our algorithm for fast searching, which means that we downsample images to different sizes in different layers and then the disparities  $h$  can be searched at different layers. In this way, the registration processing is speeded up with the “coarse-to-fine” approach. We chose three layers in our implementation.

### 3.1.3 Stitching two images together

After determining the relative displacement of two images, we can employ two different masks to modify the grey levels of the individual images and then add them together. This is the process of stitching. The masks can be of various styles, and the overlapped area of the masks might be different from the actual overlapped area of the two images. Fig. 3.6 is an example of stitching two overlapped images together, with the masks’ curves shown in the figure.

### 3.1.4 Clipping the first and the last images

After stitching the captured images one by one, the panorama has almost been obtained except for some overlapped area between the left part of the first image and the right part of the last image. Thus, the overlapped area should be cut off from one of them, after determining this area through image registration.

One panorama of the VIVA Lab, School of Information Technology and Engineering, University of Ottawa has been created using the algorithm described above as shown in Fig. 3.7.

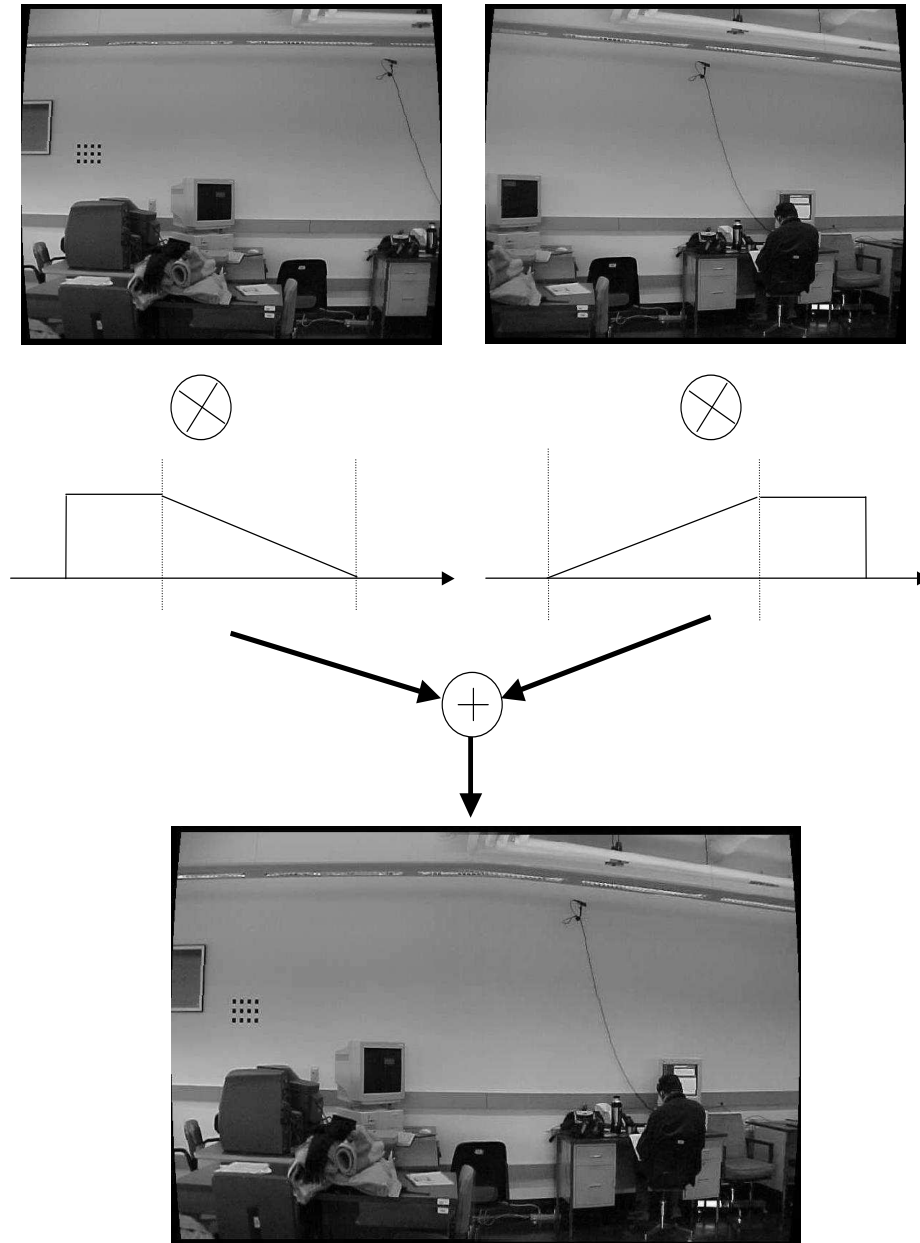


Figure 3.6: Illustration of the stitching algorithm

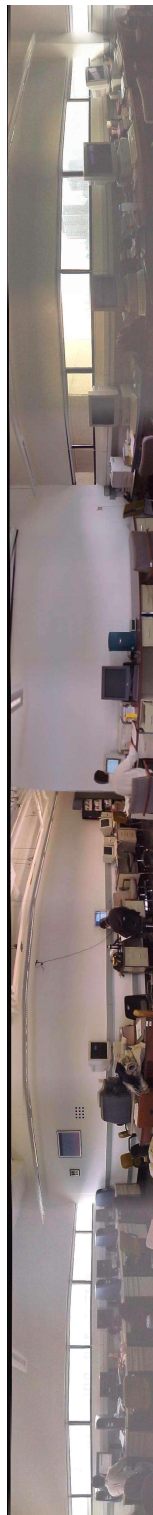


Figure 3.7: Panoramic view of the VIVA lab

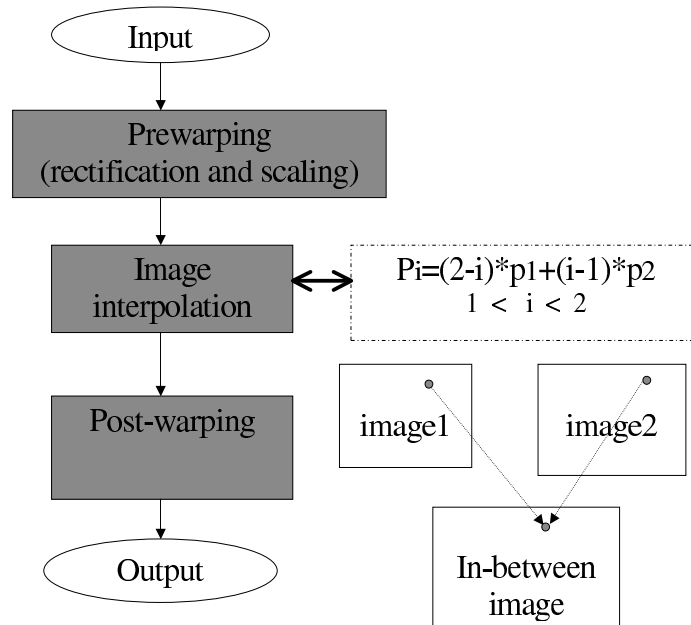


Figure 3.8: Flow chart of the proposed algorithm for view interpolation

## 3.2 View Interpolation

The idea of image interpolation is very simple, and can be expressed as:

$$P_i = (2 - i)P_1 + (i - 1)P_2 \quad (3.13)$$

where  $P_1$ ,  $P_2$  and  $P_i$  are the positions of correspondent points in images  $I_1$ ,  $I_2$  and  $I_i$ , respectively.  $I_i$  is an intermediate image between  $I_1$  and  $I_2$ , with  $1 < i < 2$ .

However, a fundamental difficulty in computer vision is to determine the correspondences in two images from different viewing directions. Thus it is impossible to directly implement equation (3.13). A scan-line-based interpolation algorithm is proposed [12], with its framework shown in Fig. 3.8. The pre-processing, or pre-warping, is image rectification and scaling. After pre-warping, corresponding points in the two rectified images are located on the same scan-line. Thus the position interpolation of the correspondences can be carried out in one dimension, which is easy to perform. Rectification is a technique in computer vision based on the epipolar geometry.



Obviously, rectification is the most essential part of the above algorithm and the method for the rectification transformation is not unique for a pair of images. In [24], the rectification is based on the assumption of orthographic views. The condition is achieved in practice by using a lens with a large focal length, which is not valid in the general case. The rectification method employed in [12] is oriented to the general imaging conditions. However, the method applies a rotation in depth to make the image planes parallel first and this is then followed by an affine transformation. The method is very complex and it works only on the condition that the views are not singular, which means that the epipoles must be outside the image borders and therefore not within the field of view. Thus, a more efficient and simple method will be used in our implementation, which is both robust and simple.

### 3.2.1 Pre-warping: rectification and scaling

#### Image morphing

Image morphing, similar to warping, is a technique used in computer vision. By changing the sampling lattice of the original image, the image from a different viewing direction can be obtained. The warping function is instantiated by a warping matrix acting on the original regular sampling lattice to form a new sampling structure. Then, by mapping the intensity and color of the correspondent pixels to their new position, the novel view is synthesized.

Rectification is essentially a morphing process, which is implemented in the homogeneous coordinate system. The homogeneous transformation of rectification is a  $3 \times 3$  matrix with 8 degrees of freedom (there are 9 elements with the common scale not significant, so only 8 degrees of freedom.)

#### Epipolar geometry and fundamental matrix

Suppose that there are two images of a common scene and  $\mathbf{u}$  is a point in the first image. If we use epipolar geometry to describe the imaging relationship between

these two images, the matching point  $\mathbf{u}'$  in the second image must lie on a specific line called the epipolar line corresponding to  $\mathbf{u}$  [10]. The epipolar lines in the second image corresponding to all points  $\mathbf{u}$  in the first image meet in a point  $\mathbf{p}'$ , which is called the epipole. Similarly, the epipole  $\mathbf{p}$  in the first image can be determined with the same principle.

Using the epipolar geometry [10], the relationship between a pair of images from different viewing directions can be described efficiently by the fundamental matrix without the reconstruction of the camera position. The  $3 \times 3$ , rank 2 fundamental matrix  $F$  of two images  $I_1$  and  $I_2$  satisfies the following relation,

$$p_2^T F p_1 = 0 \tag{3.14}$$

for any pair of corresponding points  $p_1$  and  $p_2$  located in  $I_1$  and  $I_2$  respectively. From the fundamental matrix, it is easy to calculate the positions of the epipole of each image, which is the intersection of all epipolar lines in the image. The epipole  $e$  of one view satisfies the following equation,

$$F e = 0 \tag{3.15}$$

with a similar expression for the other. Precisely locating correspondences between images and determining the exact fundamental matrix can be interlaced with each other in a refinement process.

The public domain software made available by Roth [29] was used in our project with good resulting fundamental matrix and set of correspondences.

## Rectification

The rectification method we used here is from [30], the philosophy of which is very straightforward: we map epipoles of both images to infinity to get the transformation matrix. The key issue is to determine the position of the epipole, which can be calculated through the fundamental matrix. The system for computing camera positions [31] is used to calculate the fundamental matrix.

As we mentioned above, there are 8 degrees of freedom in the transformation matrix, which requires 4 points to construct. So beside the epipoles, three points providing constraints are selected as follows, to avoid severe projective distortion.

$$(1, 0, f)^T \rightarrow (1, 0, 0)^T \quad (3.16)$$

$$(0, 0, 1)^T \rightarrow (0, 0, 1)^T \quad (3.17)$$

$$(\delta, \delta, 1)^T \rightarrow (\delta, \delta, 1)^T \quad (3.18)$$

$$(\delta, -\delta, 1)^T \rightarrow (\delta, -\delta, 1)^T \quad (3.19)$$

where  $(1, 0, f)$  is the location of the epipole at the  $x$ -axis. The direction of the  $x$ -axis is selected as the row direction and the  $y$ -axis direction is the column direction. The reason for using the  $x$ -axis and  $y$ -axis instead of row and column direction is that the positions in the calculation are no longer integer. The position of  $x$ -axis at the  $y$ -axis can be selected based on the positions of the epipole of each image, with the  $y$ -axis passing through the center of each image.  $\delta$  is an arbitrary number. By setting  $\delta \rightarrow 0$ , the transformation matrix can be obtained as

$$T_r = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -f & 0 & 1 \end{pmatrix} \quad (3.20)$$

After rectification, all epipolar lines are horizontal and parallel with each other, as shown in Fig. 3.9 and 3.10, with Fig. 3.9 the image before rectification and Fig. 3.10 the image after rectification.

### Scaling

The purpose of scaling is to make the correspondences share the same  $y$ -coordinates after rectification, which can also be represented through a transformation matrix acting on the rectified image as

$$T_s = \begin{pmatrix} 1 & 0 \\ 0 & 1/s \end{pmatrix} \quad (3.21)$$

where  $s$  is the scale determined from the correspondences. It is clear that the scaling transformation is just performed on one image and not both.

After rectification and scaling, the correspondences in two different images have almost the same y-coordinates and then the interpolation can be performed in one dimension, or row direction. This scan-line-based approach simplifies the problem from two dimensions to one dimension, so it is easy to handle.

### 3.2.2 Interpolation along the row direction

As Equation (3.13) described, the interpolation is calculating the positions of every pixel in the intermediate image, knowing their corresponding positions in the two original images. Theoretically, these correspondences for interpolation include all pairs of correspondences in the two original images. However, it is impossible to search all of these possible correspondences. An alternative implementation is that the interpolation is performed between the line segments of the two rectified images along the  $x$ -axis direction. Each line segment can be regarded as one group of correspondences.

The interpolation can also be simplified by scaling one of the two rectified images along the row direction. A set of scaling factors along the row direction for the whole image is determined by some set of reliable correspondences between the two images. The public domain software we mentioned above [29] provides the positions of these correspondences. The implementation largely reduces the calculations of the interpolation and better performance can be observed for single objects, such as a building, a statue, a face, etc. than for complex scenes with large depth variations.

### 3.2.3 Post-warping

The intermediate images after interpolation are in the rectified condition, or the specified camera pose. In order to transform the image back to its normal state, post-processing is necessary, which includes post-warping, or the inverse rectification,

after re-scaling by an interpolated scale.

$$s_i = (2 - i)s \tag{3.22}$$

One of the advantages of our method for image rectification is that it makes the post-warping straightforward. Through the epipoles of the original image  $I_1$  and  $I_2$ , assuming they are  $e_1$  and  $e_2$ , the position of the epipole of the intermediate image can be interpolated as

$$e_i = (2 - i)e_1 + (i - 1)e_2 \tag{3.23}$$

which can be regarded as the movement of the epipoles correspondent to the view change.

Then, a transformation matrix like the inverse of (3.20) is employed to warp the interpolated image back to the normal scenario.

### 3.2.4 Simulation results

The procedure of the view synthesis in our experiment is shown below. Fig. 3.9 and Fig. 3.11 are the original pictures of the same scene taken from different viewing directions. Fig. 3.10 and Fig. 3.12 are the images after rectification with Fig. 3.12 scaled in column direction (y-coordinates in this context). After scaling, the same points in the scene have almost the same y-coordinates. Fig. 3.13 is the image after interpolation, and the final result of synthesized intermediate view can be shown in Fig. 3.14, which is post-warped from Fig. 3.13. The interpolation index is  $i = 0.5$  in our experiment, which is corresponding to a viewpoint midway between the viewpoints of the two original images. From the resulting image we can see that the in-between view is properly synthesized.

## 3.3 Conclusions

In this Chapter, we introduce a framework for representing the scene through a set of panoramic images at regularly distributed nodes within the navigation area. The



Figure 3.9: One original image before rectification (the white lines are epipolar lines, which are drawn by a program from Etienne Vincent)

images to construct panoramas can either be captured by a camera or synthesized from the adjacent images, in order to avoid capturing images on a very dense grid, which is usually technically difficult. The advantage of the approach comes from the view interpolation, which means fewer captured images are required compared with other approaches. Moreover, the panorama itself is also a good method to represent the whole scene for some specified applications.

However, there are still some limitations that prevent this approach from wide and practical application, and the limitations are all coming from the image interpolation techniques.

First of all, the techniques of image interpolation are based on two constraints:

- (1) The epipolar constraint: the projection of a scene feature in one image must appear along a particular line in the second image;
- (2) some assumptions on the structure of the scene, such as monotonicity [32], which requires that the relative ordering of points along epipolar lines be preserved.



Figure 3.10: The image in Figure 3.9 after rectification (the white lines are epipolar lines that are nearly horizontal)



Figure 3.11: The other original image before rectification



Figure 3.12: The image in Figure 3.11 after rectification and scaling



Figure 3.13: The intermediate view interpolated from the images in Figure 3.10 and 3.12





Figure 3.14: Intermediate view between Figure 3.9 and 3.11 after post-warping

This condition limits the set of views that can be interpolated, although it is satisfied at least locally for a complex scene.

Secondly, the algorithm of view interpolation relies too much on the establishing of correspondence relationships between points in different images, which is one of the most difficult problems in computer vision. The calculation of the fundamental matrix, the scaling and the position interpolation along the scan line, all rely on the correspondences in the images, which makes the algorithm more complex, and may even require a human interface to implement. From this point of view, the approach is not purely a rendering method.

Thirdly, the intermediate images are only approximately synthesized, which definitely affects the precision of the rendered arbitrary views for the navigation application.

Finally, the view synthesis, or interpolation, is focused on the view from different viewing directions, and is more emphasizing on the texture changing instead of the

light field changing. Thus the purely rendering approach, or light field modelling based approaches, which are efficient in rendering and with more realistic effects, will be discussed in the next chapter.

# Chapter 4

## Light Field Modelling

In the previous chapter, we have discussed one method to represent a scene, in which the techniques of image mosaics and view interpolation are combined. The most important advantage of this approach is that few images need to be captured because the intermediate views can be synthesized through interpolation. However, the view interpolation is performed in an approximate way and it usually makes the rendering algorithm complex for real-time application. Moreover, view interpolation can only reflect the texture changes from different view directions, but not the light intensity changes.

In this chapter, we will discuss techniques that can represent both the texture and the light field changes, which is based on the implementation of the plenoptic function. In the first chapter, we have introduced the concept of the plenoptic modelling, which corresponds to rendering methods without any geometric information.

Plenoptic modelling can be regarded as the holographic representation of an environment or object in digital image format, as it tries to record all light rays from the scene to be represented. Two key issues in the plenoptic modelling are:

- i) Completely recording all light rays for a specified environment or object;
- ii) Efficiently indexing the recorded light rays.

A simple example of plenoptic modelling is the method of panoramas that we have studied in the previous chapter. For a specified position, all light rays toward it can

be recorded and indexed. Thereafter, views in any direction from this position can be rendered when and as required.

In this chapter, more complex techniques to record all light rays of a scene to a specified *area* instead of a position will be studied, which include the Light Field Rendering technique [13] and the Concentric Mosaics technique [14]. We will find that these two methods differ from each other essentially in the technique to index each individual light ray. Comparisons will be made and our conclusion is that the Concentric Mosaics technique is more attractive for practical application. For the Concentric Mosaics technique, we will study the data structure of pre-captured image samples and the rendering algorithm. Some design considerations on capturing Concentric Mosaics data sets [19] will be studied through the plenoptic sampling theory [17].

## 4.1 The Light Field Rendering and the Concentric Mosaics Technique

Levoy and Hanrahan reduced the 5D plenoptic function to a 4D representation of the scene in free space (regions free of occluders), resulting in the Light Field Rendering technique. The Concentric Mosaics technique is another clever method to further reduce the dimension of plenoptic function to 3D by stacking the pixels within one column into one nominal light ray.

### 4.1.1 The Light Field Rendering technique

In the Light Field Rendering technique, the 4D plenoptic function is constructed with the assistance of two parallel planes  $UV$  and  $ST$  as shown in Fig. 4.1. Any arbitrary light ray  $l_k$  can be determined through two points  $M(s_i, t_j)$  in the  $ST$  plane and  $N(u_m, v_n)$  in the  $UV$  plane as,

$$l_k = L(u_m, v_n, s_i, t_j) \quad (4.1)$$

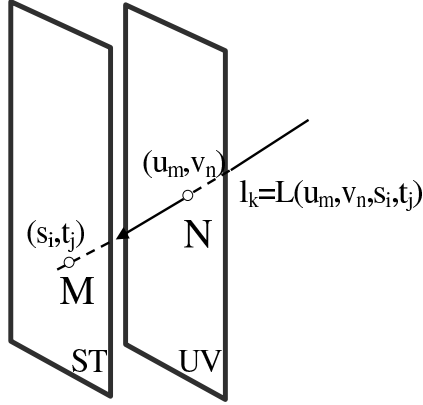


Figure 4.1: 2-plane parameterization of light rays for Light Field Rendering

where  $L(u, v, s, t)$  is the plenoptic function in the Light Field Rendering technique. Thus each light ray can be specified by its intersections with  $UV$  and  $ST$ .

Assume that a camera takes images when moving on one plane such as the  $UV$  plane and the  $ST$  plane is the focal plane (or imaging plane) of the camera. In the image taken at position  $N$  in the figure, every pixel corresponds to one light ray passing through  $N$  and all right rays with same  $(u, v)$  coordinates are recorded in one image, or  $I_{m,n}$  that can be represented as,

$$I_{m,n} = \{l_k = L(u, v, s, t) | u = u_m, v = v_n\}. \quad (4.2)$$

Thus, all light rays are represented by a set of pre-captured image samples captured in the Light Field Rendering technique. The camera can also generally move on an arbitrary trace as long as the camera positions along that trace can be obtained precisely followed by re-combining each light beam into the 2-plane parameterization framework.

However, the above capture procedure can only be applied in a virtual environment. In the practical situation, all right rays with same  $(u, v)$  coordinates cannot be recorded in a single image because the camera's field of view badly limits the light rays that can be recorded in one image. Therefore, more complex motions are required in a practical capture device. One prototype camera gantry to capture image



Figure 4.2: The Light Field Rendering camera gantry (Stanford University) [1]

samples for the Light Field Rendering method which was built at Stanford University is shown in Fig. 4.2.

Fig. 4.3 is an example of the image samples captured in the Light Field Rendering technique [13]. In part (a) of the figure, the light rays are sorted by  $(u, v)$ . Each image in the array of part (a) represents the complete set of light rays passing through one specified point on the  $UV$  plane. More precisely, it includes all possible light rays arriving on the  $ST$  plane passing through one specific position in the  $UV$  plane. The images in this part are actually the images captured by the camera at different positions on the  $UV$  plane in a virtual environment. The positions at which an image is captured are the nodes on a regular grid.

The light rays can also be sorted by  $(s, t)$  as shown in part (b) of Fig. 4.3. Each image in the array of part (b) represents all possible light rays arriving at a specific

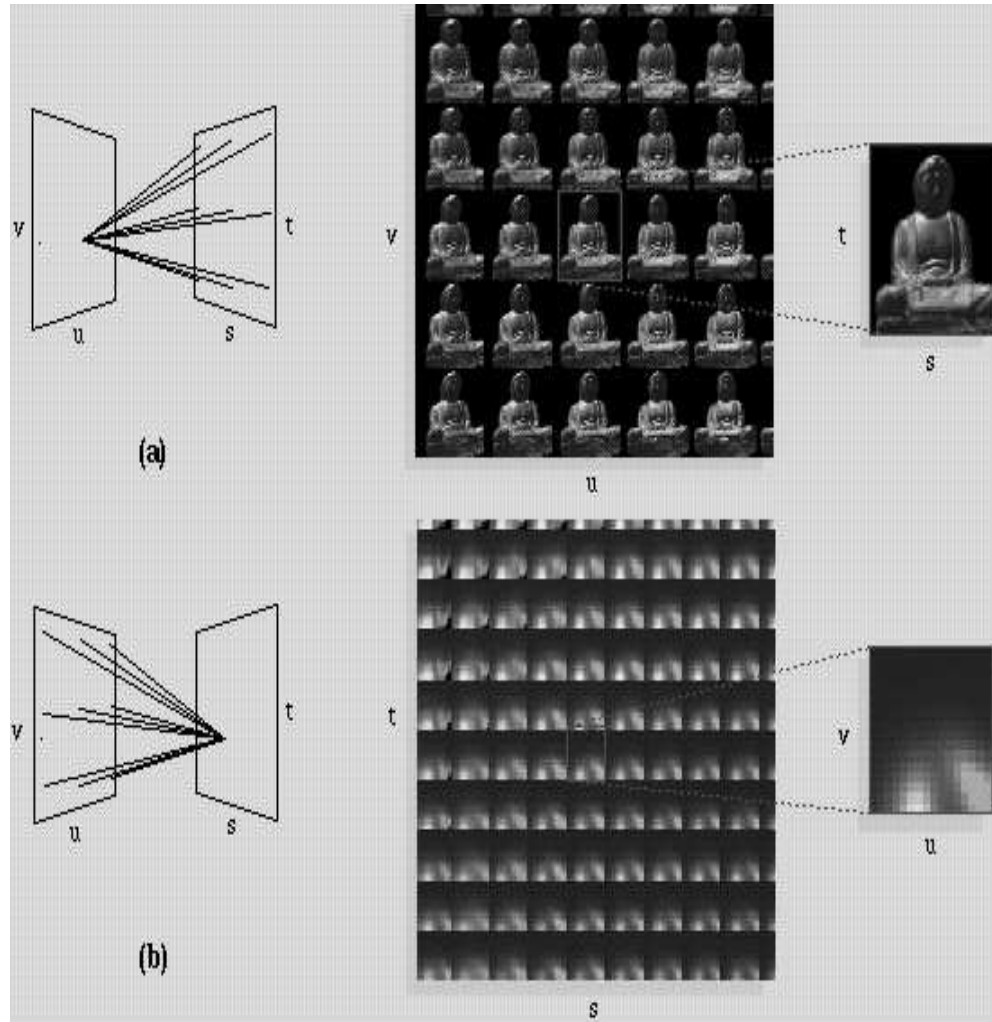


Figure 4.3: An example of the pre-captured image samples for Light Field Rendering [1]

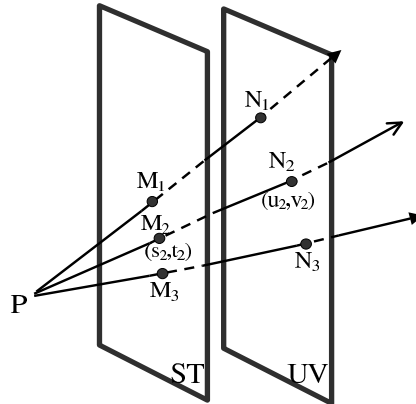


Figure 4.4: Some light rays for an arbitrary position  $P$

position in the  $ST$  plane from any possible position  $(u, v)$  on the  $UV$  plane.

Thus all light rays, which intersect with both the  $UV$  and  $ST$  planes can be recorded through a set of pre-captured images and indexed with the camera's positions on the  $UV$  plane and the pixels' positions on the  $ST$  plane.

The rendering procedure involves the selection of light rays (pixels) from the pre-captured images. A rendering view is an assembly of a set of light rays related to a specified viewing position and viewing direction. Each light ray is obtained through the coordinates of its intersections with the reference  $UV$  and  $ST$  planes, either directly or through interpolation. Three light rays related to position  $P$  are illustrated in Fig. 4.4. Usually, a virtual camera is put at  $P$  to collect all light rays of a novel view.

### 4.1.2 The Concentric Mosaics technique

The Concentric Mosaics technique is another method to generate arbitrary views in a virtual environment through pre-capturing a set of images. The capturing procedure in the Concentric Mosaics technique can be illustrated with Fig. 4.5. The camera is mounted at one end  $E$  of the rotation beam  $CE$ . When  $CE$  rotates around  $C$  at a constant velocity, the video camera takes a sequence of images. A set of pre-captured



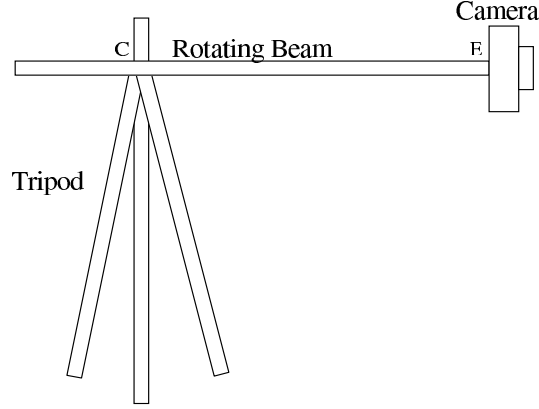


Figure 4.5: The illustration of capture procedure for Concentric Mosaics technique images are obtained after  $CE$  has completed one circle. The navigation area is the area within the inner dashed circle shown in Fig. 4.7, within which any arbitrary views can be synthesized through the Concentric Mosaics rendering algorithm. The radius  $r_{NA}$  of the dashed circle is,

$$r_{NA} = R \cdot \sin\left(\frac{\delta_c}{2}\right) \quad (4.3)$$

where  $R$  is the effective length of the rotation beam, which is the distance from the rotation center to the position of the camera on the beam, or  $CE$  in the figure and the angle  $\delta_c$  is the camera's horizontal field of view.

One practical device to implement the Concentric Mosaics technique built by Microsoft Research is shown in Fig. 4.6.

In the Concentric Mosaics technique, it is not necessary to distinguish the pixels in the same column. Thus, the pixels in one column of each image are grouped into one condensed light ray. Each condensed light ray in the pre-captured image is a sampled ray. The positions at which the images are captured on the camera path are sampled points. In Fig. 4.7,  $SP$  is a sampled point and  $SR$  is a sampled ray.

Let  $P$  be an arbitrary position within the navigation area and  $L_i$  one condensed light ray toward  $P$ . An arbitrary view at position  $P$  is constructed by collecting a set



Figure 4.6: The Concentric Mosaics capture device (Microsoft Research)[2]

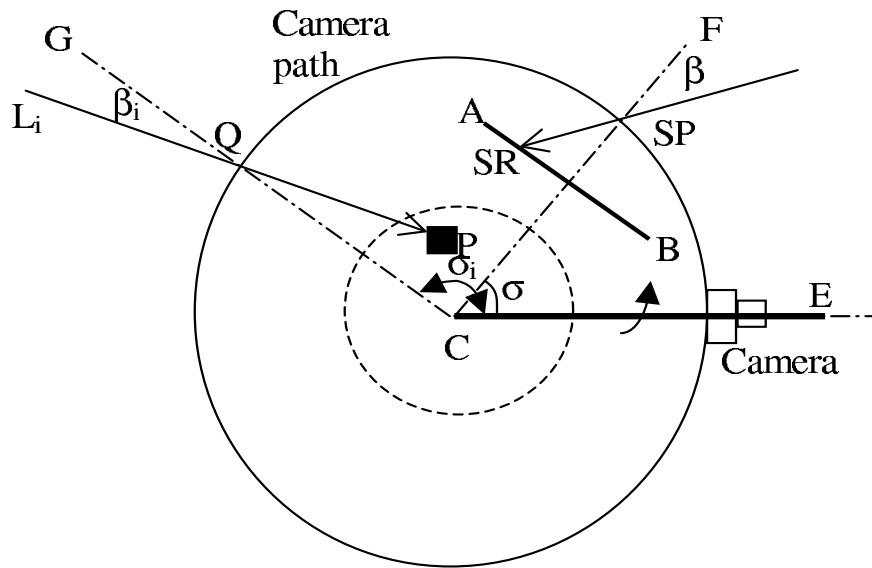


Figure 4.7: The capturing and rendering procedure of the Concentric Mosaiscs technique

of condensed light rays when the viewing direction is given, just like putting a virtual camera at  $P$ .

The condensed light ray  $L_i$  is determined by two angles  $\sigma_i$  and  $\beta_i$  as shown in Fig. 4.7. If the intersection point  $Q$  of  $L_i$  with the camera path happens to be a sampled point and there is a sampled ray corresponding to angle  $\beta_i$ , that sample ray can be directly put into the final image. However, that is not necessarily true. For a general case, the light ray  $L_i$  needs to be interpolated from the nearby light rays that have been captured. The rendering algorithm and the interpolation methods will be studied in the following sections.

There are both advantages and disadvantages of condensing the pixels in one column into one sampled ray. First, it simplifies the device to capture image samples compared with the Light Field Rendering technique. Second, the number of pre-captured images is reduced. The rendering algorithm is column-based instead of pixel-based, which is potentially simple and fast. The view changes along the column

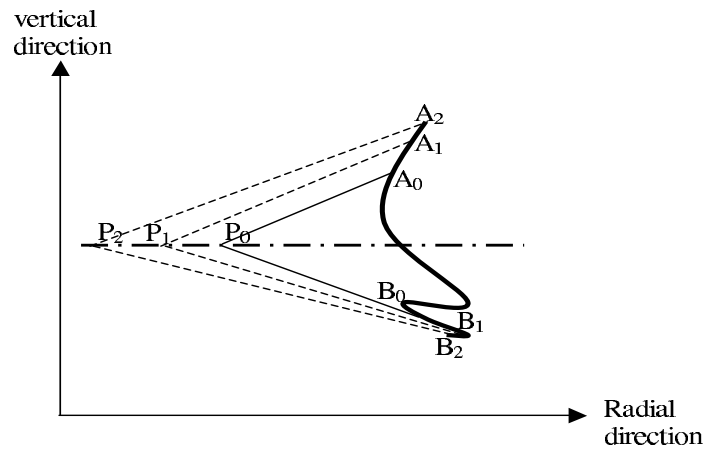


Figure 4.8: The illustration of depth distortion

direction are only scaled according to the distances from the viewing position to the environment.

However, depth distortion is introduced in the rendered views which can be explained with reference to Fig. 4.8. The vertical slit-views  $A_1B_1$  at position  $P_1$ ,  $A_2B_2$  at position  $P_2$ , and  $A_0B_0$  at position  $P_0$  correspond to one column in the image taken at  $P_1$ ,  $P_2$ , and  $P_0$  respectively. In the concentric mosaic technique, they are supposed to be scaled from one sampled ray. This is not the case and the distortion is definitely unavoidable.

### 4.1.3 Comparison of the two techniques

The Light Field Rendering technique provides a method in which the plenoptic function is strictly implemented. All light rays from an environment or an object are recorded and indexable within a spatial area, and the rendering algorithm is pixel-based. From this point of view, the Concentric Mosaics technique is an approximate way to represent the scene and the rendering algorithm is column-based, thus causing depth distortion.

The technical requirements of the camera's motion control to pre-capture image

samples in the Light Field Rendering is more complex in practice, although it is easy to implement in a computer-graphics-based virtual environment. Thus it is more useful as a method to organize the data structure for a virtual environment than to construct a virtual environment with real images due to the technology limitations. Making a beam rotate around a fixed position on the beam at a constant velocity as in the Concentric Mosaics technique is relatively much easier.

In the Light Field Rendering technique, rendering errors reside between all adjacent pixels due to the pixel-based rendering algorithm, whereas they only exist between adjacent columns in the rendered images in the column-based Concentric Mosaics rendering technique. Rendering errors come from control errors in the camera's motion and the interpolation methods. So the quality of the image rendered by the Concentric Mosaics technique is usually better than that of the image rendered by the Light Field Rendering technique. Two images of different environments rendered through the Light Field Rendering technique and Concentric Mosaics technique are provided in Fig. 4.9 (rendered based on the software in [1]) and Fig. 4.10 (rendered based on the pre-captured image data from Microsoft Research), respectively, which verifies our analysis.

The required quantity of pre-captured images will be much larger using the Light Field Rendering technique, compared with that using the Concentric Mosaics technique for the same scene. Thus from the data quantity consideration, the Light Field Rendering technique is more suitable to model an object, instead of an environment.

From the above comparisons, we conclude that the Concentric Mosaics technique is more suitable to construct a real image-based virtual environment. Thus, in the following sections, we will focus on studies of the Concentric Mosaics technique, such as the data structure of the pre-captured images, the rendering algorithm and the design considerations in the capture procedure.



Figure 4.9: A rendered image using the Light Field Rendering technique



Figure 4.10: A rendered image using the Concentric Mosaics technique

## 4.2 The Data Structure in the Concentric Mosaics Technique

The pre-captured image data has a three-dimensional data structure. Each pixel element in the image data set is determined by three indexes: the image number in the sequence, the row number and the column number in a specified image. We know that the image number corresponds to the rotation angle of the rotation beam.

As we discussed in the last section, the pixels in each column are grouped into one condensed light ray in the Concentric Mosaics technique. Thus the data structure in the Concentric Mosaics can be represented in a two-dimensional plane. In Fig. 4.11,  $SP$  is a sampled point and  $SR$  is a sample ray, which corresponds to one column in the image  $AB$  taken by the camera at the rotation angle  $\sigma_{SR}$ . The sampled ray  $SR$  corresponds to a condensed light ray specified through the angle  $\beta_{SR}$  with  $CF$ .  $CF$  is perpendicular to image  $AB$  and passes through its center.

By stacking the pixels of one column into one element, each pre-captured image has a one-dimensional data structure. Thus the data structure of the whole set of pre-captured images can be represented in a  $\sigma$ - $\beta$  plane as shown in the right part of Fig. 4.11, in which  $\beta$  varies from  $-\frac{\delta_c}{2}$  to  $\frac{\delta_c}{2}$  and  $\sigma$  varies from 0 to  $2\pi$  for one circle ( $\delta_c$  is the horizontal field of view of the capture camera). Each dot in Fig. 4.11 denotes a column within the entire set of pre-captured images. All dots in the same horizontal row correspond to one pre-captured image, and all dots in the same vertical line form a panoramic view in a certain viewing direction.

Panoramic images are constructed by re-combining the same column of every image in the sequence together in the same order as the images in the sequence. Fig. 4.12 (a) and (b) are two examples of these panoramic views, with  $\beta = 0$  and  $\beta = \frac{\delta_c}{2}$ , respectively. We can see the displacements and parallaxes between panoramic views from different viewing directions  $\beta$ , for example by observing the tracks in the different images. For the Concentric Mosaics technique, the pre-captured image sequence can be represented both with the original image sequence as taken and



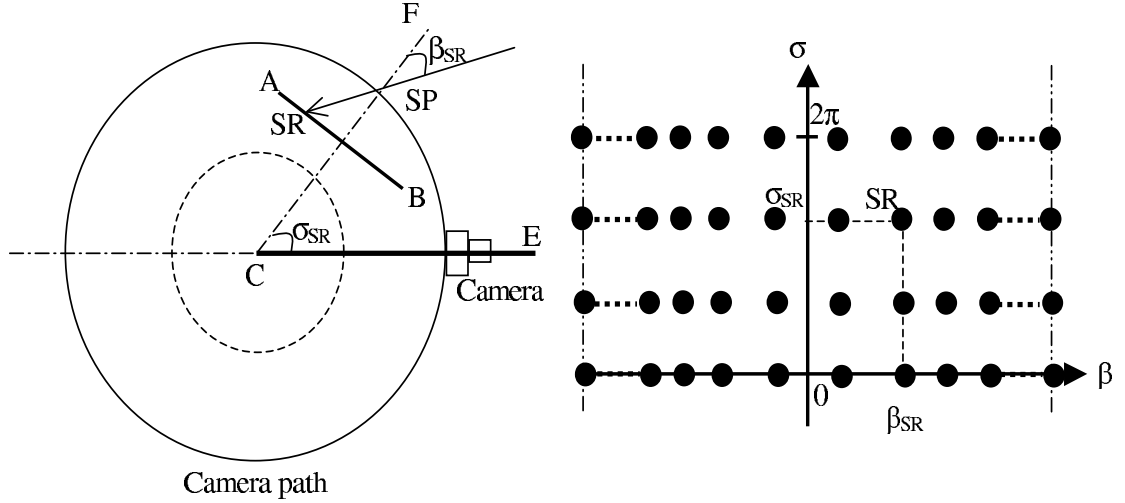


Figure 4.11: A sampled ray (condensed light ray) in two-dimensions and the data structure of Concentric Mosaics pre-captured images

with the set of panoramic views from different viewing directions for the rendering algorithm.

However, the  $\beta$  axis in the above data structure is not homogeneous, since the imaging surface is planar instead of cylindrical. In Fig. 4.13, the angles between the light rays which correspond to the successive columns in the images, or  $\Delta\beta_1, \Delta\beta_2, \dots, \Delta\beta_i, \dots$  in the figure, are not equal due to the flat imaging plane. Assume the coordinate system in the figure is constructed as:  $v$  axis is in the row direction of the image,  $Z$  axis is the normal of the image plane and the origin of  $v$  axis is  $O$ . For an arbitrary condensed light ray  $l_c$ , the angle between  $l_c$  and the  $Z$  axis is  $\beta$  with

$$\tan(\beta) = \frac{v}{f} \quad (4.4)$$

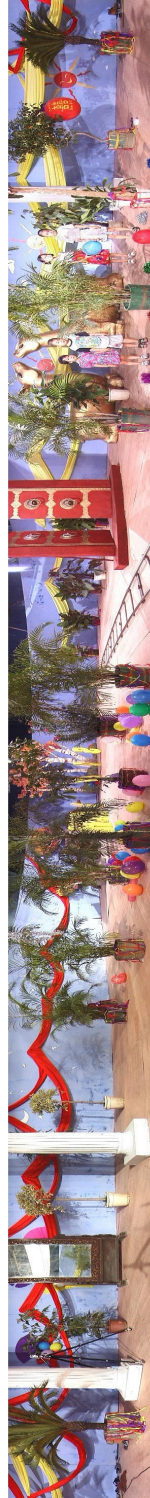
where  $f$  is the focal length of the camera.

Then,

$$\Delta\beta \approx \Delta v \frac{\cos^2(\beta)}{f} \quad (4.5)$$



(a)



(b)

Figure 4.12: The panoramic view with (a)  $\beta \approx 0$  and (b)  $\beta = \frac{\delta_c}{2}$

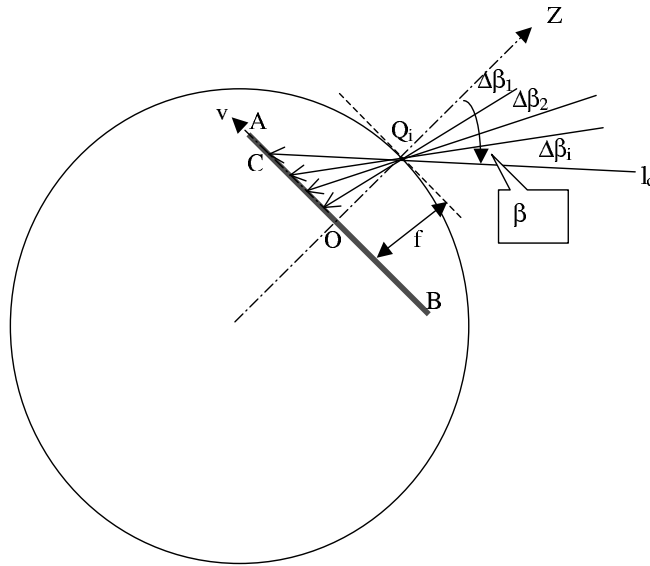


Figure 4.13: Illustration of the non-uniform sampling in the angular  $\beta$  direction

The nonlinear relationship between  $\beta$  and  $\Delta\beta$  is shown in Fig. 4.14. Thus due to the uniform sampling of  $v$  in the image plane,  $\beta$  is not uniformly sampled.

### 4.3 Rendering with the Concentric MosaiCs Technique

In this section, the rendering algorithm for the Concentric MosaiCs technique will be studied, which includes the border conditions for an arbitrary view and the rendering of an arbitrary light ray within the given view. The interpolation methods based on different depth assumptions are the most important element of the rendering algorithm in the Concentric MosaiCs technique.

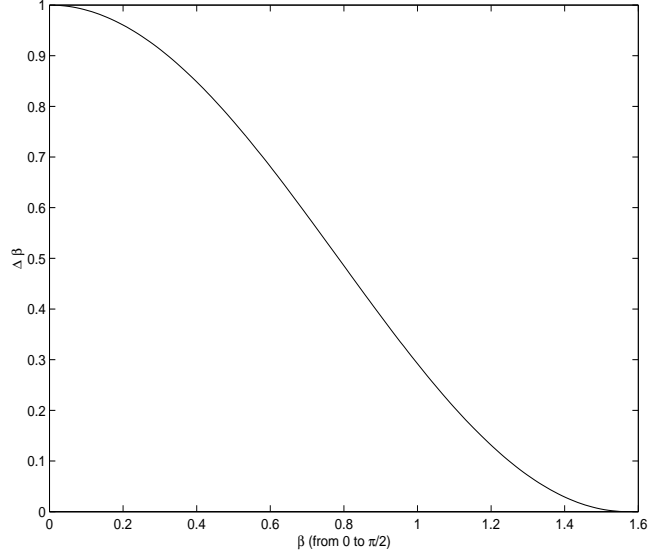


Figure 4.14: The nonlinear relationship between  $\beta$  and  $\Delta\beta$

### 4.3.1 Determining a view: border conditions

In Fig. 4.15, the solid circle represents the camera's moving path in the capture procedure and the area within the dashed circle is the navigation area. Assume that a virtual camera with the horizontal field of view (HFOV)  $\delta$  is located at  $P$ . Let the viewing direction be  $PQ_M$ , which is at angle  $\alpha_M$  between the  $PQ_M$  and the  $x$  axis in the figure (the  $x$  axis is assumed to be the starting position of the rotation beam). The outermost light rays are  $PQ_S$  and  $PQ_E$  as shown and the viewing directions  $\alpha_S$  and  $\alpha_E$  of the outermost light rays  $PQ_S$  and  $PQ_E$  can be represented by,

$$\alpha_S = \alpha_M + \frac{\delta}{2} \quad (4.6)$$

$$\alpha_E = \alpha_M - \frac{\delta}{2} \quad (4.7)$$

The angles  $\beta_S$  and  $\beta_E$  in the figure are,

$$\beta_S = \sin^{-1}\left(\frac{\sin(\alpha_S - \theta)}{R} \cdot \rho\right) \quad (4.8)$$

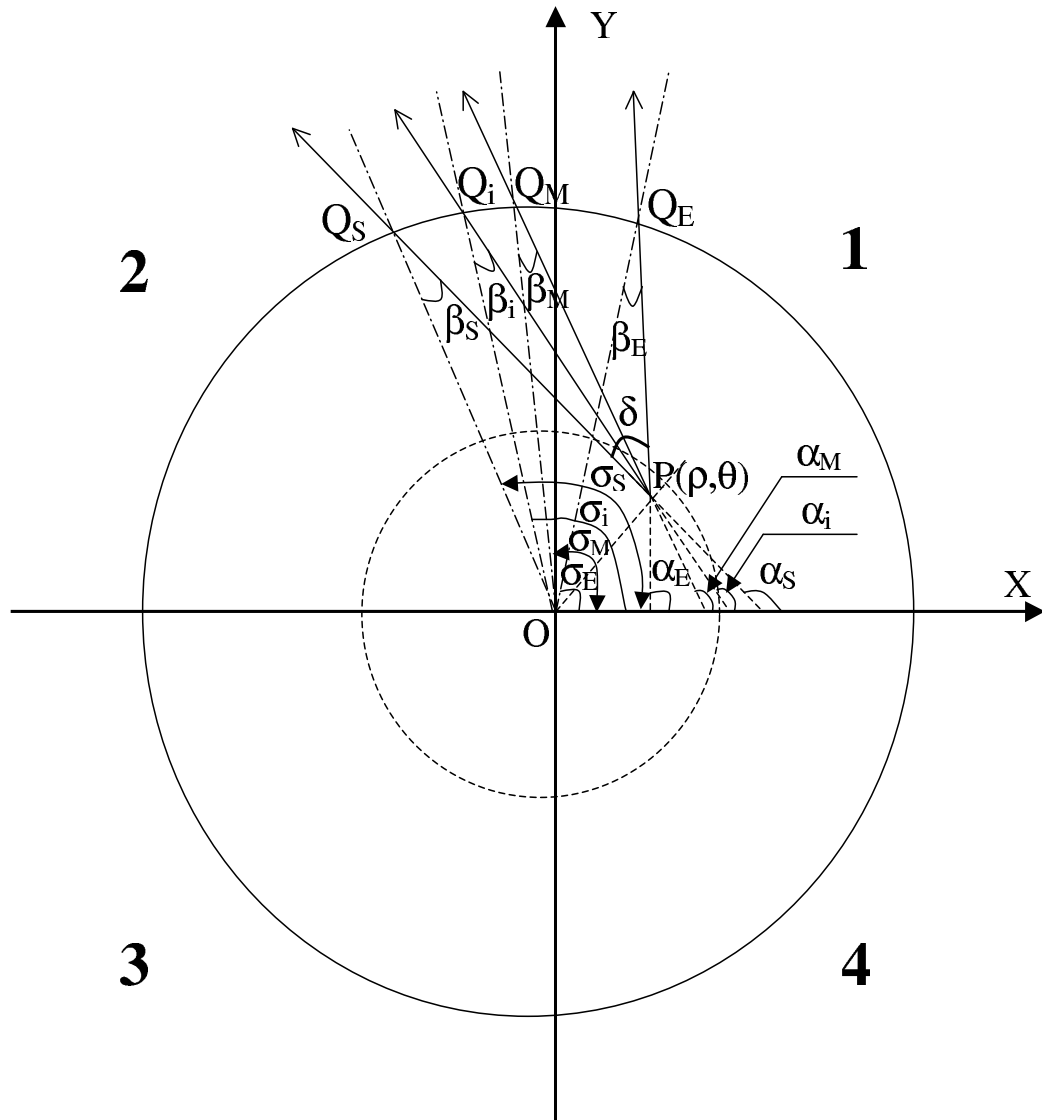


Figure 4.15: Geometric considerations for rendering with Concentric Mosaics (Note that the angles are exaggerated for the purpose of the illustration.  $\rho$  is the distance from  $O$  to  $P$ ,  $\theta$  is the angle between  $OP$  and  $X$  axis)

$$\beta_E = \sin^{-1}\left(\frac{\sin(\alpha_E - \theta)}{R} \cdot \rho\right) \quad (4.9)$$

where  $R$  is the length of the rotation beam.

The angles  $\sigma_S$  and  $\sigma_E$ , which correspond to the positions of the capture camera at  $Q_S$  and  $Q_E$  are

$$\sigma_S = \alpha_S - \beta_S \quad (4.10)$$

$$\sigma_E = \alpha_E - \beta_E \quad (4.11)$$

Assuming that there are  $N_{\text{total}}$  images taken during one complete rotation circle, which corresponds to  $2\pi$  in angle, the indices of first image  $N_S$  and last image  $N_E$  of the above view that will be used in the rendering procedure are

$$N_S = \frac{\sigma_S}{2\pi} \cdot N_{\text{total}} \quad (4.12)$$

$$N_E = \frac{\sigma_E}{2\pi} \cdot N_{\text{total}} \quad (4.13)$$

Thus the total number  $N$  of images that will be use to render this view is

$$N = |N_E - N_S| + 1. \quad (4.14)$$

### 4.3.2 Determining an arbitrary light ray

Assuming the size of images taken by the virtual camera is  $N_{\text{row}}$  by  $N_{\text{col}}$ , there will be  $N_{\text{col}}$  columns in the rendered image for the position  $P$ . These  $N_{\text{col}}$  columns correspond to  $N_{\text{col}}$  light rays, where each of them can be represented by a viewing direction  $\alpha_i$ ,

$$\alpha_i = \alpha_E + \frac{\delta}{N_{\text{col}}} \cdot (i - 1), i = 1, 2, \dots, N, \quad (4.15)$$

and the corresponding value of  $\beta_i$  is,

$$|\beta_i| = \left| \sin^{-1}\left(\frac{\sin(\alpha_i - \theta)}{R} \cdot \rho\right) \right|. \quad (4.16)$$

The sign of  $\beta_i$  depends on different quadrants (the dark font numbers 1, 2, 3, 4 in the figure indicate different quadrants) at which  $P$  is located and the angle between  $PQ_i$

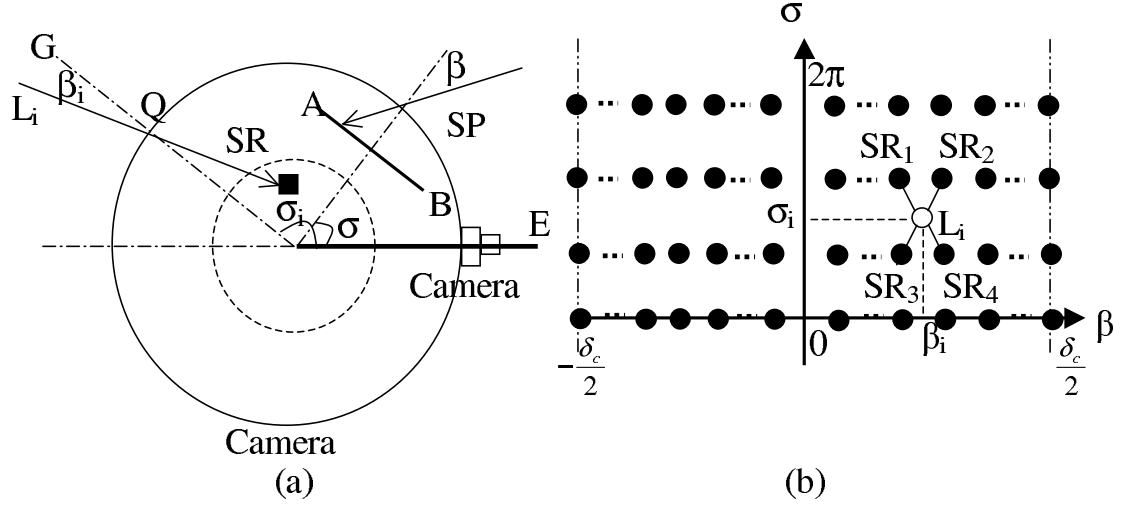


Figure 4.16: The interpolation in the rendering algorithm

and  $OP$  in each quadrant. Different cases have to be considered in the implementation. The angle  $\sigma_i$  which determines the image number in the captured data set is given by

$$\sigma_i = \alpha_i - \beta_i. \quad (4.17)$$

### 4.3.3 Interpolation methods

Based on the data structure of the pre-captured images in the Concentric Mosaics technique, the interpolation problem in the rendering algorithm can be clearly illustrated in Fig. 4.16. Any arbitrary condensed light ray, such as  $L_i$  in Fig. 4.16(a), is determined by two angles  $\sigma_i$  and  $\beta_i$  as shown in the figure. If the intersection point  $Q$  of  $L_i$  with the camera path happens to be a sampled point and there is a sampled ray corresponding to  $\beta_i$ , that sample ray can be directly put into the final image. However, that is not necessarily true. For a general case,  $L_i$  is one point illustrated in Fig. 4.16(b), which will be interpolated from its nearby sampled rays  $SR_1$ ,  $SR_2$ ,  $SR_3$  and  $SR_4$  in the figure, as

$$L_i = \omega_1 SR_1 + \omega_2 SR_2 + \omega_3 SR_3 + \omega_4 SR_4 \quad (4.18)$$

where  $\omega_1$ ,  $\omega_2, \omega_3$ , and  $\omega_4$  are the weights for interpolation.

Depth information of the environment is required to calculate the best interpolation weights, but this is difficult to obtain. Thus the infinite depth assumption and the constant depth assumption are used in practice. The nearest sampled ray approximation can also be considered as a special case of the above interpolation formula, with only one weight equal to one while the others are zero. The following three interpolation methods, which are Nearest Sampled Rays, Linear Interpolation with Infinite Depth Assumption and Linear Interpolation with Constant Depth Assumption, are proposed in [18]. The rendering procedure will be related in detail combined with interpolation methods for implementation, which are usually omitted in the papers.

### Nearest sampled rays (NSR)

One efficient and fast method to render a light ray is to find the nearest sampled ray from the nearest sampled point. The geometric relationship of the light ray  $L_i$  with its nearby sampled rays  $SR_1$ ,  $SR_2$ ,  $SR_3$  and  $SR_4$  is shown in Fig. 4.17, with the  $\sigma$ - $\beta$  data structure in part (b). Although the sampling structure in the  $\sigma$ - $\beta$  plane is not uniform as we have illustrated in the previous section with Fig. 4.13, it will not cause significant difference when we apply a local uniformity approximation. Thus, the nearest sampled ray approximation is to select the nearest one from  $SR_1$ ,  $SR_2$ ,  $SR_3$  and  $SR_4$  to represent  $L_i$ .

However, studies [18] have shown that it will cause significant aliasing if we just calculate the angle  $\beta_i$  (the angle between the view direction  $PQ$  and the radial direction  $OQ$  at the point of intersection  $Q$ ) and find the nearest sampled ray in sampled point  $Q_2$  based on  $\beta_i$ . One method to find the nearest sampled ray is proposed in [18]. In Fig. 4.17, the light rays  $P_1Q_1$  and  $P_2Q_2$  are parallel to the light ray  $L_i$ , or  $PQ$ , and passing through the nearest sampled points to  $P$ , namely  $Q_1$  and  $Q_2$ . The light ray  $L_i$  will be represented by one of four sampled rays  $SR_1$ ,  $SR_2$ ,  $SR_3$  and  $SR_4$ .

$$SR_1 = [N_1] \leftrightarrow [M_{i,1}] \quad (4.19)$$



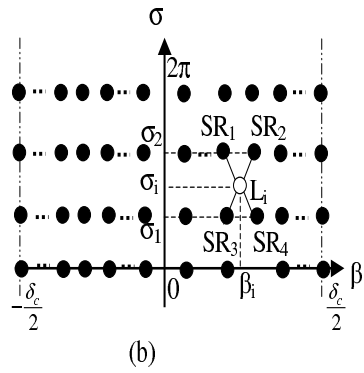
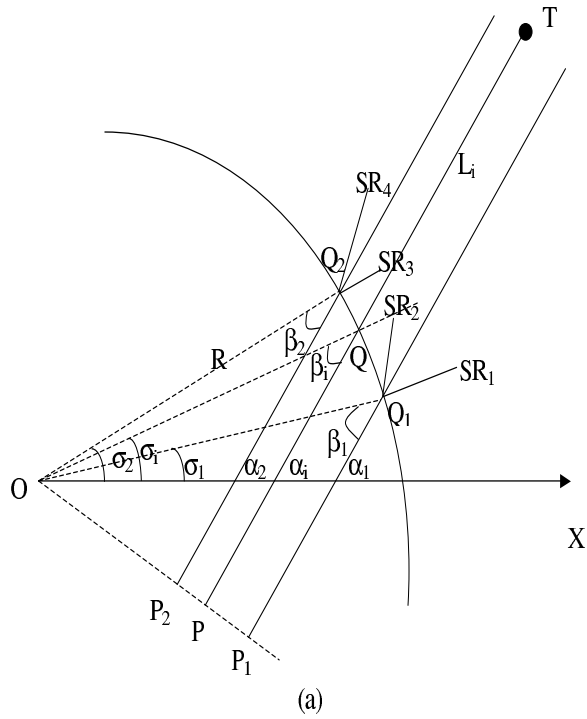


Figure 4.17: Nearest Point Approximation and Infinite Depth Assumption Interpolation (Note that the angles are exaggerated for the purpose of the illustration)

$$SR_2 = [N_1] \leftrightarrow [M_{i,2}] \quad (4.20)$$

$$SR_3 = [N_2] \leftrightarrow [M_{i,3}] \quad (4.21)$$

$$SR_4 = [N_2] \leftrightarrow [M_{i,4}] \quad (4.22)$$

where the function  $[N_{\text{image}}] \leftrightarrow [M_{\text{column}}]$  represents taking column  $M_{\text{column}}$  from the image  $N_{\text{image}}$  with  $1 \leq N_{\text{image}} \leq N_{\text{total}}$ . Here,

$$N_i = \frac{\sigma_i}{2\pi} \cdot N_{\text{total}} \quad (4.23)$$

$$N_2 = \lceil N_i \rceil \quad (4.24)$$

$$N_1 = \lfloor N_i \rfloor \quad (4.25)$$

where  $\lceil x \rceil$  and  $\lfloor x \rfloor$  are the operations that round the element  $x$  to the nearest integer toward infinity and minus infinity, respectively.

The column numbers of the corresponding pre-captured images are

$$M_{i,1} = \lfloor M_i^- \rfloor \quad (4.26)$$

$$M_{i,2} = \lceil M_i^- \rceil \quad (4.27)$$

$$M_{i,3} = \lfloor M_i^+ \rfloor \quad (4.28)$$

$$M_{i,4} = \lceil M_i^+ \rceil \quad (4.29)$$

where

$$M_i^+ = \frac{N_{\text{row}}}{2} \pm \frac{\tan(\beta_2)}{\tan(\frac{\delta}{2})} \cdot \frac{N_{\text{row}}}{2} \quad (4.30)$$

$$M_i^- = \frac{N_{\text{row}}}{2} \pm \frac{\tan(\beta_1)}{\tan(\frac{\delta}{2})} \cdot \frac{N_{\text{row}}}{2} \quad (4.31)$$

The choice of ‘plus’ or ‘minus’ depends on the relationship between view direction  $P_2Q_2$  and the radial direction  $OQ_2$ , and the relationship between view direction  $P_1Q_1$  and the radial direction  $OQ_1$ . The parameters  $\beta_1$  and  $\beta_2$  are given by

$$\beta_2 = \alpha_2 - \sigma_2 \quad (4.32)$$

$$\beta_1 = \alpha_1 - \sigma_1 \quad (4.33)$$

where

$$\alpha_2 = \alpha_1 = \alpha_i \quad (4.34)$$

$$\sigma_2 = \frac{N_2}{N_t} \cdot 2\pi \quad (4.35)$$

$$\sigma_1 = \frac{N_1}{N_t} \cdot 2\pi \quad (4.36)$$

In the method of nearest sampled rays, the nearest sampled point,  $Q_1$  or  $Q_2$  in the figure is determined by the distance  $PP_1$  and  $PP_2$ . Once the nearest sampled point is determined, the nearest sampled rays at that sampled point is determined through the parameters  $\phi_1$  and  $\phi_2$  at  $Q_1$ , or parameters  $\phi_3$  and  $\phi_4$  at  $Q_2$ , where

$$\phi_1 = |M_{i,1} - M_i^-| \quad (4.37)$$

$$\phi_2 = |M_{i,2} - M_i^-| \quad (4.38)$$

$$\phi_3 = |M_{i,3} - M_i^+| \quad (4.39)$$

$$\phi_4 = |M_{i,4} - M_i^+| \quad (4.40)$$

Thus for the NSR interpolation method,

$$L_i = \begin{cases} SR_1 & \text{if } P_1P < PP_2 \ \phi_1 < \phi_2 \\ SR_2 & \text{if } P_1P < PP_2 \ \phi_1 > \phi_2 \\ SR_3 & \text{if } P_1P > PP_2 \ \phi_3 < \phi_4 \\ SR_4 & \text{if } P_1P > PP_2 \ \phi_3 > \phi_4 \end{cases} \quad (4.41)$$

It is obvious that aliasing still remains in the above method because each ray is in fact approximated with a sampled ray that is not exactly parallel to the view direction and has a distance offset to the view direction. Thus more precise rendering algorithms require some kind of interpolation, which corresponds to a low pass filter to reduce aliasing as much as possible. As we mentioned before, the precise rendering requires the depth information about the environment, which is difficult or even impossible to obtain. In the following, we will investigate the methods of linear interpolation based on two simple assumptions on depth: (1) infinite depth and (2) constant depth.

### Linear interpolation with infinite depth assumption (LIIDA)

Instead of selecting one sampled ray from four nearby candidates, all four sampled rays are used to calculate the condensed light ray  $L_i$  in the linear interpolation with infinite depth assumption,

$$L_i = \omega_1 SR_1 + \omega_2 SR_2 + \omega_3 SR_3 + \omega_4 SR_4 \quad (4.42)$$

with the weights  $\omega_1$ ,  $\omega_2$ ,  $\omega_3$  and  $\omega_4$  calculated by [18],

$$\omega_1 = \frac{|PP_2| \phi_2}{|P_1P_2| (\phi_1 + \phi_2)} \quad (4.43)$$

$$\omega_2 = \frac{|PP_2| \phi_1}{|P_1P_2| (\phi_1 + \phi_2)} \quad (4.44)$$

$$\omega_3 = \frac{|PP_1| \phi_4}{|P_1P_2| (\phi_3 + \phi_4)} \quad (4.45)$$

$$\omega_4 = \frac{|PP_1| \phi_3}{|P_1P_2| (\phi_3 + \phi_4)} \quad (4.46)$$

### Linear interpolation with constant depth assumption (LICDA)

The difference between the linear interpolation with constant depth assumption and with infinite depth assumption is the different methods to calculate the weights for the interpolation.

The geometric relationship for the constant depth assumption can be shown in Fig. 4.18. The weights are [18],

$$\omega_1 = \frac{\gamma_1 \phi_2}{(\gamma_1 + \gamma_2)(\phi_1 + \phi_2)} \quad (4.47)$$

$$\omega_2 = \frac{\gamma_1 \phi_1}{(\gamma_1 + \gamma_2)(\phi_1 + \phi_2)} \quad (4.48)$$

$$\omega_3 = \frac{\gamma_2 \phi_4}{(\gamma_3 + \gamma_4)(\phi_3 + \phi_4)} \quad (4.49)$$

$$\omega_4 = \frac{\gamma_2 \phi_3}{(\gamma_3 + \gamma_4)(\phi_3 + \phi_4)} \quad (4.50)$$

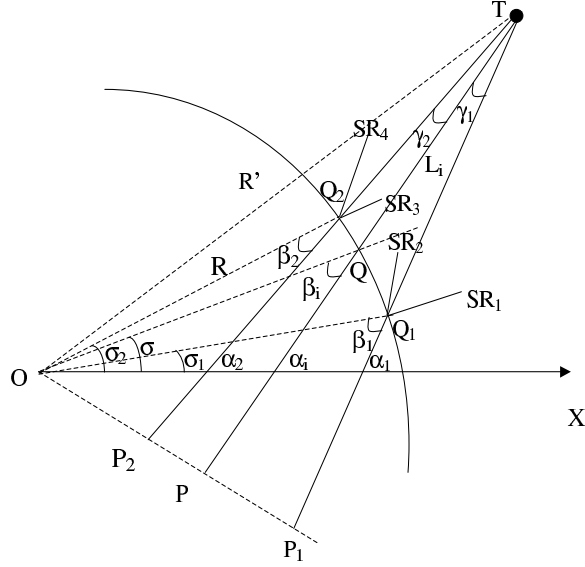


Figure 4.18: Linear interpolation with constant depth assumption (Note that the angles are exaggerated for the purpose of the illustration)

where

$$\gamma_1 = \alpha_1 - \alpha_i \quad (4.51)$$

$$\gamma_2 = \alpha_i - \alpha_2. \quad (4.52)$$

The parameters  $\alpha_1$  and  $\alpha_2$  are given by [18],

$$\alpha_2 = \tan^{-1}\left(\frac{R' \sin(\Theta) - R \sin(\sigma_2)}{R' \cdot \cos(\Theta) - R \cdot \cos(\sigma_2)}\right) \quad (4.53)$$

$$\alpha_1 = \tan^{-1}\left(\frac{R' \sin(\Theta) - R \sin(\sigma_1)}{R' \cdot \cos(\Theta) - R \cdot \cos(\sigma_1)}\right) \quad (4.54)$$

where  $R'$  is the distance from the rotation center to any point of the scene which is a constant based on the constant depth assumption.  $\Theta$  is the angle between  $OT$  and  $x$  axis in the figure,

$$\Theta = \alpha_i - \sin^{-1}\left(\frac{R \sin(\beta_i)}{R'}\right). \quad (4.55)$$

Then with,

$$\beta_2 = \beta_i + (\alpha_2 - \alpha_i) - (\sigma_2 - \sigma_i) \quad (4.56)$$

$$\beta_1 = \beta_i + (\alpha_1 - \alpha_i) - (\sigma_1 - \sigma_i), \quad (4.57)$$

similar to the interpolation under infinite depth assumption, we have

$$M_{i,1} = \lfloor M_i^- \rfloor \quad (4.58)$$

$$M_{i,2} = \lceil M_i^- \rceil \quad (4.59)$$

$$M_{i,3} = \lfloor M_i^+ \rfloor \quad (4.60)$$

$$M_{i,4} = \lceil M_i^+ \rceil \quad (4.61)$$

and

$$M_i^+ = \frac{N_{\text{row}}}{2} \pm \frac{\tan(\beta_2)}{\tan(\frac{\delta}{2})} \cdot \frac{N_{\text{row}}}{2} \quad (4.62)$$

$$M_i^- = \frac{N_{\text{row}}}{2} \pm \frac{\tan(\beta_1)}{\tan(\frac{\delta}{2})} \cdot \frac{N_{\text{row}}}{2} \quad (4.63)$$

and

$$\phi_1 = |M_{i,1} - M_i^-| \quad (4.64)$$

$$\phi_2 = |M_{i,2} - M_i^-| \quad (4.65)$$

$$\phi_3 = |M_{i,3} - M_i^+| \quad (4.66)$$

$$\phi_4 = |M_{i,4} - M_i^+| \quad (4.67)$$

#### 4.3.4 Simulation results and observations

Fig. 4.19, 4.20 and 4.21 show images rendered using the methods of NSR, LIIDA and LICDA, respectively. We can see that the quality of the image rendered through LIIDA is very close to the quality of the image rendered through LICDA, and that both are much better than the quality of the image rendered with NSR.

The interpolation in the spatial domain corresponds to a low pass filter in the frequency domain and an optimal filter must exist for proper rendering. However, the design of the optimal filter requires depth information, which is difficult to obtain in practice.



Figure 4.19: Rendering with nearest sampled rays method



Figure 4.20: Rendering through linear interpolation with infinite depth assumption



Figure 4.21: Rendering through linear interpolation with constant depth assumption

The constant depth assumption uses the average depth of the environment, which is a compromise between no depth information and accurate depth information. The average depth of the environment is usually large, which makes the infinite depth assumption a good approximation. Further comparisons between LIIDA and LICDA can be found in [18]. The nearest sample rays method introduces significant aliasing without any filtering consideration.

In the following, we study the sampling problem of the Concentric Mosaics technique to get some considerations on the designing of the capture devices for practical application.



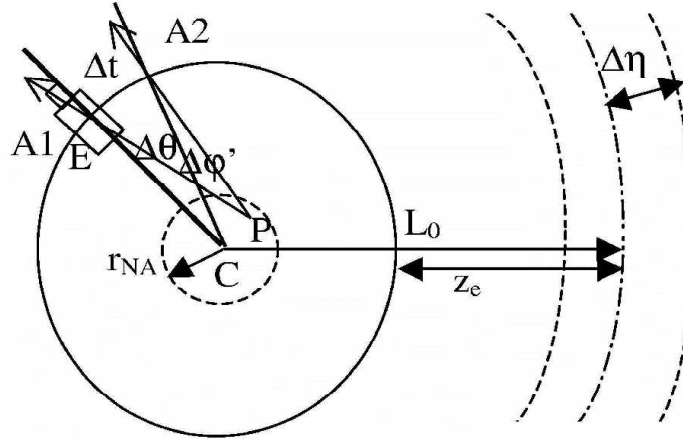


Figure 4.22: The sampling model of Concentric Mosaics

## 4.4 Design Considerations in the Capture of Concentric Mosaics Data

The capture procedure in the Concentric Mosaics technique is shown in Fig. 4.22, with CE the rotation beam. Design parameters to choose in the Concentric Mosaics technique are: (i) the length of the rotation beam  $R$ , or the distance from  $C$  to  $E$  in Fig. 4.22; (ii) the rotation angle  $\Delta\theta$  between the adjacent camera positions in Fig. 4.22.

The following quantities are assumed known:

(i) the average depth  $L_0$  and the depth variation  $\Delta\eta$  of the scene (the depth value should be located within the range  $(L_0 - \Delta\eta, L_0 + \Delta\eta)$ );

(ii) some parameters of the camera: horizontal field of view, HFOV, and image size,  $N_{\text{row}}$  by  $N_{\text{col}}$ .

As in Fig. 4.22,  $A1$  and  $A2$  are adjacent positions of the camera at which the images are taken, which correspond to the rotation angle between the adjacent camera positions, or  $\Delta\theta$  in angle and  $\Delta t = R\Delta\theta$  in arc distance.

In practical applications, it is inconvenient to obtain depth information about the real scene, so we will use the minimum sampling with depth uncertainty to discuss the design considerations of the Concentric Mosaiscs technique. By the formula for the minimum sampling with depth uncertainty [17],

$$\Delta t_{\max} = \frac{\min_{z_e} z_e^2 - \Delta \eta^2}{2f_c K_{f_v} \Delta \eta} \quad (4.68)$$

where  $z_e$  is the estimated distance from the scene to the camera. More than one estimated distance at different scene positions may be used in the light field rendering technique to improve the quality of the rendered image. In the Concentric Mosaiscs technique, the average distance of the scene to the camera is used as the only estimated distance.  $K_{f_v}$  is the highest frequency of the captured data, limited by the resolution limitation of camera, or

$$K_{f_v} = \frac{1}{2\Delta \nu_c} \quad (4.69)$$

where  $\Delta \nu_c$  is the pixel size of the capturing camera, and  $f_c$  is the focal length of the capturing camera.

Using the approximation,

$$\frac{\Delta \nu_c}{f_c} = \Delta \phi = \frac{\text{HFOV}}{N_{\text{col}}} \quad (4.70)$$

where  $\Delta \phi$  is the angle resolution for one pixel,

$$\Delta t_{\max} = \frac{z_e^2 - \Delta \eta^2}{\Delta \eta} \left( \frac{\text{HFOV}}{N_{\text{col}}} \right) \quad (4.71)$$

which is convenient for practical calculation.

Thus

$$\Delta \theta = \frac{\Delta t_{\max}}{R} = \frac{z_e^2 - \Delta \eta^2}{\Delta \eta R} \left( \frac{\text{HFOV}}{N_{\text{col}}} \right) \quad (4.72)$$

From equation (4.72) we can determine the relationship between the rotation angle across the adjacent positions at which the images are taken and the length of the rotation beam for the given scene and capturing camera. Is there any principle to follow for choosing  $R$  and thus  $\Delta \theta$  for a specified scene with a given camera?

#### 4.4.1 Considerations from sampling rate

Assume that a virtual camera is located at  $P$ , with the same camera parameters as the capturing camera. Then the number of columns  $N'_{\text{col}}$  of one rendered frame, which equals the number of image samples within the HFOV of the virtual camera located at  $P$ , can be calculated by,

$$N'_{\text{col}} = \frac{\text{HFOV}}{\Delta\phi'} = \frac{\text{HFOV} \cdot |PA_{1,2}|}{\Delta t} \quad (4.73)$$

where

$$|PA_{1,2}| \approx |PA_1| \approx |PA_2| \quad (4.74)$$

is the distance from  $P$  to  $A_1$  and  $A_2$ ,  $\Delta\phi'$  is the angle between  $PA_1$  and  $PA_2$ . Then

$$(N'_{\text{col}})_{\min} = \frac{\text{HFOV}}{(\Delta\phi')_{\max}} = \frac{\text{HFOV} \cdot |PA_{1,2}|_{\min}}{\Delta t_{\max}} \quad (4.75)$$

Considering the navigation area provided by the Concentric Mosaics technique, which is the dashed circle in Fig. 4.22 with radius  $r_{\text{NA}}$ , we can get

$$|PA_{1,2}|_{\min} \approx R - r_{\text{NA}} \quad (4.76)$$

where

$$r_{\text{NA}} = R \cdot \sin\left(\frac{\text{HFOV}}{2}\right). \quad (4.77)$$

Assume that the rendered images reach the same resolution as the captured images, or in the form of the number of columns per frame,

$$(N'_{\text{col}})_{\min} \geq N_{\text{col}}. \quad (4.78)$$

Using the geometric relationship

$$L_0 = R + z_e \quad (4.79)$$

and chaining equations (4.71), (4.75), (4.76), (4.77), (4.79) with (4.78),

$$R_{\text{MIN}} = L_0 + \left(\frac{\Delta\eta}{2}\right)(1 - \sin\omega) - \sqrt{\Delta\eta^2 + \Delta\eta^2 \frac{(1 - \sin\omega)^2}{4} + L_0 \cdot \Delta\eta \cdot (1 - \sin\omega)} \quad (4.80)$$

where  $w = \text{HFOV}/2$ . Thus equation (4.80) gives us the optimal length of the rotation beam for all rendered images to achieve at least the same resolution as the captured images. However, the optimal length here is just based on the sampling consideration, so we use  $R_{\text{MIN}}$  (Minimum) instead of  $R_{\text{opt}}$  (Optimal) to indicate that it is the minimum length of the rotation beam for the rendered images to achieve the camera resolution. When the length of the rotation beam is shorter than  $R_{\text{MIN}}$ , fewer samples are required, which means that the highest frequency of the rendered image will be less than that of the camera. In the limit, as  $R$  tends to zero, the Concentric Mosaics technique will create panoramas. Thus, even if more samples are intentionally added by taking more images along the rotation direction when  $R$  is less than  $R_{\text{MIN}}$ , the rendered images can never benefit from these extra samples. In the opposite case, when the length of the rotation beam is longer than  $R_{\text{MIN}}$ , more samples along the rotation direction are required for sampling to avoid aliasing. More samples than required are provided for rendering, with down-sampling applied to reach the proper image aspect ratio.

#### 4.4.2 Considerations from the number of samples

The navigation area will increase as the length of the rotation beam increases, but so will the number of image samples. The increase of the radius of the navigation area  $\Delta r_{\text{NA}}$  can be represented from equation (4.77) by

$$\Delta r_{\text{NA}} = \Delta R \cdot \sin\left(\frac{\text{HFOV}}{2}\right). \quad (4.81)$$

Based on the angle between two positions at which the images are taken, the number of image samples for one rotation can be calculated by

$$N_R = \frac{2\pi}{\Delta\phi} \quad (4.82)$$

so,

$$N_R = \frac{2\pi R \Delta\eta}{L_0^2 - \Delta\eta^2 + R^2 - 2L_0 R} \left(\frac{N_{\text{col}}}{\text{HFOV}}\right) \quad (4.83)$$

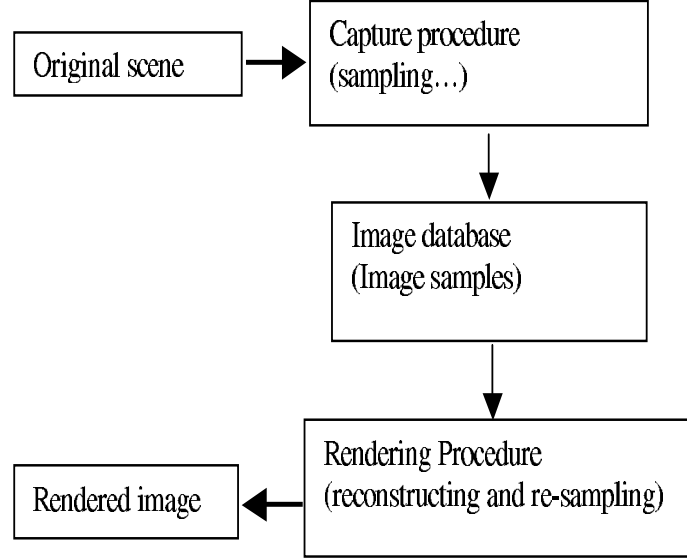


Figure 4.23: The frequency domain interpolation

and its increasing tendency with the increase of the length of the rotation beam,

$$N_R = \left(1 + \frac{2R(L_0 - R)}{L_0^2 - \Delta\eta^2 + R^2 - 2L_0R}\right) \left(\frac{2\pi\Delta\eta N_{\text{col}}}{\text{HFOV}}\right). \quad (4.84)$$

#### 4.4.3 The frequency domain interpretation for our analysis

The frequency domain model of image based rendering in the Concentric Mosaics technique can be illustrated through Fig. 4.23. The low pass filter of the capturing camera is associated with the capturing procedure and it determines the sampling step in formula (4.68) in order to satisfy the Nyquist criterion. The rendering procedure can be regarded as a reconstruction and resampling process, up-sampling or down-sampling. Thus it is also associated with a low pass filter, and the cut-off frequency is equal to the frequency bound of the image samples at the optimal condition.

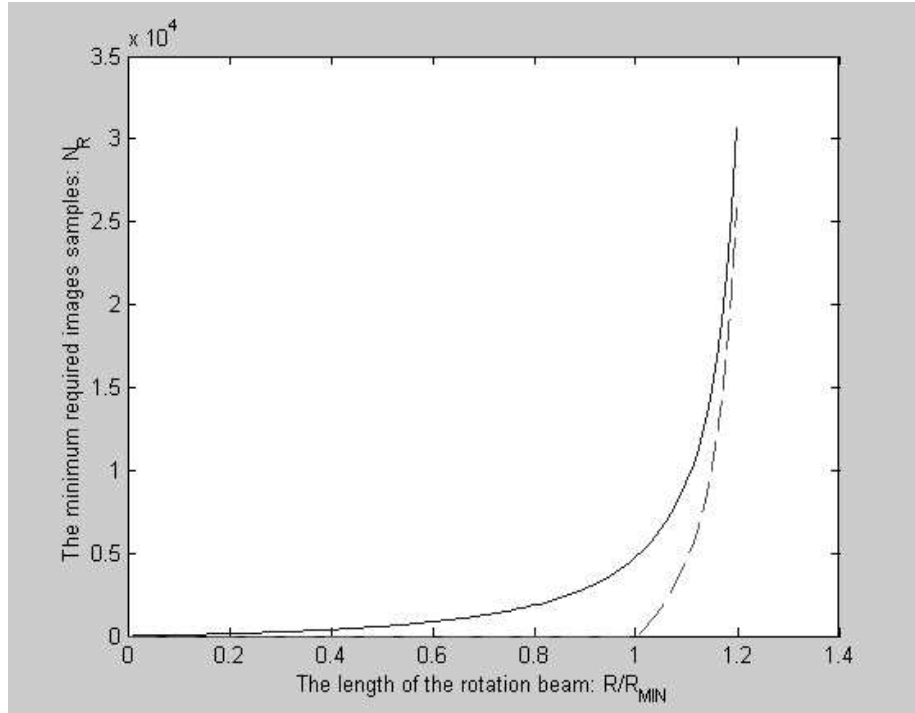


Figure 4.24: The minimum number  $N_R$  of image samples at different relative lengths  $R/R_{\text{MIN}}$  of the rotation beam

#### 4.4.4 Simulations

##### Simulation 1: The relationship between the number of image samples with the different lengths of the rotation beam

We assume an example environment with the following parameters: (i) the average depth  $L_0 = 10m$ ; (ii) depth variation  $\Delta\eta = 3m$ . The imaging parameters are: (i) the captured image size: 360pixels by 288pixels; (ii) the HFOV of the capturing camera is  $43^\circ$ .

The minimum number  $N_R$  of image samples at different lengths  $R$  of the rotation beam for the specified environment is shown in Fig. 4.24 in solid line, with the dashed line indicating the increase of image samples with respect to that at the minimum length  $R_{\text{MIN}}$  (5.57m), when  $R$  is larger than  $R_{\text{MIN}}$ . The lengths of the rotation beam



Figure 4.25: Down-sampling factor  $t=1$  (from original data set)

are relative values, compared with  $R_{\text{MIN}}$ . We can see that when  $R$  is larger than  $R_{\text{MIN}}$ , the minimum number of image samples is increasing very fast, and thus the image data amount, while the increase of the radius of navigation is linear with the increase of the length of the rotation beam, or  $\Delta r_{\text{NA}} = 0.37\Delta R$  in our case here.

### **Simulation 2: The image quality at different sampling rates with fixed length of rotation beam**

The simulation of the relationship between the quality of the rendered images with the number of image samples along the rotation is performed based on a Concentric Mosaics data set provided by the Microsoft Research, Beijing. The original data set has 2967 pictures in one rotation, and the image samples are down-sampled by a factor  $t$  in the rotation direction. The rendered images using linear interpolation with a constant depth assumption are shown in Fig. 4.25, Fig. 4.26 and Fig. 4.27 with different down sampling factors  $t = 1$ , (original);  $t = 2$ ; and  $t = 3$  (i.e. with the number of image samples  $N_{R1} = 2967$ ;  $N_{R2} = 1483$ ;  $N_{R3} = 989$ ). We find that the image quality is lowered significantly as the number of image samples along the



Figure 4.26: Down-sampling factor  $t=2$

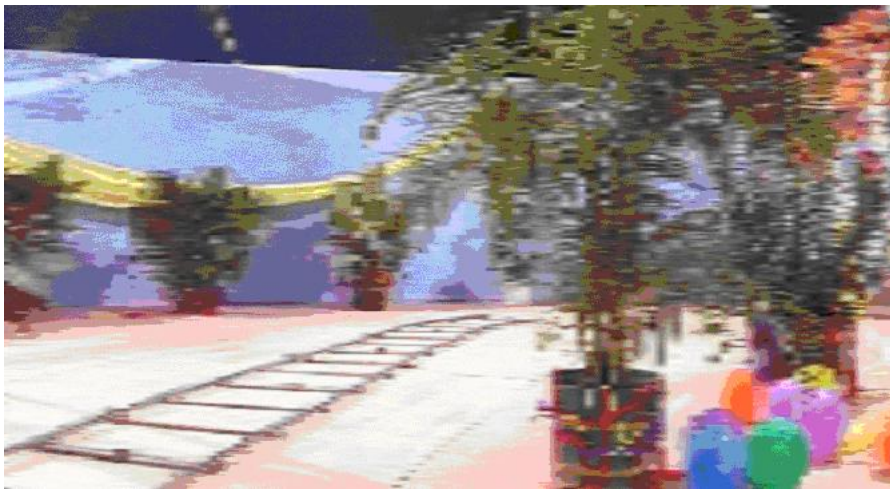


Figure 4.27: Down-sampling factor  $t=3$



rotation direction is reduced.

## Chapter 5

# Rendering of Stereo Views in the Concentric Mosaics Technique

Two main factors prevent the wide application of stereo images and videos. The first one is in the capturing procedure, since special devices or techniques must be employed to record both the left eye view and the right eye view at any viewing position. The other one is the method to view the stereo images. Special devices are required to view the stereoscopic images to guarantee that the left eye sees only the left view and the right eye sees only the right one. The viewing devices are different in principle with each other and can be classified as shuttered glasses, polarized glasses, and colored glasses, etc.

The advantage of stereo views over monoscopic ones is the perception of depth, thus providing a more realistic feeling, which is very attractive for some multimedia applications. The recently developed Image-Based Rendering technology aims toward the synthesis of arbitrary views of an environment within a certain navigation area. Thus the synthesis, rather than capturing, of the left and right views of the stereo image pair makes the stereo images easy to be produced based on the same pre-captured image database.

Among the several stereo image viewing methods, the technique using shutter

glasses is the most advanced method and provides the best stereoscopic effect, although good shutter glass technology can be quite expensive. On the other hand, the technique using colored glasses (anaglyph) is the simplest one with negligible cost compared to other methods, and a new method to generate anaglyph stereo images with strong stereo effect has been proposed in [21]. A sample pair of anaglyph glasses is attached to the thesis.

In this chapter, the visualization of the stereo pairs is mathematically represented, especially in the optical spectrum domain. The system for viewing stereo images on a monitor using shutter glasses is introduced, with the stereo image pairs synthesized through the Concentric Mosaics rendering technique.

A new method to generate anaglyph images is derived using projection theory to solve the optimization problem [21]. This new method includes the mathematical modelling on the visualization of the stereo pairs in the spectrum domain, the mathematical representation of the anaglyph technique, the optimization problem to generate anaglyph images and its projection solution. Section 5.1 and 5.2 are mainly the reproduction from [21], with some extensions on the color recovery and intensity studies.

Then we will combine the Concentric Mosaics technique with the anaglyph technique, which makes the stereoscopic application flexible for any ordinary user. An algorithm for fast rendering of stereoscopic views [19] is proposed based on the Concentric Mosaics technique, through the pre-processing of the image data. Finally, the stereo effect of the anaglyph images is compared with that using shutter glasses through informal subjective evaluations.

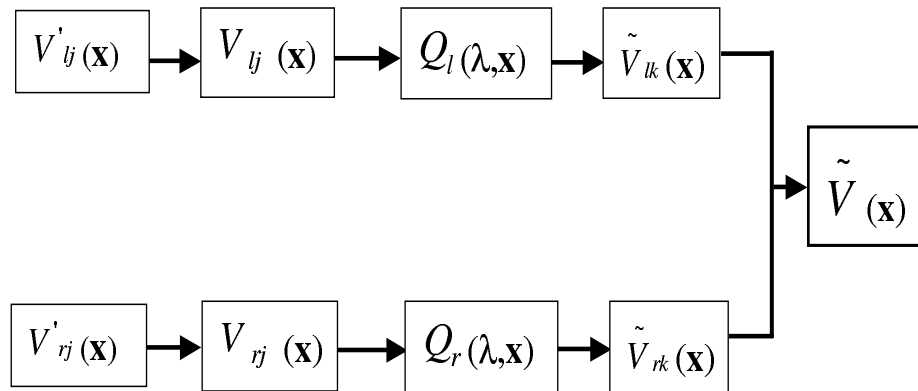


Figure 5.1: Visualization of the stereo images

## 5.1 The Visualization of Stereo Views on a Monitor

The visualization of stereo pairs is studied through representation of the stereo views in mathematical format. The filtering effects in the spectrum during the visualization procedure are studied in order to derive the new projection method to generate analyph images in the next section. A shutter-glasses-based system to view stereo images on a monitor is implemented, in which the left and the right image of a stereo pair are displayed on the monitor alternately.

### 5.1.1 Visualization of the input stereo pair

The visualization of a stereo image pair is shown in Fig. 5.1. A true-color stereo pair  $V'_{lj}(\mathbf{x})$  and  $V'_{rj}(\mathbf{x})$  are the input left view and right view respectively. The subscripts  $l$  and  $r$  denote left and right,  $j = 1, 2, 3$  and  $\mathbf{x} \in \mathcal{L}$ . It is assumed that the three components  $j = 1, 2, 3$  are gamma-corrected RGB (in that order) that can be directly displayed on a standard CRT monitor; the 'prime' symbol denotes gamma-corrected signals.  $\mathcal{L}$  is the sampling raster for the image which is arbitrary and can be either spatial or spatiotemporal. Thus each image of a stereo pair is represented by samples

in a three-dimensional space.

After going through the display gamma, which is denoted by the function  $g(\cdot)$ , the three components of left image  $V_{lj}(\mathbf{x})$  and right image  $V_{rj}(\mathbf{x})$  excite the display RGB phosphors.

$$V_{lj}(\mathbf{x}) = g(V'_{lj}(\mathbf{x})) \quad (5.1)$$

$$V_{rj}(\mathbf{x}) = g(V'_{rj}(\mathbf{x})) \quad (5.2)$$

The spectral density  $Q_l(\lambda, \mathbf{x})$  and  $Q_r(\lambda, \mathbf{x})$  of the light emanating from point  $\mathbf{x}$  in the left and right images are the result of the left image  $V_{lj}(\mathbf{x})$  and the right image  $V_{rj}(\mathbf{x})$ , weighting the spectral density functions of the RGB display phosphors,

$$Q_l(\lambda, \mathbf{x}) = \sum_{j=1}^3 V_{lj}(\mathbf{x}) d_j(\lambda), \quad (5.3)$$

$$Q_r(\lambda, \mathbf{x}) = \sum_{j=1}^3 V_{rj}(\mathbf{x}) d_j(\lambda), \quad (5.4)$$

where  $d_j(\lambda)$ ,  $j = 1, 2, 3$  denote the spectral density functions of the RGB display phosphors, which are different for different phosphors used.

The final color perceived by a human observer at point  $\mathbf{x}$  in the left and right images is determined by the projection of  $Q_l(\lambda, \mathbf{x})$  and  $Q_r(\lambda, \mathbf{x})$  onto the visual subspace using color-matching functions  $\bar{p}_k(\lambda)$  for the chosen set of primaries. For the left view,

$$\begin{aligned} \tilde{V}_{lk}(\mathbf{x}) &= \int Q_l(\lambda, \mathbf{x}) \bar{p}_k(\lambda) d\lambda \\ &= \sum_{j=1}^3 V_{lj}(\mathbf{x}) \int \bar{p}_k(\lambda) d_j(\lambda) d\lambda \\ &= \sum_{j=1}^3 c_{kj} V_{lj}(\mathbf{x}), \quad k = 1, 2, 3. \end{aligned} \quad (5.5)$$

The integral is over the wavelengths of the visible spectrum, approximately 370 nm to 730 nm.

Thus, in matrix notation,

$$\tilde{\mathbf{V}}_l(\mathbf{x}) = \mathbf{C}\mathbf{V}_l(\mathbf{x}) \quad (5.6)$$

where

$$\tilde{\mathbf{V}}_l(\mathbf{x}) = \left[ \tilde{V}_{l1}(\mathbf{x}) \quad \tilde{V}_{l2}(\mathbf{x}) \quad \tilde{V}_{l3}(\mathbf{x}) \right]^T \quad (5.7)$$

$$\mathbf{V}_l(\mathbf{x}) = \left[ V_{l1}(\mathbf{x}) \quad V_{l2}(\mathbf{x}) \quad V_{l3}(\mathbf{x}) \right]^T \quad (5.8)$$

and

$$[\mathbf{C}]_{kj} = c_{kj} = \int \bar{p}_k(\lambda) d_j(\lambda) d\lambda. \quad (5.9)$$

Similarly, for the right view,

$$\tilde{\mathbf{V}}_r(\mathbf{x}) = \mathbf{C}\mathbf{V}_r(\mathbf{x}) \quad (5.10)$$

In an ideal stereoscopic visualization system, the left eye sees only the image defined by  $\tilde{\mathbf{V}}_l(\mathbf{x})$  and the right eye sees only the image defined by  $\tilde{\mathbf{V}}_r(\mathbf{x})$ .

Thus, the value of the stereoscopic image at each point  $\mathbf{x}$ , which is visualized by the human being, can be considered to be an element of a six-dimensional vector space  $\mathcal{S}_6$ . Arranged as a column matrix, we have

$$\tilde{\mathbf{V}}(\mathbf{x}) = \left[ \tilde{V}_{l1}(\mathbf{x}) \quad \tilde{V}_{l2}(\mathbf{x}) \quad \tilde{V}_{l3}(\mathbf{x}) \quad \tilde{V}_{r1}(\mathbf{x}) \quad \tilde{V}_{r2}(\mathbf{x}) \quad \tilde{V}_{r3}(\mathbf{x}) \right]^T. \quad (5.11)$$

We can form a basis for this space using the columns of  $\mathbf{C}$  as follows:

$$\begin{aligned} \mathbf{c}_{li} &= \left[ c_{1i} \quad c_{2i} \quad c_{3i} \quad 0 \quad 0 \quad 0 \right]^T \\ \mathbf{c}_{ri} &= \left[ 0 \quad 0 \quad 0 \quad c_{1i} \quad c_{2i} \quad c_{3i} \right]^T \\ &i = 1, 2, 3 \end{aligned} \quad (5.12)$$

In terms of this basis, we have

$$\tilde{\mathbf{V}}(\mathbf{x}) = \sum_{j=1}^3 V_{lj}(\mathbf{x}) \mathbf{c}_{lj} + \sum_{j=1}^3 V_{rj}(\mathbf{x}) \mathbf{c}_{rj}. \quad (5.13)$$

If we define the  $6 \times 6$  matrix

$$\mathbf{C}_2 = \left[ \mathbf{c}_{l1} \quad \mathbf{c}_{l2} \quad \mathbf{c}_{l3} \quad \mathbf{c}_{r1} \quad \mathbf{c}_{r2} \quad \mathbf{c}_{r3} \right] = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} \quad (5.14)$$

then we can write in matrix form

$$\tilde{\mathbf{V}}(\mathbf{x}) = \mathbf{C}_2 \mathbf{V}(\mathbf{x}). \quad (5.15)$$

The set of realizable stereoscopic images has values that lie in the convex subset of  $\mathcal{S}_6$

$$\left\{ \sum_{j=1}^3 v_{lj} \mathbf{c}_{lj} + \sum_{j=1}^3 v_{rj} \mathbf{c}_{rj} \mid 0 \leq v_{lj} \leq 1, 0 \leq v_{rj} \leq 1, j = 1, 2, 3 \right\}. \quad (5.16)$$

### 5.1.2 The viewing and rendering of stereo views

In an ideal stereoscopic visualization system, the left eye sees only the left view  $\tilde{\mathbf{V}}_l(\mathbf{x})$  and the right eye sees only the right view  $\tilde{\mathbf{V}}_r(\mathbf{x})$ . This can not be achieved unless special methods have been adopted because the images displayed on a monitor will be viewed by both the eyes of a viewer at the same time.

Many methods have been proposed to view stereo images on a monitor. Generally, the principles are separating the left views and the right views in the spatial domain, the time domain or the spectrum domain. One of the methods to separate the left and right images in the spectrum domain is the anaglyph technique, which will be further studied in the next section.

The method to separate the left image and the right image in the spatial domain is simple. The screen of a monitor is divided into two parts, the left part and the right part, and the left and right image are displayed on the left and right part of the monitor, respectively. Special techniques such as mechanical structures are required to guarantee the left eye only sees the left part of the monitor and the right eye only sees the right part.

The most advanced technique to view the stereo images on a monitor separates the left image and right image in the time domain with viewing through a pair of shutter glasses. The left and right images are alternately displayed on a monitor. When the left image is displayed on the monitor, the right part of the glasses turns opaque and blocks the light to the right eye and the left part of the glass remains transparent and lets the light pass through. Thus only the left eye can see the image

on the monitor at this moment. When the right image is displayed on the monitor, the inverse state of the left and right part of the glasses is activated to guarantee the right eye sees the right image and left eye sees nothing. The lenses of the glasses turn transparent and opaque alternatingly acting as a shutter to the coming light. One material that has this property, turning transparent and opaque under different electrical conditions, is liquid crystal.

A liquid crystal glasses based viewing system should include:

- a special display card with the stereo-support features and with high refresh rate in order to display the left and right view alternatingly at the same time maintaining a proper refresh rate for each eye to avoid flicker.
- control equipment which is for the synchronization between the alternating display and the switching of the glasses' states.( One state is that the left part is transparent and the right part is opaque; the other is the left part is opaque and the right part is transparent.)
- a pair of liquid-crystal glasses for each observer.

An Oxygen GVX420 video card, and a Stereographics ENT-B emitter as the wireless remote control equipment are used in our liquid crystal glasses-based viewing system. Each viewer must wear a pair of glasses, the CrystalEyes model made by Stereographics Corporation. The emitter and the glasses are the product of the same corporation in order to match with each other.

The rendering of stereo image pairs is convenient for the Image-Based Rendering technique. The two eyes' views are rendered instead of captured, as long as the distance between two eyes is given. Two images are rendered at the same time for left eye and right eye and then they are displayed alternatingly on the monitor. However, the system may be too expensive or specialized for ordinary users, which will definitely limit its broad application.

In the following section, we will study the anaglyph technique, by which the stereo images are viewed on an ordinary monitor with a pair of colored glasses.



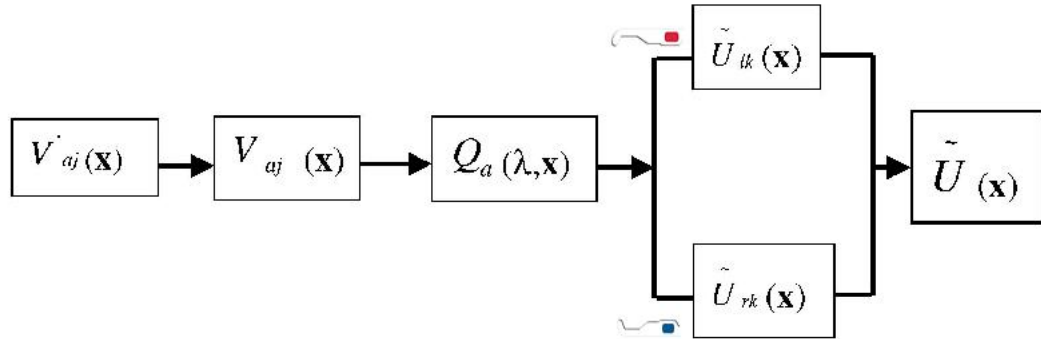


Figure 5.2: Visualization of the anaglyph images

## 5.2 The Anaglyph Technique

Compared with using a pair of shuttered glasses to view stereo images on a monitor, the anaglyph technique using a pair of colored glasses is much easier to apply for a personal computer user. A projection method to generate anaglyph stereo images with strong stereo effect was proposed in [21], which uses the spectral absorption curves of the glasses, the spectral density functions of the display primaries and the colorimetric properties of the human observer.

### 5.2.1 Visualization of an anaglyph image

The procedure of visualizing an anaglyph image can be described in Fig. 5.2. Instead of a pair of images for left view and right view, only a single anaglyph image  $V'_{aj}(\mathbf{x})$  is input into the visualizing system. The display gamma and the filtering by the spectral density of the RGB display phosphors is the same process as described in the previous section. However, the light  $Q_a(\lambda, \mathbf{x})f_l(\lambda)$  and  $Q_a(\lambda, \mathbf{x})f_r(\lambda)$  entering left eye and right eye has been filtered through a pair of spectrum complementary filters with spectral absorption functions  $f_l(\lambda)$  and  $f_r(\lambda)$ , respectively. Thus the stereoscopic view  $\tilde{U}(\mathbf{x})$  is formed, which is also represented in a six-dimensional vector space.

Assume that the three components of the anaglyph image  $V'_{aj}(\mathbf{x})$ ,  $j = 1, 2, 3$ ,  $\mathbf{x} \in \mathcal{L}$  are in the same gamma-corrected RGB display primary system as the stereo pair in the previous section. After going through the display gamma,

$$V_{aj}(\mathbf{x}) = g(V'_{aj}(\mathbf{x})), j = 1, 2, 3. \quad (5.17)$$

The spectral density of the light emitted from the screen at  $\mathbf{x}$  is given by

$$Q_a(\lambda, \mathbf{x}) = \sum_{j=1}^3 V_{aj}(\mathbf{x}) d_j(\lambda) \quad (5.18)$$

The light from the CRT passes through two filters with spectral absorption functions  $f_l(\lambda)$  and  $f_r(\lambda)$  before arriving at the left and right eyes respectively. Thus the light spectral distribution at the left and right eyes is  $Q_a(\lambda, \mathbf{x})f_l(\lambda)$  and  $Q_a(\lambda, \mathbf{x})f_r(\lambda)$  respectively. The corresponding sets of XYZ tristimulus values are

$$\begin{aligned} \tilde{U}_{lk}(\mathbf{x}) &= \int Q_a(\lambda, \mathbf{x}) f_l(\lambda) \bar{p}_k(\lambda) d\lambda \\ &= \sum_{j=1}^3 V_{aj}(\mathbf{x}) \int \bar{p}_k(\lambda) d_j(\lambda) f_l(\lambda) d\lambda \\ &= \sum_{j=1}^3 a_{lkj} V_{aj}(\mathbf{x}) \end{aligned} \quad (5.19)$$

or in matrix form  $\tilde{\mathbf{U}}_l(\mathbf{x}) = \mathbf{A}_l \mathbf{V}_a(\mathbf{x})$ , where

$$\tilde{\mathbf{V}}_a(\mathbf{x}) = \left[ \tilde{V}_{a1}(\mathbf{x}) \quad \tilde{V}_{a2}(\mathbf{x}) \quad \tilde{V}_{a3}(\mathbf{x}) \right]^T \quad (5.20)$$

$$[\mathbf{A}_l]_{kj} = a_{lkj} = \int \bar{p}_k(\lambda) d_j(\lambda) f_l(\lambda) d\lambda. \quad (5.21)$$

and

$$\tilde{\mathbf{U}}_l(\mathbf{x}) = \left[ \tilde{U}_{l1}(\mathbf{x}) \quad \tilde{U}_{l2}(\mathbf{x}) \quad \tilde{U}_{l3}(\mathbf{x}) \right]^T. \quad (5.22)$$

Similarly,  $\tilde{\mathbf{U}}_r(\mathbf{x}) = \mathbf{A}_r \mathbf{V}_a(\mathbf{x})$ , where

$$[\mathbf{A}_r]_{kj} = a_{rkj} = \int \bar{p}_k(\lambda) d_j(\lambda) f_r(\lambda) d\lambda, \quad (5.23)$$

and

$$\tilde{\mathbf{U}}_r(\mathbf{x}) = \left[ \tilde{U}_{r1}(\mathbf{x}) \quad \tilde{U}_{r2}(\mathbf{x}) \quad \tilde{U}_{r3}(\mathbf{x}) \right]^T. \quad (5.24)$$

The goal is for the stereo pair perceived by viewing  $\tilde{\mathbf{U}}_l(\mathbf{x})$  and  $\tilde{\mathbf{U}}_r(\mathbf{x})$ ,  $\mathbf{x} \in \mathcal{L}$ , to be as similar as possible to the ideal one perceived by viewing  $\tilde{\mathbf{V}}_l(\mathbf{x})$  and  $\tilde{\mathbf{V}}_r(\mathbf{x})$ ,  $\mathbf{x} \in \mathcal{L}$ .

It is impossible to make  $\tilde{\mathbf{U}}_l(\mathbf{x}) = \tilde{\mathbf{V}}_l(\mathbf{x})$  and  $\tilde{\mathbf{U}}_r(\mathbf{x}) = \tilde{\mathbf{V}}_r(\mathbf{x})$  in general, since the filters  $f_l(\lambda)$  and  $f_r(\lambda)$  each block certain wavelength bands. Specifically, if we want to reproduce a feature that is dark in the left view and bright in the right view due to disparity, the light emitted at point  $\mathbf{x}$  must lie mostly in the stopband of the left filter and in the passband of the right filter. Thus, the two filters must be spectrum-complementary in some way.

The stereoscopic image values formed by viewing the anaglyph image through spectacles with the colored filters also lie in the six-dimensional space  $\mathcal{S}_6$ . However, they are constrained to lie in a three-dimensional subspace. Define the following three vectors in  $\mathcal{S}_6$ :

$$\mathbf{r}_j = \left[ a_{l1j} \quad a_{l2j} \quad a_{l3j} \quad a_{r1j} \quad a_{r2j} \quad a_{r3j} \right]^T, \quad j = 1, 2, 3. \quad (5.25)$$

Then

$$\tilde{\mathbf{U}}(\mathbf{x}) = \sum_{j=1}^3 V_{aj}(\mathbf{x}) \mathbf{r}_j \quad (5.26)$$

which lies in  $\mathcal{R} = \text{span}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ , a three-dimensional subspace of  $\mathcal{S}_6$ . The set of all *realizable* anaglyph stereoscopic images lies in the convex subset of  $\mathcal{R}$

$$\left\{ \sum_{j=1}^3 v_{aj} \mathbf{r}_j \mid 0 \leq v_{aj} \leq 1, j = 1, 2, 3 \right\}. \quad (5.27)$$

If we define the matrix

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_l \\ \mathbf{A}_r \end{bmatrix} \quad (5.28)$$

then equation (5.26) can be expressed in matrix form as

$$\tilde{\mathbf{U}}(\mathbf{x}) = \mathbf{R} \mathbf{V}_a(\mathbf{x}). \quad (5.29)$$

where  $\mathbf{R}$  is  $6 \times 3$  matrix and  $\mathbf{V}_a$  is a  $3 \times 1$  matrix, thus ending up with a  $6 \times 1$  matrix for  $\tilde{\mathbf{U}}(\mathbf{x})$ ,

$$\tilde{\mathbf{U}}(\mathbf{x}) = \left[ \tilde{U}_{l1}(\mathbf{x}) \quad \tilde{U}_{l2}(\mathbf{x}) \quad \tilde{U}_{l3}(\mathbf{x}) \quad \tilde{U}_{r1}(\mathbf{x}) \quad \tilde{U}_{r2}(\mathbf{x}) \quad \tilde{U}_{r3}(\mathbf{x}) \right]^T \quad (5.30)$$

### 5.2.2 Optimization problem with projection solution

The formation of an anaglyph image can now be posed as an optimization problem: given a stereoscopic pair  $\mathbf{V}_l(\mathbf{x})$ ,  $\mathbf{V}_r(\mathbf{x})$ ,  $\mathbf{x} \in \mathcal{L}$ , we seek an anaglyph image  $\mathbf{V}_a(\mathbf{x})$ ,  $\mathbf{x} \in \mathcal{L}$  with  $0 \leq V_{aj}(\mathbf{x}) \leq 1$  such that the perceived image  $\tilde{\mathbf{U}}$  is as similar as possible as the input image  $\tilde{\mathbf{V}}$ . The ideal solution of  $\mathbf{V}_a(\mathbf{x})$  should be obtained by optimization based on an error metric, which computes numerically the subjective difference between a stereo pair and an anaglyph type approximation. However, there is no such error metric and it is not easy to define because it will be based on many subjective experiments.

Although we can not let  $\tilde{\mathbf{U}}$  equal  $\tilde{\mathbf{V}}$  because of the spectral characteristics of the two filters, we still can solve the problem by minimizing the weighted distance between  $\tilde{\mathbf{U}}$  and  $\tilde{\mathbf{V}}$  based on following assumptions:

- The approximation is carried out independently at each sample location;
- The error metric at each point is a weighted squared error between  $\tilde{\mathbf{U}}$  and  $\tilde{\mathbf{V}}$ ;
- A global scaling of the  $V_{aj}$  is used to account for the attenuation of the filters.

The last assumption lets us take into account the reduction in luminance due to the overall attenuation of the filters.

Given these assumptions, the  $V_{aj}(\mathbf{x})$  are determined by applying the projection theorem. In order to apply the projection theorem, an inner product on  $\mathcal{S}_6$  is defined to obtain a suitable distance measure with the resulting norm. A general inner product has the form

$$\langle \mathbf{v}_1 | \mathbf{v}_2 \rangle = \mathbf{v}_1^T \mathbf{W} \mathbf{v}_2 \quad (5.31)$$

where  $\mathbf{W}$  is a positive-definite matrix. The corresponding norm is

$$\|\mathbf{v}\|^2 = \langle \mathbf{v} | \mathbf{v} \rangle = \mathbf{v}^T \mathbf{W} \mathbf{v}. \quad (5.32)$$

If  $\mathbf{W} = \mathbf{I}$ , the  $6 \times 6$  identity matrix, this results in a familiar Euclidean distance in the Cartesian product of the XYZ space with itself. Use of other diagonal matrices  $\mathbf{W}$  can allow weighting of the X, Y or Z component more heavily than other two components. Non-diagonal weighting matrices can correspond to distances with respect to other sets of primaries than XYZ. The diagonal matrix  $\mathbf{W}$  in our application is introduced to allow weighting of the Y component more heavily than X and Z to favor reproduction of the correct luminance at the expense of greater color errors.

The projection approach is then to determine for each  $\mathbf{x}$  the element of  $\mathcal{R}$  that is closest in the sense of the chosen norm to  $\tilde{\mathbf{V}}(\mathbf{x})$ , i.e., find  $\hat{\mathbf{V}}_a(\mathbf{x})$  such that  $\|\tilde{\mathbf{V}}(\mathbf{x}) - \sum_{j=1}^3 \hat{\mathbf{V}}_{aj}(\mathbf{x}) \mathbf{r}_j\|$  is minimized.

The method to solve the minimization problem using the projection theory is standard. By introducing the  $3 \times 3$  Grammian matrix  $\Phi$ ,

$$\Phi = \begin{bmatrix} \langle \mathbf{r}_1 | \mathbf{r}_1 \rangle & \langle \mathbf{r}_2 | \mathbf{r}_1 \rangle & \langle \mathbf{r}_3 | \mathbf{r}_1 \rangle \\ \langle \mathbf{r}_1 | \mathbf{r}_2 \rangle & \langle \mathbf{r}_2 | \mathbf{r}_2 \rangle & \langle \mathbf{r}_3 | \mathbf{r}_2 \rangle \\ \langle \mathbf{r}_1 | \mathbf{r}_3 \rangle & \langle \mathbf{r}_2 | \mathbf{r}_3 \rangle & \langle \mathbf{r}_3 | \mathbf{r}_3 \rangle \end{bmatrix} \quad (5.33)$$

and the  $3 \times 1$  matrix  $\beta(\mathbf{x})$ ,

$$\beta(\mathbf{x}) = \begin{bmatrix} \langle \mathbf{r}_1 | \tilde{\mathbf{V}}(\mathbf{x}) \rangle \\ \langle \mathbf{r}_2 | \tilde{\mathbf{V}}(\mathbf{x}) \rangle \\ \langle \mathbf{r}_3 | \tilde{\mathbf{V}}(\mathbf{x}) \rangle \end{bmatrix}, \quad (5.34)$$

the projection is given by

$$\hat{\mathbf{V}}_a(\mathbf{x}) = \Phi^{-1} \beta(\mathbf{x}). \quad (5.35)$$

Further details on the projection theory can be found in [33]. Using the previously defined matrix  $\mathbf{R}$ , these equations can be expressed as

$$\begin{aligned} \hat{\mathbf{V}}_a(\mathbf{x}) &= (\mathbf{R}^T \mathbf{W} \mathbf{R})^{-1} \mathbf{R}^T \mathbf{W} \tilde{\mathbf{V}}(\mathbf{x}) \\ &= (\mathbf{R}^T \mathbf{W} \mathbf{R})^{-1} \mathbf{R}^T \mathbf{W} \mathbf{C}_2 \mathbf{V}(\mathbf{x}). \end{aligned} \quad (5.36)$$

since,

$$\Phi = \mathbf{R}^T \mathbf{W} \mathbf{R} \quad (5.37)$$

$$\beta(\mathbf{x}) = \mathbf{R}^T \mathbf{W} \tilde{\mathbf{V}}(\mathbf{x}) \quad (5.38)$$

Note that the  $3 \times 6$  matrix  $(\mathbf{R}^T \mathbf{W} \mathbf{R})^{-1} \mathbf{R}^T \mathbf{W} \mathbf{C}_2$  is fixed and can be precomputed.

The resulting components  $\hat{V}_{aj}$  will not in general lie in the required interval  $[0,1]$ , and thus normalization is usually required. The normalization is applied by inputting the possible maximal image into the calculation system, or equation (5.36), to obtain a weighted matrix. The matrix is then used to adjust the XYZ components of the generated anaglyph images.

The possible maximal images of a stereo pair to generate an anaglyph image in our representation are two uniform white images for both the left and right view, respectively

$$\mathbf{V}_l(\mathbf{x}) = \mathbf{V}_r(\mathbf{x}) = [1 \ 1 \ 1]^T \quad (5.39)$$

thus

$$\mathbf{V}(\mathbf{x}) = \mathbf{E} = [1 \ 1 \ 1 \ 1 \ 1 \ 1]^T. \quad (5.40)$$

If we use this image pair to generate an anaglyph image, we obtain

$$\hat{\mathbf{E}}_a = (\mathbf{R}^T \mathbf{W} \mathbf{R})^{-1} \mathbf{R}^T \mathbf{W} \mathbf{C}_2 \mathbf{E} \quad (5.41)$$

through the projection equation (5.36).

Ideally, the maximal input image pair should correspond to the maximal anaglyph image  $\mathbf{V}_{aw}$ ,

$$\mathbf{V}_{aw} = [1 \ 1 \ 1]^T. \quad (5.42)$$

Using equation (5.41) and equation (5.42), the diagonal normalizing matrix for premultiplying is,

$$\mathbf{N} = \text{diag}(V_{awj} / \hat{E}_{aj}). \quad (5.43)$$

Thus with normalization included, the anaglyph image is given by

$$\begin{aligned} \hat{\mathbf{V}}_a(\mathbf{x}) &= \mathbf{N} (\mathbf{R}^T \mathbf{W} \mathbf{R})^{-1} \mathbf{R}^T \mathbf{W} \mathbf{C}_2 \mathbf{V}(\mathbf{x}) \\ &= \mathbf{P} \mathbf{V}(\mathbf{x}). \end{aligned} \quad (5.44)$$

In this case the fixed  $3 \times 6$  matrix  $\mathbf{P} = \mathbf{N}(\mathbf{R}^T \mathbf{W} \mathbf{R})^{-1} \mathbf{R}^T \cdot \mathbf{W} \mathbf{C}_2$  can be precomputed. The final step is clipping to the range  $[0,1]$  and application of gamma correction.

### 5.2.3 Simulation of color recovery and intensity disparity of the left and right views

In the procedure for generating anaglyph images, the spectral characteristics of the original images are modified and therefore color distortion is usually unavoidable. In this section, we will study the color recovery problem and the intensity disparity of the left and right views.

We use the XYZ coordinate system, and so define the color-matching functions  $\bar{p}_k(\lambda)$ ,  $k = 1, 2, 3$  to be the standard  $\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$  and  $\bar{z}(\lambda)$  respectively. These functions are tabulated and graphed in [34]. The display phosphor densities for a Sony Trinitron monitor are used in our experiment to obtain the matrix C in equation (5.9).

$$\mathbf{C} = \begin{bmatrix} 0.4641 & 0.3055 & 0.1808 \\ 0.2597 & 0.6592 & 0.0811 \\ 0.0357 & 0.1421 & 0.9109 \end{bmatrix} \quad (5.45)$$

This matrix is similar to standard ones for converting from various RGB spaces to XYZ, e.g., the Rec. 709 matrix on page 148 of [35], but is slightly different. If different phosphors are used, a different matrix would result.

The spectral transmission curves  $f_l(\lambda)$  and  $f_r(\lambda)$  for the red and blue filters were measured on a pair of commercial anaglyph glasses using a spectrophotometer and the results are shown in Fig. 5.3. Assuming that the red filter is on the left and that the blue filter is on the right, the matrices  $\mathbf{A}_l$  and  $\mathbf{A}_r$  corresponding to these filters are given by

$$\mathbf{A}_l = \begin{bmatrix} 0.2564 & 0.0273 & 0.0058 \\ 0.1143 & 0.0161 & 0.0026 \\ 0.0002 & 0.0003 & 0.0016 \end{bmatrix} \quad (5.46)$$

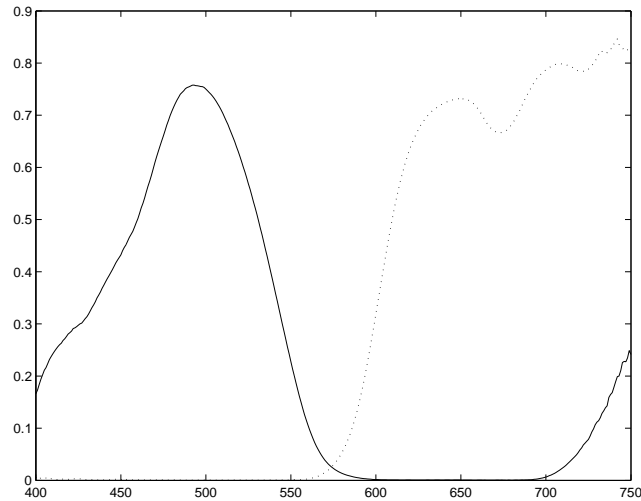


Figure 5.3: Transmission of a pair of commercial anaglyph glasses as a function of wavelength

$$\mathbf{A}_r = \begin{bmatrix} 0.0068 & 0.0502 & 0.0731 \\ 0.0175 & 0.2269 & 0.0426 \\ 0.0179 & 0.0859 & 0.4159 \end{bmatrix}. \quad (5.47)$$

We select white as our simulation color. The simulations are carried out by inputting two white views with different intensity, or

$$\mathbf{V}_l(\mathbf{x}) = [d d d]^T \quad (5.48)$$

$$\mathbf{V}_r(\mathbf{x}) = [1 1 1]^T \quad (5.49)$$

and thus

$$\mathbf{V}(\mathbf{x}) = \mathbf{E} = [d d d 1 1 1]^T \quad (5.50)$$

As  $d$  changes around 1, we input  $\mathbf{V}(\mathbf{x})$  to generate the anaglyph image and then calculate the left view and right view perceived by left and right eyes, respectively to study several issues:

(1) The color recovery of the anaglyph image through calculating its color coordinates, with the result shown in Fig. 5.4. In the figure, the solid line represents the  $x$  coordinates, the dashed line represents the  $y$  coordinates and the dotted line



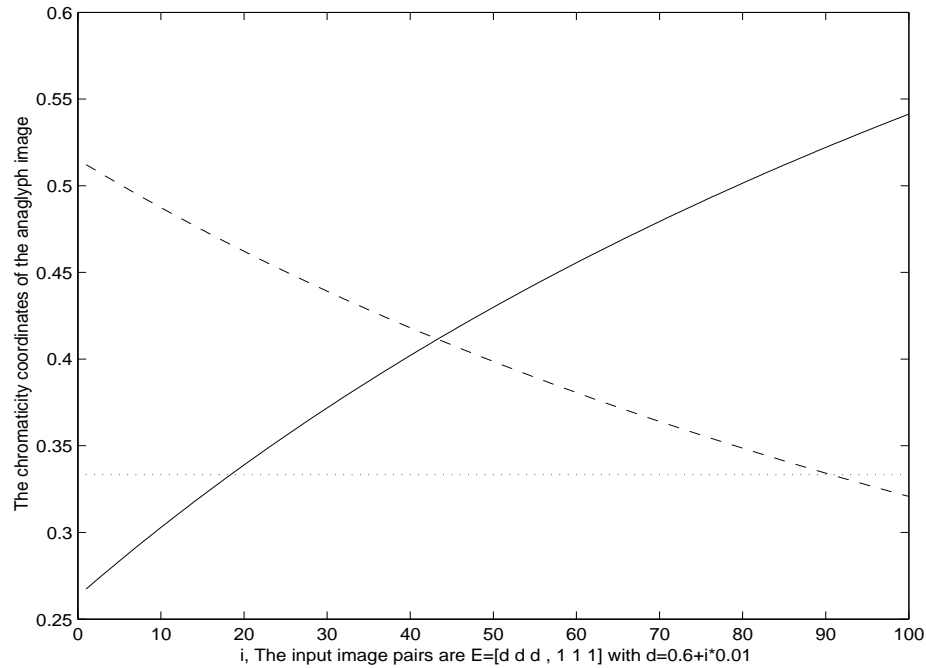


Figure 5.4: The study of color recovery of the anaglyph images (The solid line represents the  $x$  chromaticity coordinates, the dashed line represents the  $y$  chromaticity coordinates of the generated anaglyph images in the XYZ colorimetric system and the dotted line represents the white coordinates with  $x, y = 0.3333$  for reference white.)

represents the white coordinates with  $x, y = 0.3333$ . Assuming a pair of uniform white images are the left and right views, the generated anaglyph image should also produce a white image if the color recovery is perfect. By searching the minimal distance from the white point ( $x = 0.3333, y = 0.3333, z = 0.3333$ ), we get  $d = 0.88$ . Thus if we reduce the left input image by a coefficient 0.88, we will get the best color recovery of the anaglyph image.

(2) The intensity difference between left and right views through the calculation of the intensity perceived by left and right eyes, respectively. The result is shown in Fig. 5.5 with solid line for left view and dashed line for right indicates that the intensity disparities between two eyes' views are quite large.

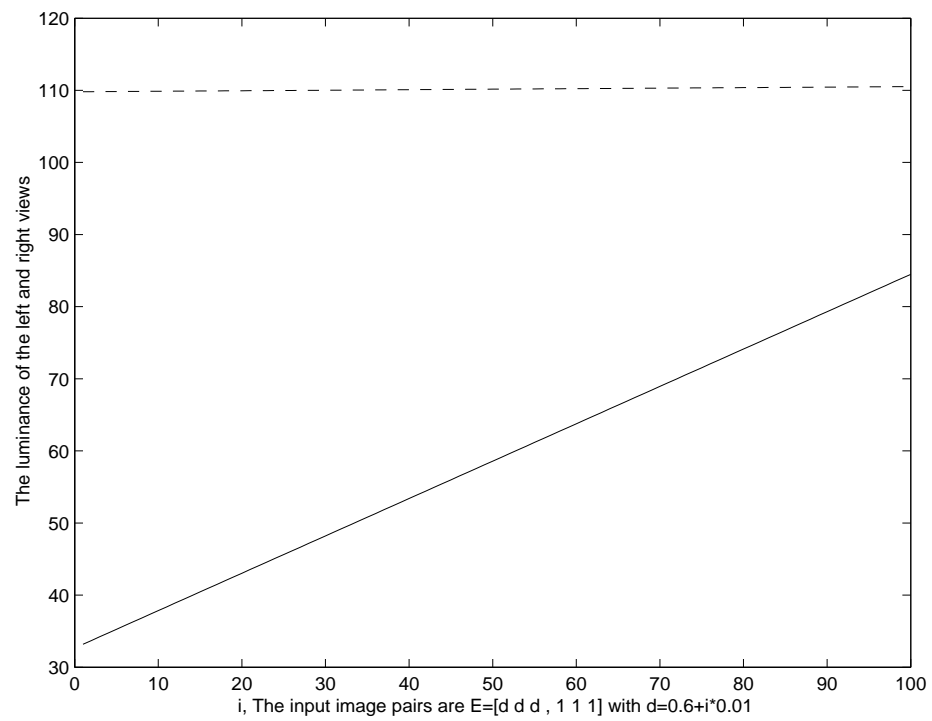


Figure 5.5: The study of intensity disparity of the left and right views (solid line represents the luminance of left views and dashed line represents the luminance of right views)

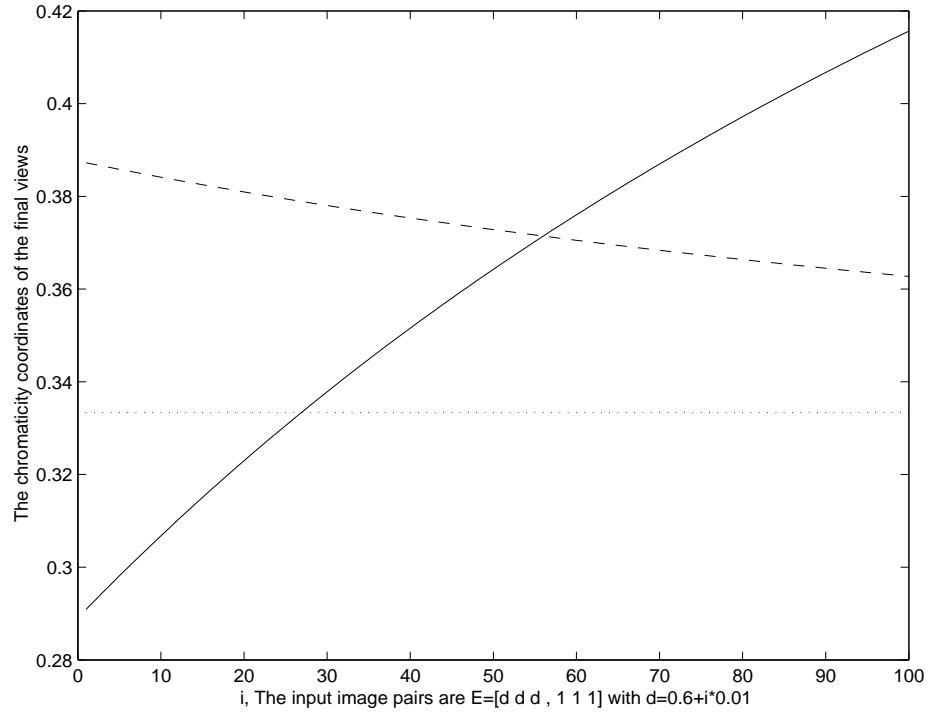


Figure 5.6: The study of color recovery of the final views (The solid line represents the  $x$  chromaticity coordinates, the dashed line represents the  $y$  chromaticity coordinates of the finally perceived views in the XYZ colorimetric system and the dotted line represents the white coordinates with  $x, y = 0.3333$  for reference white.)

(3) The color recovery of the perceived views by calculating the color coordinates of the final views: the addition of the left and right views. The result is shown in Fig. 5.6, with the solid line representing the  $x$  coordinates, the dashed line representing the  $y$  coordinates and the dotted line representing the white coordinates with  $x, y = 0.3333$ . Also, by searching the minimal distance from the white point ( $x = 0.3333, y = 0.3333, z = 0.3333$ ), we get  $d = 0.76$ . Thus if we reduce the left input image by a coefficient 0.76, we will get the best color recovery of the final view.



Figure 5.7: The left view (reduced size)

#### 5.2.4 Simulation results on generating anaglyph images

The simulations on generating anaglyph images are carried out using a pair of images shown in Fig. 5.7 and Fig. 5.8. Fig. 5.9, Fig. 5.10 and Fig. 5.11 show the results, with the coefficients  $d = 1, 0.88, 0.76$  respectively to modify the left input image.

From the simulation results, we can find that the colors of the anaglyph image with  $d = 0.88$  are very similar with the original images, which is coincident with our conclusions in the last section. The stereo effects of these three images are very similar through the subjective observations. Thus, the multiplication by a factor  $d = 0.88$  to the left input image to generate anaglyph images can produce anaglyph images and maintain the original color of the scene maximally.



Figure 5.8: The right view (reduced size)

### **5.3 The Fast Rendering of Anaglyph Views in the Concentric Mosaics Technique**

To synthesize one stereo view, two images for the left and right eye must be generated separately, so it requires more processing time to synthesize a stereo view than a monoscopic one. Moreover, the processing time to generate an anaglyph image is even longer. Thus the processing time for the combination of Image-Based Rendering technique and the anaglyph technique will be a challenge to its practical real time application.

In this section, an algorithm is proposed in which the pre-captured images are first pre-processed from ordinary images to anaglyph images. Then the rendering of the stereo images is just like that of any arbitrary monoscopic image. Thus the synthesizing time for a stereo image is no more than that of the synthesis of any monoscopic image, which is even less than that of the synthesis of two separate views

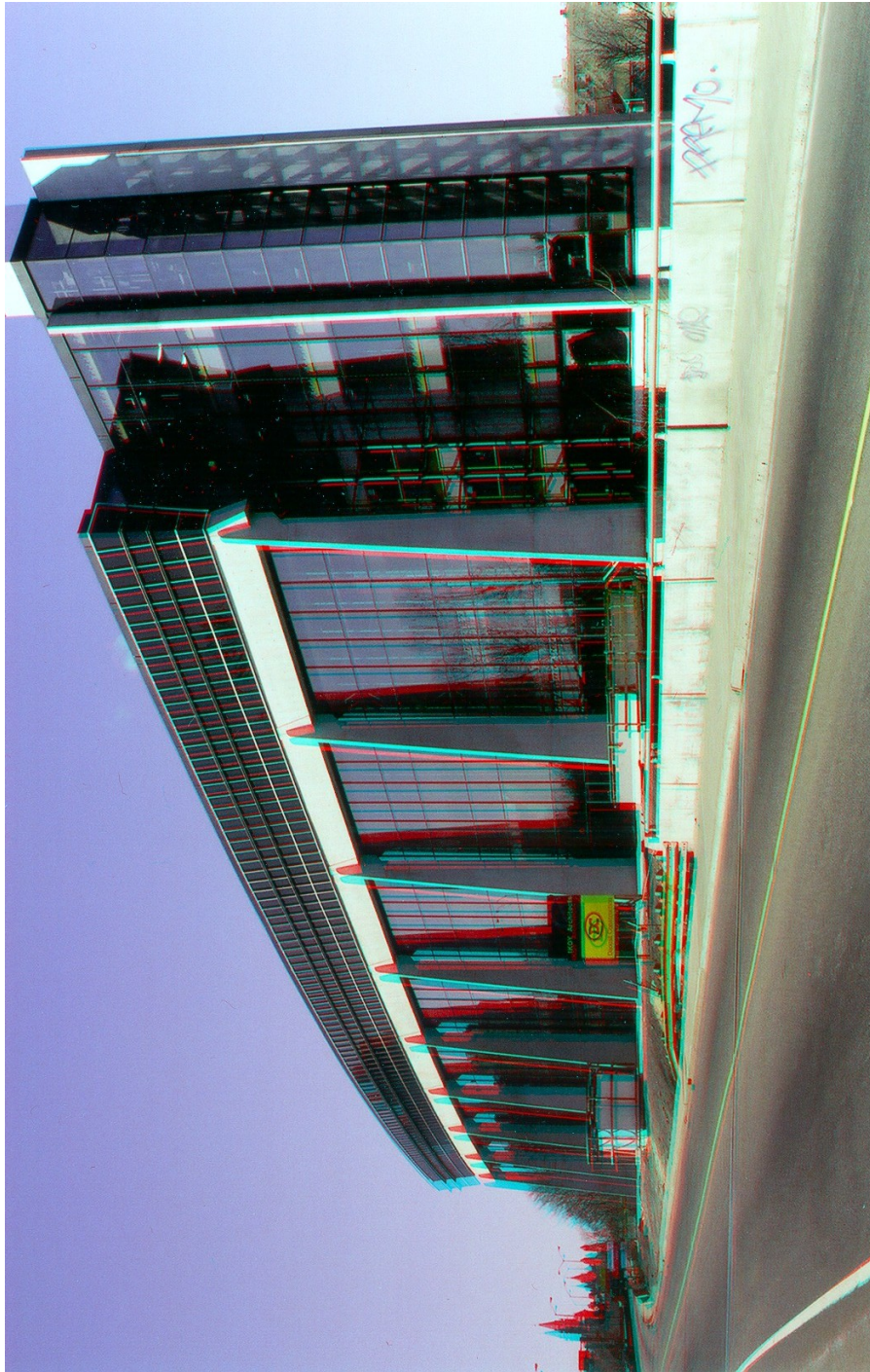


Figure 5.9: The anaglyph image with  $d=1$



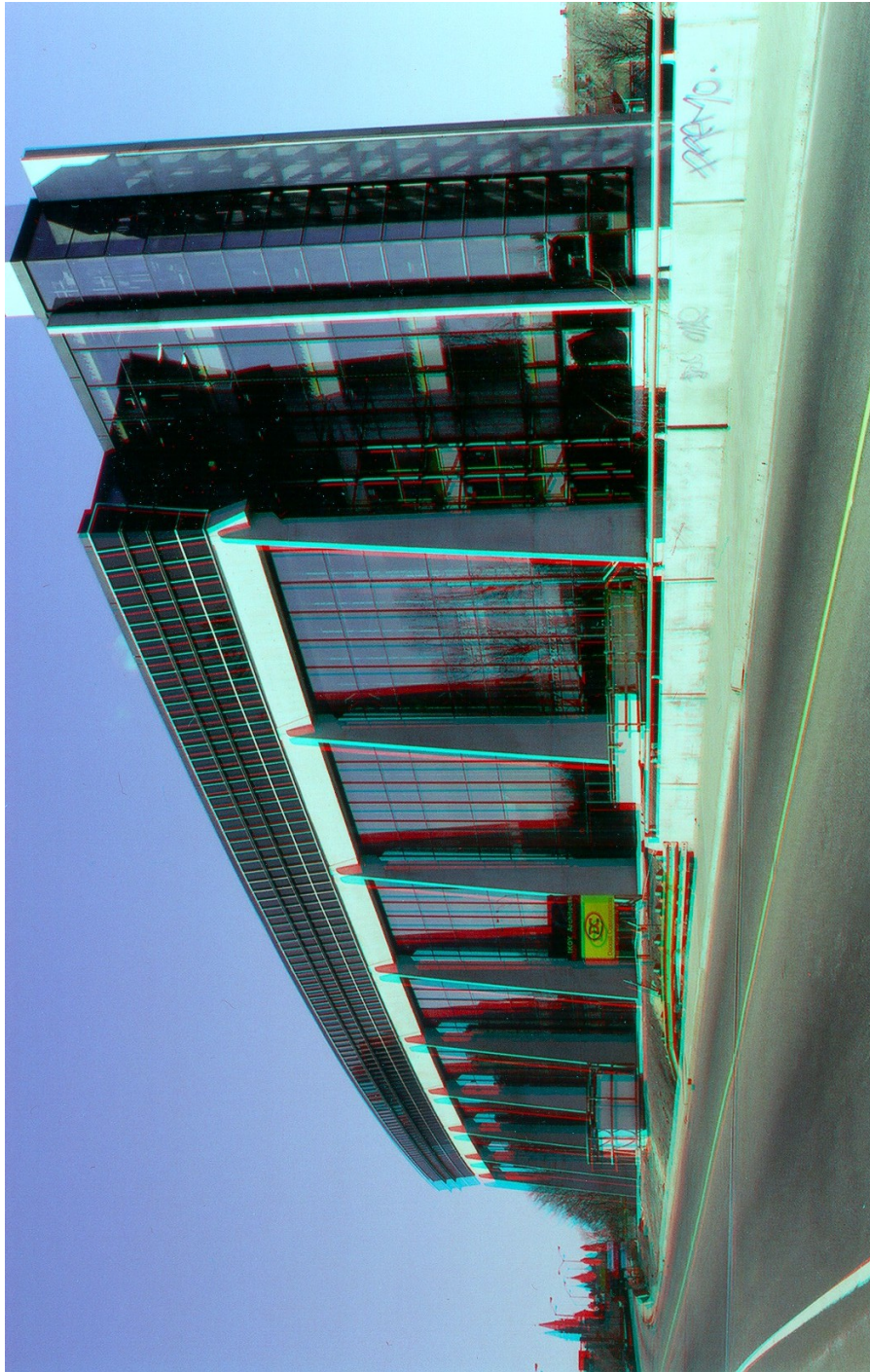


Figure 5.10: The anaglyph image with  $d=0.88$



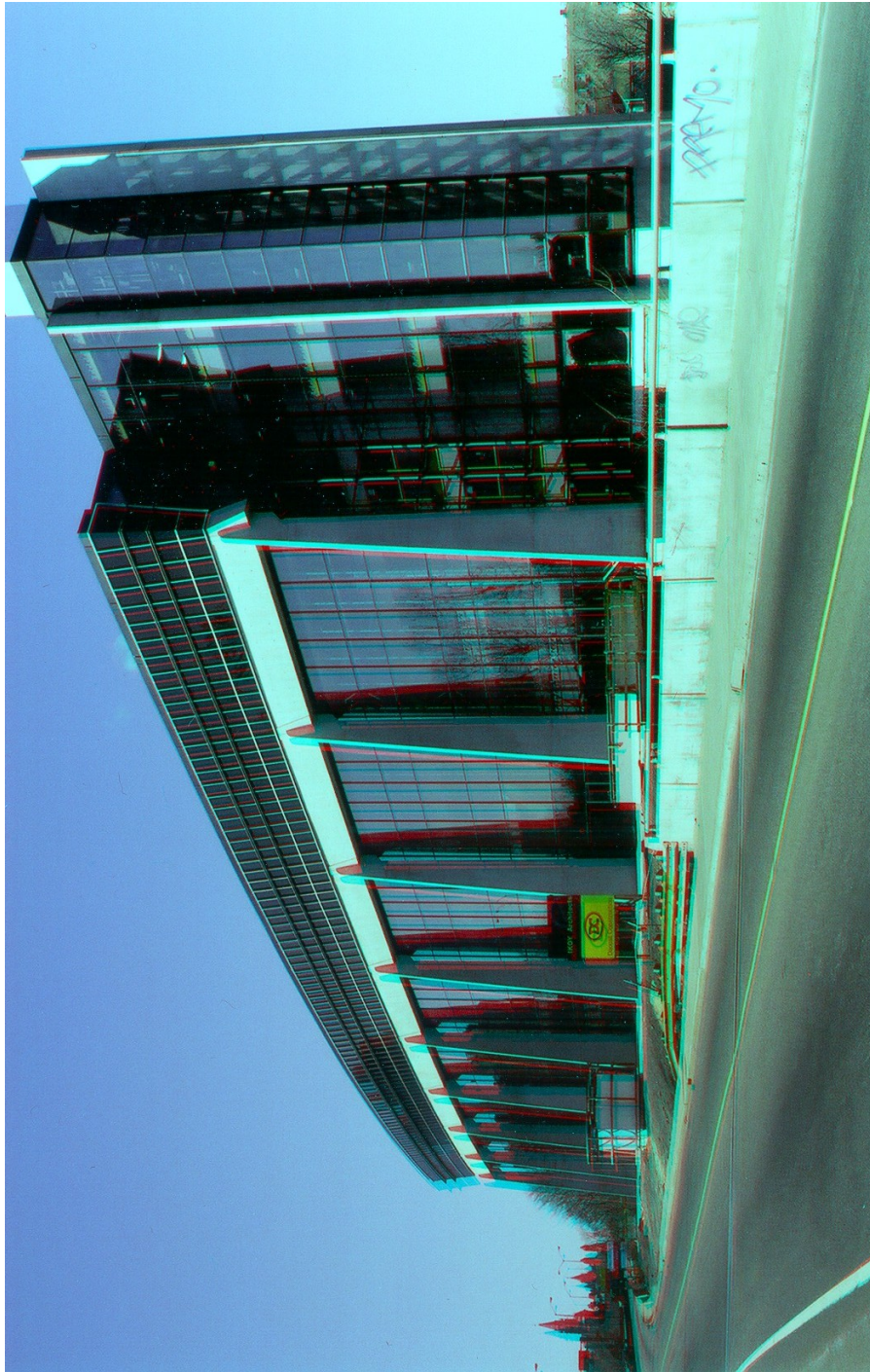


Figure 5.11: The anaglyph image with  $d=0.76$



by other stereo image viewing techniques. In the following discussion, we will focus on our Image-Based rendering approach using the Concentric Mosaics technique.

### 5.3.1 The distance changing between left light rays and right light rays

Fig. 5.12 illustrates the procedure for rendering a stereo view. Assume that a viewer with two eyes  $E_L$  and  $E_R$  is located at an arbitrary position P and that the distance between two eyes is  $d$ .

Let us first discuss the normal viewing direction, or the viewing direction that is perpendicular to the baseline  $E_L E_R$  of the two eyes.  $E_L L$  and  $E_R R$  are the normal viewing directions from  $E_L$  and  $E_R$  respectively. Thus the offset between the two images that will be selected to render the stereo view in the normal direction is  $AB$  as shown in the figure.

Assume that the angle between AD and AB is  $\theta$ , which is also the angle between  $CN$  and  $PM$  because  $PM$  is perpendicular with  $AD$  and  $CN$  is perpendicular with  $AB$ . In the right triangle  $ABD$ ,

$$AB = \frac{AD}{\cos \theta} \quad (5.51)$$

where  $AD = d$ . From the definition of navigation area, we know

$$\theta \leq \frac{\text{HFOV}}{2} \quad (5.52)$$

Thus, assuming that the camera's horizontal field of view (HFOV) is  $43^\circ$ , which is the common value for cameras, we obtain

$$d \leq \| AB \| \leq 1.07d. \quad (5.53)$$

Beside the normal viewing direction, we assume an arbitrary viewing direction  $E_L L_1$  and  $E_R R_1$  as shown in Fig. 5.12, which forms an angle  $\gamma$  with the normal direction. The exact formula to calculate the offset  $\| A_1 B_1 \|$  is complex because it depends on the position P. Thus we make an assumption that the change of  $\| A_1 B_1 \|$

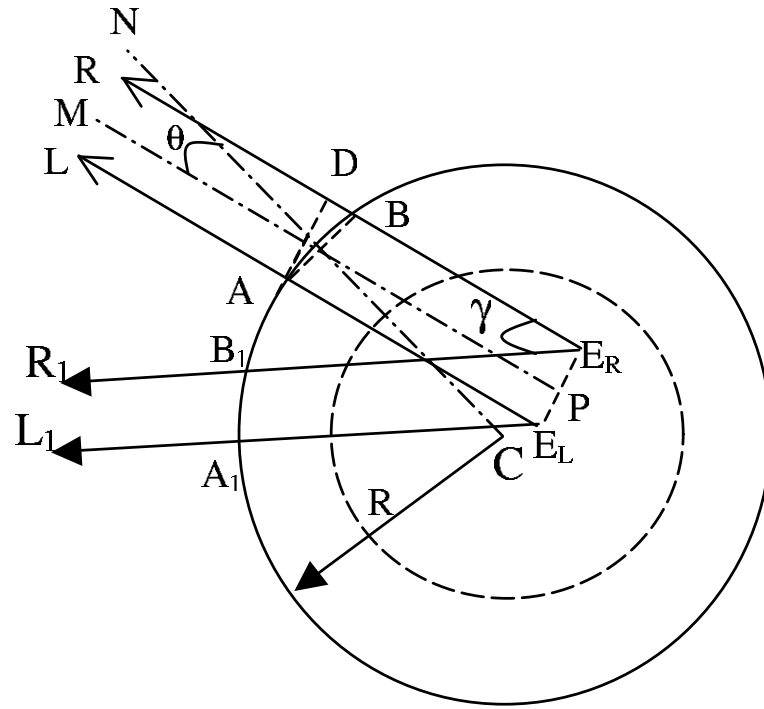


Figure 5.12: The analysis on the viewing distance between two eyes (The distance between two eye  $E_R$  and  $E_L$  is  $d$ .)

is not very large so that we can use the fixed offset at normal viewing direction to approximate it. We will test our assumption by the simulation results.

From the above analysis, we find that the change of offset between two images that will be used in the synthesis of stereo views is not very large. A new algorithm for the fast rendering of stereo views is proposed based on this fact through pre-processing of the pre-captured images.

The normal procedure to synthesize an anaglyph view is time-consuming. The left and right views are rendered separately first, and then an anaglyph image is generated.

The proposed algorithm pre-processes the pre-captured image database from the ordinary images to the anaglyph-type images first, and then the rendering of arbitrary views is based on the anaglyph-type image database. In this way, we transfer the

processing time of creating anaglyph images to the pre-processing procedure and only one image is required to be rendered for stereo viewing.

The procedure for creating an anaglyph-type image database is simple. Assume that the pre-captured image database  $D_I$  is formed by a set of consecutive images as,

$$D_I = \left( I_0 \ I_1 \ I_2 \ I_3, \dots, \ I_{N-1} \right) \quad (5.54)$$

with  $N$  images in total.

An anaglyph image  $I_k^{\text{an}}$  in a new anaglyph-type image database  $D_I^{\text{an}}$  is created through two images  $I_{k-M}$  and  $I_{k+M}$  in  $D_I$ , with a fixed offset  $2M + 1$  (in the unit of number of images).  $M$  is determined by the capture conditions and the distance between two eyes. Note that we use

$$k + M - N \text{ instead of } K + M \quad (5.55)$$

in the case  $k + M > N - 1$ , and

$$N + M - k \text{ instead of } K - M \quad (5.56)$$

in the case  $k - M < 0$ .

In this way, the anaglyph-type image database is formed as

$$D_I^{\text{an}} = \left( I_0^{\text{an}} \ I_1^{\text{an}} \ I_2^{\text{an}} \ I_3^{\text{an}}, \dots, \ I_{N-1}^{\text{an}} \right) \quad (5.57)$$

### 5.3.2 Simulation results and conclusions

The simulations were performed using a Concentric Mosaics data set provided by Microsoft Research, Beijing.

We created anaglyph views with the proposed method based on the anaglyph-type image database and compared the stereo quality with our liquid crystal glasses-based viewing system.

Fig. 5.13 shows an anaglyph view from our proposed method. Our subjective conclusions are that the stereo effect of the anaglyph image is quite acceptable compared



Figure 5.13: A rendered anaglyph image with the proposed fast algorithm



Figure 5.14: A rendered anaglyph image with the usual approach

with that from using the shuttered glasses method. For comparison, an anaglyph view with the usual procedure, by first rendering left and right views and then generating the anaglyph view, is shown in Fig. 5.14. We can see the stereo quality is almost the same. Rendering stereo views is just like rendering monoscopic views by the algorithm proposed, which is both fast and convenient. Moreover, the anaglyph technique makes the stereoscopic viewing possible for any ordinary observers. Combining the Image-Based Rendering techniques with anaglyph technique, people distributed at different locations all over the world will be able to enjoy navigating in a same stereoscopic real image-based virtual environment over the Internet.

# Chapter 6

## Conclusions and Future Work

### 6.1 Summary of the Thesis

In this thesis, Image-Based Rendering techniques have been studied for the purpose of constructing a real-image-based virtual reality. Some currently developed methods, including panoramas, view interpolation, the Light Field Rendering technique and the Concentric Mosaics technique, are discussed.

The general mathematical model for the scene representation through plenoptic function is studied in Chapter 2. The techniques for Image-Based Rendering (for both monoscopic and stereoscopic views) can be categorized into this model.

Methods to evaluate various Image-Based Rendering techniques are required and I know of no such method until now. However, the evaluation standard for an Image-Based Rendering technique should include the following factors:

- The fidelity of the rendered image with respect to the scene or the object to be represented;
- The quality (resolution) of the rendered image;
- The complexity of the technique in both capture and rendering procedures;
- The quantity of the pre-captured images.

From our studies in this thesis, we find that compromise must be reached among these factors.

Using view interpolation, fewer images are required to be captured, compared with the other two methods. However, the method for view interpolation is complex and the intermediate views are only approximately interpolated. This has been demonstrated in Chapter 3.

The Light Field Rendering technique can theoretically achieve the highest fidelity between rendered images and the scene or object to be represented. However, the capture devices are complex and the motion control errors in the capture procedure will transfer into the rendered image. Thus the quality of the rendered images is poor as we illustrated in Chapter 4. Furthermore, the image quantity is huge even for a small object.

Depth distortion is unavoidable in the Concentric Mosaics technique, causing differences between the rendered images and the scene that is represented. The differences depend on the depth variations of the scene and usually it is not a very serious problem. Compared with the Light Field Rendering technique, the capture devices are not complex for the practical application. The motion control errors will also transfer into the rendered images, but the rendering errors only reside between adjacent columns instead of all adjacent pixels as they do in the Light Field Rendering technique.

The advantages of the stereoscopic view over the monoscopic one are obvious, especially for the application of navigating in a virtual environment. The anaglyph technique makes viewing stereo images on a monitor possible for a personal computer user using a pair of colored glasses with negligible price. Thus, in Chapter 5, the combination of Image-Based Rendering techniques with anaglyph technique provides an opportunity for an ordinary personal computer user to navigate in a real image-based stereo virtual environment, which could be museum, shopping mall or any other places, over Internet.

## 6.2 Thesis Contributions

This thesis has made contributions in three main areas, each of which has resulted in a conference paper.

A simple and efficient algorithm for image rectification based on epipolar geometry was applied in view interpolation. From the simulation results of image rectification, we found that the warping distortion does not severely degrade the rectified images. Previous work for rectification is either based on the assumption of orthogonal projection, which is not a good approximation in practice, or based on the spatial transformation of the camera pose, which makes the algorithm very complex and sometimes badly distorts the rectified images. The more severe the rectification distortion, the worse the quality of the synthesized in-between views. This work was published in [15].

Based on the scene sampling theory, some design issues in the Concentric Mosaics technique were studied in this thesis. These issues include how to determine the length of the rotation beam and the rotation velocity. There is no theoretical criterion found in the previous work to build up the devices for pre-capturing the image data base. This work will appear in [19].

The anaglyph technique is introduced to combine with Image-Based Rendering techniques as one kind of Image-Based Stereoscopic View Rendering technique. In particular, a fast anaglyph-based stereoscopic view rendering algorithm is proposed for the Concentric Mosaics technique by pre-processing the pre-captured images and it has been verified by the simulation results. This was presented in [16].

## 6.3 Future Work

Image-Based Rendering is a relatively new topic in image processing, generated from the practical requirement. There is still much work to do in the future, including some fundamental studies.

First of all, how many pre-captured images are minimally required to represent an



environment no matter what specific technique is used? It certainly depends on the environment to be represented, such as the variations of depth, texture, light field, etc.

Second, as we have mentioned at the beginning of this Chapter, how can we evaluate and compare one Image-Based Rendering technique with other methods? How can we define the fidelity of the rendered image with respect to the scene or the object to be represented, as we know usually no such standard image exists for every view.

In addition, the final goal of the image-based virtual environment application is oriented toward the ordinary personal computer user. Thus the compression for rendering, which means both compression and rendering with the compressed image data, is an important research topic. Using the Concentric Mosaics technique as an example, the pre-captured images are loaded into the computer's RAM at the beginning of rendering in our system for initial studies. This requires a large RAM and even so it is impractical for a large environment. The studies on the rendering based on compressed pre-captured images and the rendering speed, which is important for real-time application, were not carried out in this thesis. Some work has been initiated in the compression for rendering in the literature, such as MPEG-2 based RBC (Reference Block Codec) [36] [37], data rebining [38], 3D wavelet [39] [40]. An evaluation software package using RBC can be downloaded from Microsoft Research's public web site [41]. However, much work on compression and real-time rendering is required in the future in order to navigate in a large, even dynamic environment through limited network transmission rate. And the image quality should as least be comparable with the current TV standard.

# Bibliography

- [1] <http://www.graphics.stanford.edu/papers/light>, (visited July, 2002).
- [2] R. Szeliski and S. M. Seitz, “Lecture notes (winter, 2001): Vision for graphics (University of Washington) (<http://www.cs.washington.edu/education/courses/cse590ss/01wi/>),” (visited July, 2002).
- [3] S. E. Chen, “Quicktime VR — an image-based approach to virtual environment navigation,” *Computer Graphics (SIGGRAPH’95)*, pp. 29–38, August 1995.
- [4] H.-Y. Shum and S. Kang, “A review of image-based rendering techniques,” *Proc. SPIE, Visual Communications and Image Processing 2000*, pp. 2–13, June 2000.
- [5] P. Debevec, Y. Yu, and G. Borshukov, “Efficient view-dependent image-based rendering with projective texture-mapping,” *Proc. 9th Eurographics Workshop on Rendering*, pp. 105–116, 1998.
- [6] W. Mark, L. McMillan, and G. Bishop, “Post-rendering 3D warping,” *Proc. Symposium on 3D Graphics*, pp. 7–16, 1997.
- [7] J. Shade, S. Gortler, L.-W. He, and R. Szeliski, “Layer depth images,” *Computer Graphics (SIGGRAPH’98)*, pp. 231–242, July 1998.
- [8] C. Chang, G. Bishop, and A. Lastra, “LDI tree,” *Computer Graphics (SIGGRAPH’99)*, pp. 291–298, August 1999.

- [9] S. Avidan and A. Shashua, “Novel view synthesis in tensor space,” *Conference on Computer Vision and Pattern Recognition*, pp. 1034–1040, 1997.
- [10] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [11] S. E. Chen and L. William, “View interpolation for image synthesis,” *Computer Graphics (SIGGRAPH’93)*, pp. 279–288, 1993.
- [12] S. M. Seitz and C. R. Dyer, “View morphing,” *Computer Graphics (SIGGRAPH’96)*, pp. 21–30, 1996.
- [13] M. Levoy and P. Hanrahan, “Light field rendering,” *Computer Graphics (SIGGRAPH’96)*, pp. 31–42, August 1996.
- [14] H.-Y. Shum and L. He, “Rendering with concentric mosaics,” *Computer Graphics (SIGGRAPH’99)*, pp. 299–306, January 1999.
- [15] X. Sun and E. Dubois, “A method for the synthesis of intermediate views in image-based rendering using image rectification,” *Proceedings of the 2002 IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 1, pp. 991–994, May 2002.
- [16] X. Sun and E. Dubois, “Image-based stereo rendering using the anaglyph technique,” *Proceedings of the 21st Biennial Symposium on Communications*, pp. 506–510, June 2002.
- [17] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum, “Plenoptic sampling,” *Computer Graphics (SIGGRAPH’2000)*, pp. 307–318, July 2000.
- [18] M. Wu, H. Sun, and H.-Y. Shum, “Real-time stereo rendering of concentric mosaics with linear interpolation,” *Proc. SPIE, Visual Communications and Image Processing 2000*, vol. 4067, pp. 23–30, 2000.

- [19] X. Sun and E. Dubois, "Scene sampling for the concentric mosaics technique," *Accepted by International Conference on Image Processing 2002*, September 2002.
- [20] L. McMillan and G. Bishop, "Plenoptic modelling: An image-based rendering system," *Computer Graphics (SIGGRAPH'95)*, pp. 39–46, August 1995.
- [21] E. Dubois, "A projection method to generate anaglyph stereo images," *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 3, pp. 1661–1664, May 2001.
- [22] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," *Computer Graphics (SIGGRAPH'96)*, pp. 43–54, August 1996.
- [23] H.-Y. Shum and R. Szeliski, "Panoramic image mosaics," *Technical Report, Microsoft Research, MSR-TR-97-23*.
- [24] S. M. Seitz and C. R. Dyer, "Physically-valid view synthesis by image interpolation," *Proc. Workshop on Representations of Visual Scenes*, pp. 18–25, 1995.
- [25] R. Szeliski and J. Coughlan, "Hierarchical spline-based image registration," *International Journal of Computer Vision*, vol. 22, no. 3, pp. 199–218, 1997.
- [26] T. Beier and S. Neely, "Feature-based image metamorphosis," *Computer Graphics (SIGGRAPH'92)*, pp. 35–42, 1992.
- [27] P. Heckbert, "Survey of texture mapping," *IEEE Computer Graphics and Applications*, vol. 6, no. 11, pp. 56–67, 1986.
- [28] B. Lucas and T. Kanade, "An iterative image registration technique with an application in stereo vision," *Seventh International Joint Conference on Artificial Intelligence(IJCAI-81)*, pp. 674–679, 1981.
- [29] G. Roth, <http://www2.vit.iit.nrc.ca/~gerhard/PVT>, (visited July, 2002).

- [30] R. Hartley and R. Gupta, "Computing matched-epipolar projections," *Proceedings of the IEEE conference on computer Vision and Pattern Recognition*, pp. 549–555, 1993.
- [31] G. Roth and A. Whitehead, "Using projective vision to find camera positions in an image sequence," *Proceedings of Vision Interface 2000*, pp. 225–232, May 2000.
- [32] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 7, no. 2, pp. 139–154, 1985.
- [33] A. Mertens, *Signal Analysis. Wavelets, Filter Banks, Time-Frequency Transforms and Applications*. Chichester, UK: John Wiley & Sons, 1999.
- [34] G. Wyszecki and W. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulas*. New York, NY: John Wiley & Sons, 1967.
- [35] C. Poynton, *A Technical Introduction to Digital Video*. New York, NY: John Wiley & Sons, 1996.
- [36] C. Zhang and J. Li, "Compression and rendering of concentric mosaics with reference block codec (RBC)," *Proc. SPIE: Visual Communications and Image Processing*, vol. 4067, pp. 43–54, 2000.
- [37] C. Zhang and J. Li, "Interactive browsing of 3D environment over the internet," *Proc. SPIE: Visual Communications and Image Processing*, vol. 4310, pp. 509–520, 2001.
- [38] Y. Wu, C. Zhang, J. Li, and J. Xu, "Smart-rebinning for compression of concentric mosaic," *Proceedings of ACM Multimedia 2000*, 2000.
- [39] Y. Wu, L. Luo, J. Li, and Y.-Q. Zhang, "Rendering of 3D wavelet compressed concentric mosaic scenery with progressive inverse wavelet synthesis (PIWS),"

*Proc. SPIE, Visual Communications and Image Processing*, vol. 4067, pp. 31–42, 2000.

- [40] L. Luo, Y. Wu, J. Li, and Y.-Q. Zhang, “Compression of concentric mosaic scenery with alignment and 3D wavelet transform,” *Proc. SPIE: Image and Video Communication and Processing*, vol. 3974, pp. 89–100, January 2000.

- [41] <http://research.microsoft.com/downloads>, (visited July, 2002).