

HIGH FIDELITY AND VERSATILE VISUALIZATION OF 3D VIDEO

Takeshi Takai

HIGH FIDELITY AND VERSATILE VISUALIZATION OF 3D VIDEO

Takeshi Takai

A B S T R A C T

This thesis presents visualization of 3D video that is a real movie recording dynamic visual events in the real world as they are: time-varying 3D object shapes with a high-fidelity surface texture. We consider two different types of 3D video visualization, we call one of them “high fidelity visualization” and “versatile visualization.”

The goal of the high fidelity visualization is to generate free-viewpoint object images that reflect input multi-view images with high fidelity. For the high fidelity visualization, we propose the viewpoint dependent vertex-based texture mapping method that enables us to visualize a 3D video object even though it has the shape of limited accuracy.

The goal of the versatile visualization is to generate free-viewpoint object images with versatile effects, i.e., geometric directing by utilizing a shape and/or motion, and photometric directing by estimating reflectance parameters and lighting environment. For the geometric directing, we semi-automatically generate camerawork —composition of the scene and parameters of a virtual camera from which we observe an object— by utilizing the shape of an object. For the photometric directing, we propose two types of methods for lighting environment estimation of real scenes utilizing reference objects, i.e., the skeleton cube and the difference sphere. The skeleton cube has a capability to estimate complex lighting environment, and the difference sphere can estimate the precise parameters of lighting environment. We describe estimation methods using the skeleton cube and the difference sphere independently, and moreover discuss the combination of the two methods. With the estimated lighting environments, we estimate the reflectance parameters of an object, and we can visualize 3D video with arbitrary lighting effects, i.e., the photometric directing.

In this thesis, we first present an overview of free-viewpoint image generation, and introduce 3D video that we have been developed. We then describe the high fidelity visualization of 3D video. Before describing the versatile visualization of 3D video, we present the lighting environment estimation methods, i.e., the skeleton cube and the difference sphere for the photometric directing. We finally present the versatile visualization and demonstrate its effectiveness with visualized 3D video images.

ACKNOWLEDGEMENTS

I first wish to express my gratitude to my supervisor, Prof. Takashi Matsuyama. He gave me the opportunity to study at Kyoto University, and guided me diligently to complete my thesis.

I also wish to express my gratitude to my thesis committee, Prof. Michihiko Minoh and Prof. Koji Koyamada, for their useful comments and suggestions on my thesis.

I would like to thank Associate Prof. Atsuto Maki for making a lot of comments and supporting me to accomplish my thesis. In addition, he read my manuscripts many times and gave me very useful comments both technically and grammatically. I would also like to thank Prof. Kazuhiko Sumi, Assistant Prof. Hitoshi Habe, and Assistant Prof. Hiroaki Kawashima. Discussions with them were truly advantageous to me. Also, they were good fellows to drink with. I would like to thank Ms. Yuki Onishi, Ms. Tomoko Hayama, and Ms. Satomi Taku for their assistance of office jobs. Mr. Xiaojun Wu and Mr. Shohei Nobuhara were my comrades on writing the doctoral thesis. We supported each other to survive from a stormy sea of studies. The members of Matsuyama laboratory were good ones. In particular, the people of the golden age, I am very proud of a member of them. I will never forget you!

I would like to thank all the people who were concerned me during my student life, although I do not mention the names.

I finally wish to express my deepest thanks to my family who has always supported me both emotionally and economically, and my special thanks go to my wife Shiho and my daughter Chihiro.

Contents

Abstract	i
Acknowledgements	iii
1 Introduction	1
1.1 Free-viewpoint Image Generation from Multi-view Images	1
1.1.1 Categorization of Free-viewpoint Image Generation Methods . .	2
1.1.2 3D Video	5
1.2 3D Video Visualization	7
1.2.1 High Fidelity Visualization	8
1.2.2 Versatile Visualization	9
1.3 Lighting Environment Estimation	13
1.3.1 Related Work	13
1.3.2 Our Methodology: Skeleton Cube and Difference Sphere	15
1.4 Overview of Thesis	16
2 High Fidelity Visualization of 3D video	19
2.1 Introduction	19
2.2 3D Video Capturing System Organization	20
2.3 Viewpoint Independent Patch-Based Method	20
2.4 Viewpoint Dependent Vertex-Based Method	22
2.4.1 Definitions	22
2.4.2 Visible Vertex from Camera c_j	24
2.4.3 Algorithm	25
2.5 Performance Evaluation	26
2.6 Discussions	31

3	Skeleton Cube for Lighting Environment Estimation	35
3.1	Introduction	35
3.2	Lighting Environment Model	36
3.2.1	Photometric Model	37
3.2.2	Types of Light Sources	37
3.2.3	Arrangement of Light Sources	39
3.3	Skeleton Cube	42
3.3.1	Geometric Features	42
3.3.2	Photometric Features	43
3.4	Lighting Environment Estimation	45
3.4.1	Sensing System	45
3.4.2	Computational Algorithm	47
3.5	Verifying Utilities of Skeleton Cube	47
3.5.1	Evaluating Effectiveness of Lighting Environment Estimation with Self-shadows	47
3.5.2	Evaluating Capability of Self-Shadow Generation of Skeleton Cube	54
3.6	Performance Evaluation	54
3.6.1	Computer Simulation	54
3.6.2	Real Scene	57
3.7	Discussions	65
4	Difference Sphere: An approach to Near Light Source Estimation	67
4.1	Introduction	67
4.2	Model Definitions	68
4.2.1	Assumptions	68
4.2.2	Coordinate Systems	68
4.2.3	Lighting Environment	68
4.2.4	Radiance	69
4.3	Characteristics of Single and Difference Sphere	70
4.3.1	Radiance	70
4.3.2	Surface Analysis	73
4.4	Algorithm of Light Source Estimation	79
4.4.1	Region Representation with Gray-Level Contour Lines	80

4.4.2	Estimation of Light Source Candidates' Parameters	81
4.4.3	Verification of Estimated Light Source Candidates	81
4.5	Performance Evaluation	84
4.5.1	CG Simulation	84
4.5.2	Real Scene	86
4.6	Discussions	90
4.6.1	Range of Identifying a Near Light Source	90
4.6.2	Summary	92
5	Versatile Visualization of 3D Video	93
5.1	Introduction	93
5.2	Geometric Directing: Camerawork Creation using Object's Shape and Motion	94
5.2.1	Overview	94
5.2.2	Algorithm of Camerawork Creation	95
5.2.3	GUI Application for Geometric Directing	99
5.2.4	Performance Evaluation	99
5.3	Photometric Directing: Lighting Effect Control	104
5.3.1	Overview	104
5.3.2	Estimation of Lighting Environment	104
5.3.3	Estimation of Reflectance Parameters	105
5.3.4	Lighting Effect Control	107
5.3.5	Performance Evaluation	107
5.4	3D Video Viewer	110
5.4.1	Specification of 3D Video Viewer	111
5.4.2	Format of 3D Video Object	112
5.5	Discussions	113
6	Concluding Remarks	117
6.1	Summary	117
6.2	Future Work	118
6.2.1	High Fidelity Visualization	118
6.2.2	Versatile Visualization	118

List of Figures

1.1	Generation process of the shape of an object	7
1.2	Procedure of 3D Video Visualization	12
2.1	PC cluster for real-time active 3D object behavior reconstruction system.	21
2.2	Viewpoint independent patch-based method	22
2.3	Viewpoint and camera position	23
2.4	Depth buffer	24
2.5	Relations between patches	25
2.6	Images generated by the Viewpoint Independent Patch-Based Method .	27
2.7	Images generated by the Viewpoint Dependent Vertex-Based Method . .	27
2.8	Sample images of generated 3D video with D-mesh	28
2.9	Camera Setting	29
2.10	Root-mean-square error of RGB value (1) (Mesh)	30
2.11	Root-mean-square error of RGB value (1) (D-mesh)	31
2.12	Subdivision of a surface patch	31
2.13	Mean side length in pixel on image planes of cameras	32
2.14	Root-mean-square errors of RGB value (2)	32
2.15	Subtraction between a real image and a generated image (frame #106) .	33
2.16	Example images visualized with original and subdivided patches (frame #103)	34
2.17	Visualized 3D video with subdivided patches (frame#103)	34
3.1	The Skeleton Cube.	36
3.2	A point light source and an illuminated surface	39
3.3	An area light source and illuminated surface	40

x List of Figures

3.4	An example of the background lighting and the foreground lighting . . .	41
3.5	The geometry of reflection	44
3.6	Appearance of an area light source viewing from a point on an inner surface of the skeleton cube	45
3.7	The geometry of the skeleton cube	46
3.8	Distribution of light sources	49
3.9	Visualized matrix K (Point light source)	51
3.10	Visualized matrix K (Area light source)	52
3.11	Contributions of singular values of matrix K (Point light source)	53
3.12	Contributions of singular values of matrix K (Area light source)	53
3.13	Self-occlusion of the skeleton cube	55
3.14	Relationship between camera and light source	58
3.15	Estimation of the position of a light source	60
3.16	Sampling points for reflectance parameter estimation	61
3.17	Lighting condition	62
3.18	Arrangement of skeleton cube and light sources	63
3.19	Images of the skeleton cube and a digital camera in a studio	64
3.20	Images for lighting environment estimation captured by Nikon D70	64
3.21	Skeleton cube illuminated by estimated lighting environment	65
4.1	Coordinate systems.	69
4.2	Point light source	70
4.3	Directional light source	71
4.4	Single spheres and difference sphere	73
4.5	Surface classification	74
4.6	Relationship between a single directional light source and a reference sphere	76
4.7	Relationship between a single point light source and a reference sphere.	79
4.8	An example of an estimation flow	83
4.9	Procedures of light source estimation.	85
4.10	Procedure of light source estimation (Real scene)	88
4.11	The experiment of adding virtual object in a real scene	89
4.12	Reference sphere illuminated by a point light source	90

4.13 Relationship between l_{x_p, x_d} and d_p ($r = 1$)	91
5.1 Virtual Scene Setup	96
5.2 Relationship between a virtual camera and an object (I)	97
5.3 Relationship between object and scene coordinate systems	98
5.4 Relationship between a virtual camera and an object (II)	98
5.5 GUI application for 3D video editing and visualization	99
5.6 3D video campturing studio 1	100
5.7 Images rendered by specifying a composition of a scene	101
5.8 3D video sequence directed using object's shape and motion	103
5.9 Combination of the skeleton cube and the difference sphere	105
5.10 3D video capturing studio 2	108
5.11 Captured images for reconstruction of an object	109
5.12 Sample images of a reconstructed object	110
5.13 Sample images of an object with median color	111
5.14 Sample images of object colored by estimated reflectance parameters	112
5.15 Sample images of object with new lighting environment 1	113
5.16 Sample images of object with new lighting environment 2	114
5.17 Rendered image for Sanyo display	115
5.18 Rendered image for Sharp display	115
5.19 Format of texture file	116
5.20 An example of texture file	116

List of Tables

1.1	Categorization of free-viewpoint image generation	3
3.1	Settings of lighting environment	56
3.2	Root mean square errors of estimated radiant intensities at each quantizational level (Max intensity: 10000)	56
3.3	Ratio of root mean square errors between calculated radiance and radiance under estimated lighting environment at each quantizational level (compared with max value of calculated radiance)	57
3.4	Specifications of Nikon D70 [Nik04]	57
3.5	Estimated reflectance parameters	61
3.6	Root-mean-square errors of radiance between the pixel values of captured image and the values calculated with the estimated parameters	62
4.1	Estimated parameters of lighting environment in CG image	86
4.2	Estimated parameters of lighting environment in real scene	87
5.1	Specifications of Sony EVI G20	100
5.2	Object's motion	101
5.3	Specified compositions	101
5.4	Specifications of Sony DFW-VL500 [Son01]	107
5.5	Parameters of a new point light source	110

Chapter 1

Introduction

1.1 Free-viewpoint Image Generation from Multi-view Images

Free-viewpoint image generation from multi-view images is one of the most interesting topics in computer vision, which aims to reconstruct 3D scenes from 2D images. The free-viewpoint image enables us to observe a captured scene from an arbitrary viewpoint, and its applications cover wide varieties of personal and social human activities. Those include entertainment, education, sports, medicine, and culture. Moreover, the free-viewpoint image can also be used for 3D display systems because it can generate stereoscopic images by selecting an appropriate pair of viewpoints. Since several manufactures are recently producing display devices that can represent 3D images from stereoscopic images at a reasonable price, the free-viewpoint image will become popular to our ordinary lives soon.

One of the advantage of the free-viewpoint image is that it is a visual media that has three dimensional information. The visual media is an important means of communication such as often represented by “Seeing is Believing.” The visual media has been developed from a drawing picture that reflects a painter’s subject and technique to a photograph that represents a real scene in which some events happened. After these media, a movie has been developed as a sequence of photographs at each time, and it has been drastically changing our lives. A movie also gives an extraordinary effect that cannot be realized by photographs, i.e., an effect to control the time. It enables us to observe an action of human by reversing the time, an emergence of a butterfly by shrinking the time, and so on. On the other hand, there is a hard problem to control another dimension—an arbitrary viewpoint control in the 3D space, because it is impossible to arrange cameras at all positions in the 3D space

2 Chapter 1. Introduction

in order to capture an object. Hence, the free-viewpoint image generation using multi-view images has been studied vigorously. The free-viewpoint image enables us not only to enjoy observing an object, but also to support analysis, tracking, recognition, etc., which is difficult to solve with the information of a single viewpoint.

Another advantage of the free-viewpoint image is that it can be used for 3D display systems. That is, it enables us to obtain images captured by a stereo pair in arbitrary positions. The stereo movies that we can see at present in theaters or amusement parks are the movies that are captured by a stereo camera or generated by the computer graphics. Therefore, we cannot see movies of real scenes by changing a viewpoint dynamically. The free-viewpoint image, however, enables us to see an interactive stereo-movie of real scenes, and it is an outstanding achievement since Wheatstone developed the equipment for seeing a stereograph using two parallax pictures, i.e., the stereoscope, in 1838.

As we described the above, the free-viewpoint image has a lot of potential, and it should accomplish novel visual expressions or applications should be appeared. In the following section, we categorize methods for free-viewpoint image generation from multi-view images, before describing *3D video* that is being developed in our laboratory.

1.1.1 Categorization of Free-viewpoint Image Generation Methods

We categorize methods for free-viewpoint image generation from multi-view images as follows:

- ◆ **Shape of object:**

- ❖ *Generic property reconstruction*

- A method in which 3D shape of an object is reconstructed

- ❖ *Appearance reconstruction*

- A method in which 3D shape of an object is not reconstructed, but the object is represented by combining multiple images appropriately.

- ◆ **Color of object:**

- ❖ *Generic property reconstruction*

- A method in which reflectance property is recovered

❖ *Appearance reconstruction*

A method in which reflectance property is not recovered, but color under the lighting environment in which the object is captured is represented.

Summarizing the above, we have the following table (Table 1.1):

Table 1.1: Categorization of free-viewpoint image generation

Shape	appearance	appearance	generic property	generic property
Color	appearance	generic property	appearance	generic property
Type	I	II	III	IV

Type I (Shape: Appearance, Color: Appearance): It is a method for generating free-viewpoint images without reconstructing a 3D shape of an object explicitly. This type has advantages of visualizing photorealistic images without modeling a 3D shape and texture of an object.

A typical example of Type I is *image based rendering* (IBR) method, in which free-viewpoint images are generated by switching or synthesizing multi-view images. Debevec et al. presented the *bullet time* shots in the film—The Matrix [Deb99]. The shots are captured by a large number of cameras that are precisely calibrated, and show amazing scenes that cannot be realized by an ordinary camera. Kanade et al. presented the *Eyevision* [Kan01] which controls pan and tilt parameters of 32 cameras, and show an image sequence by simply selecting an appropriate camera in each image frame. Though the bullet time shots and the Eyevision are not a full free-viewpoint image, they show an impact of changing viewpoints dynamically. Tanimoto et al. propose the free viewpoint television [FT04], which synthesizes an image from arbitrary viewpoint by a ray-space method. It can generate natural free-viewpoint images in real time with a large number of cameras and a PC cluster system. Mukaigawa et al. propose to reconstruct 3D shape of an object as an intermediate representation though the final result does not contain the shape [MNO98]. Also, Kitahara et al. propose to represent an object by a set of 2D slice images as the intermediate representation [KSA⁺01].

As described above, the IBR method enables us to reconstruct an appearance of captured scenes by switching or synthesizing multi-view images. However, it is hard to

4 Chapter 1. Introduction

analyze shape or motion of an object in the scene because geometric information is not recovered in the IBR method.

Type II (Shape: Appearance, Color: Generic property): In order to recover the reflectance property of an object, an analysis based on a reflectance model with the 3D geometric property of an object is highly required. Accordingly, it is difficult to realize Type II, Nishino et al. [NSI01] and Tokai et al. [YXSM99], however, propose methods for recovering texture information under arbitrary lighting environments without utilizing reflectance models by applying eigen image analysis to multiple images that are captured under various lighting environments.

Type III (Shape: Generic property, Color: Appearance): A laser range finder can precisely measure depth to an object's surface viewed from the device. Using a laser range finder and a digital camera, Ikeuchi et al. reconstruct a shape and a texture of a huge object considering alignment of a large set of depth data and recovery of reflectance parameters [INHO03]. While the shape and the texture of an object are accurately reconstructed by the method, a moving object cannot be handled because measuring of the shape with a laser range finder takes a lot of time. A simple way for capturing a shape of a moving object is to utilize a video camera and analyze the captured images. Kanade et al. present to recover the shape and the texture of an object with a large set of cameras that are distributed on a hemisphere [KRN97]. They reconstruct the shape of an object with a stereo matching method and recovers a texture by extracting from the captured images. While the stereo matching is a method for reconstructing the accurate shape theoretically, there are some practical difficulties — varying an appearance depending on the baseline length, stereo matching for a uniform texture, etc., and therefore, a large set of cameras and a dense distribution are required for the method. Moezzi et al. present to reconstruct the shape by a volume intersection method, and recover a texture by coloring volumes of the captured object [MTG97]. Matusik et al. propose to generate free-viewpoint images with image-based visual hulls [MBR⁺00], where the shape is reconstructed by a volume intersection method, and the texture is recovered by the following procedure: 1) project rays from a desired viewpoint to the estimated shape, 2) back-project the rays to the captured images, and 3) calculate the colors of the rays by verifying the visibilities. Though the volume intersection method allows to reconstruct the shape by a fewer set of cameras, only

an approximate shape of an object is reconstructed theoretically. Hence, a texture mapping method for the shape of limited accuracy is required.

Type IV (Shape: Generic property, Color: Generic property): It is a method for reconstructing the 3D shape and recover the reflectance property of an object. It enables us to realize rich expressions that are difficult for other types, e.g. visualizing an object under an arbitrary lighting environment. In order to realize the expression, it is required to recover the reflectance property and to estimate the lighting environment.

Among the above types of the free-viewpoint image generation, one of the major characteristic of 3D video that we are developing is to generate the shape of a target object. It enables us to perform the two type of visualization, *high fidelity visualization* and *versatile visualization*. The high fidelity visualization realizes Type III, which visualizes 3D video with high fidelity to the input multi-view images. The other one, the versatile visualization, realizes Type IV, which illustrates the advantage of generating the shape of an object. We first estimate lighting environment in which 3D video object is captured with reference objects. We then estimate the reflectance parameters using the shape and the estimated lighting environment. Once the reflectance parameters are estimated, we can visualize 3D video with arbitrary effects. Moreover, the generation of the shape enables us to realize expressions utilizing the 3D shape, such as camerawork synchronizing with a motion of an object, and so on. This *editability* for geometry and photometry is the advantage over the IBR methods. In the following section, we introduce 3D video and its visualization.

1.1.2 3D Video

3D video is a real movie recording dynamic visual events in the real world as they are: time-varying 3D object shapes with a high-fidelity surface texture. In our laboratory, we have been conducting the 3D video project, where the following novel techniques have been developed:

- ◆ **Shape reconstruction with the volume intersection method:** It enables us to analyze shape and motion of a captured object. Furthermore, we have developed a real-time volume reconstruction system [WM03], and we can utilize the shape information for object tracking that will enable us to capture an arbitrary part of the object with specified quality.

6 Chapter 1. Introduction

- ◆ **Mesh deformation:** The shape of an object that is reconstructed with the volume intersection method is not perfectly accurate because the shape is represented as the visual hull of the object. We employ the deformable mesh model [NM04] to obtain the accurate shape of the object.
- ◆ **Texture mapping with viewpoint dependent vertex-based method:** It can generate high fidelity textures for the shape of limited accuracy, that is, the high fidelity visualization [TT02].
- ◆ **Visualization using geometric and photometric information:** We can visualize 3D video with versatile effects, that is, the versatile visualization.

Figure 1.1 illustrates the basic process of generating the shape of an object in our system:

1. **Synchronized Multi-View Image Acquisition:** A set of multi-view object images are taken simultaneously by a group of distributed video cameras (Figure 1.1; top row).
2. **Silhouette Extraction:** Background subtraction is applied to each captured image to generate a set of multi-view object silhouettes (Figure 1.1; second top row).
3. **Silhouette Volume Intersection:** Each silhouette is back-projected into the common 3D space to generate a visual cone enclosing the 3D object. Then, such 3D cones are intersected with each other to generate the voxel representation of the object shape (Figure 1.1; third bottom).
4. **Surface Shape Computation:** The discrete marching cubes method [KKI99] is applied to convert the voxel representation to the surface patch representation (Figure 1.1; second bottom).
5. **Mesh deformation:** We deform the surface patches with intrinsic and extrinsic force to obtain an accurate 3D mesh representation of the object (Figure 1.1; bottom).

In this thesis, we intensively discuss visualization of 3D video, that is, we discuss how to visualize 3D video effectively when the shape of an object (limited accuracy) and multi-view images are given.

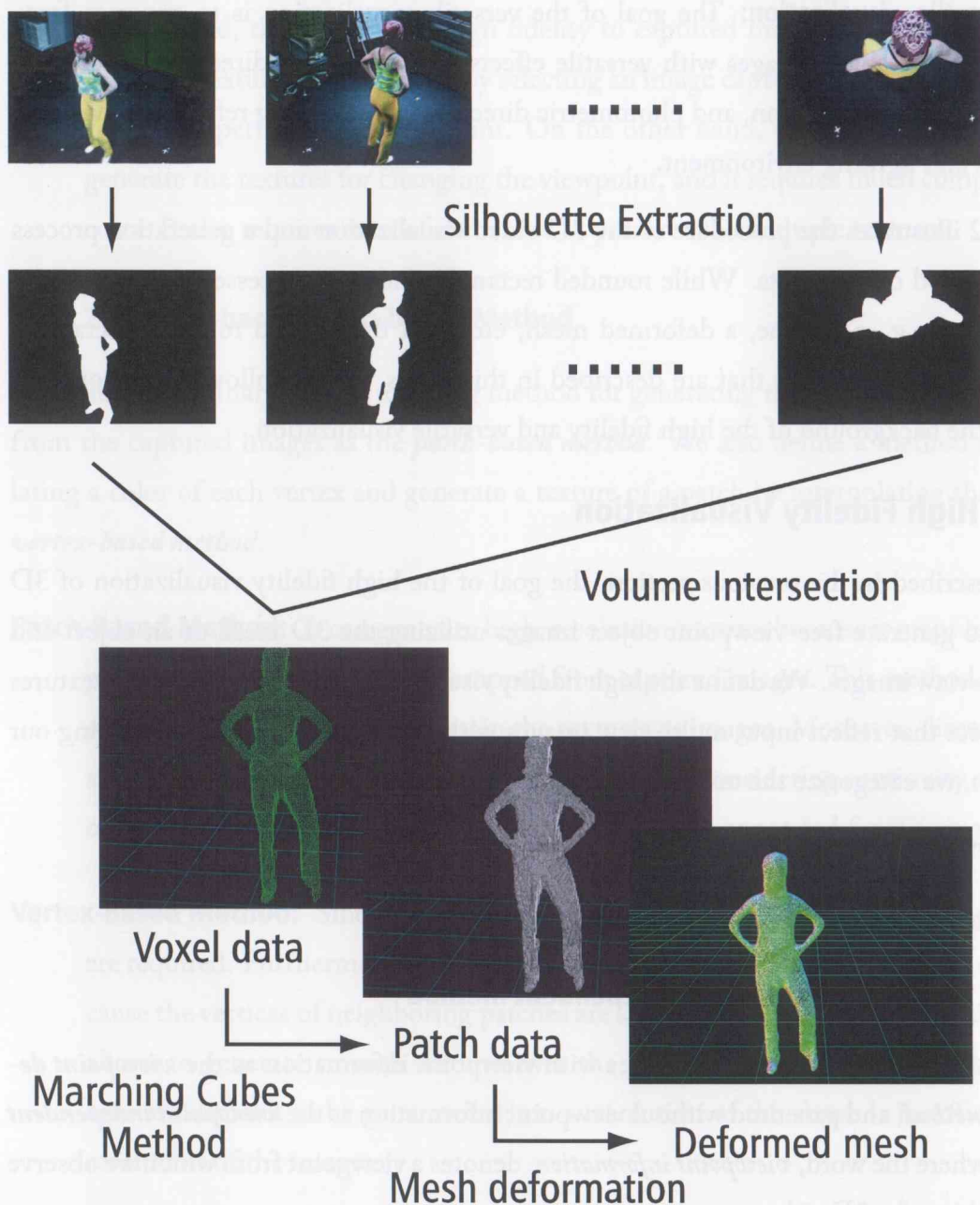


Figure 1.1: *Generation process of the shape of an object*

1.2 3D Video Visualization

We consider two different types of 3D video visualization. We call one of them high fidelity visualization, and the other versatile visualization. We define them as following:

- ◆ **High fidelity visualization:** The goal of the high fidelity visualization is to generate free-viewpoint object images that reflect input multi-view images with high fidelity.

- ◆ **Versatile visualization:** The goal of the versatile visualization is to generate free-viewpoint object images with versatile effects, i.e., geometric directing by utilizing a shape and/or motion, and photometric directing by estimating reflectance parameters and lighting environment.

Figure 1.2 illustrates the procedure of the 3D video visualization and a generation process of the required components. While rounded rectangles indicate processes to generate the components, e.g. a volume, a deformed mesh, etc., and the colored rounded-rectangles are our proposed methods that are described in this thesis. In the following sections, we describe the background of the high fidelity and versatile visualization.

1.2.1 High Fidelity Visualization

As we described in the previous section, the goal of the high fidelity visualization of 3D video is to generate free-viewpoint object images utilizing the 3D mesh of an object and the multi-view images. We define the high fidelity visualization as generation of the textures of an object that reflect input multi-view images with high fidelity. Before describing our algorithm, we categorize the texture mapping algorithm by

- ◆ **Viewpoint dependent/independent method,**
- ◆ **Patch-based/vertex-based method.**

1.2.1.1 Viewpoint Dependent/Independent Method

We define a method to visualize images with viewpoint information as the *viewpoint dependent method*, and a method without viewpoint information as the *viewpoint independent method*, where the word, *viewpoint information*, denotes a viewpoint from which we observe a target object for 3D video.

Viewpoint independent method: It generates a single texture for each patch. While this method has a capability of fast visualization once textures are generated, there is a hard problem of selecting an image for generation of textures among the multi-view images. If an inappropriate image is selected, discontinuity appears on the surface of the generated object.

Viewpoint dependent method: It generates textures by taking account of viewpoint information, that is, the textures dynamically vary when the viewpoint is moved. With

this method, the object with high fidelity to captured images can be visualized because the textures are generated by selecting an image captured by the camera that has similar properties to the viewpoint. On the other hand, the method requires to re-generate the textures for changing the viewpoint, and it requires much computational costs. It also requires a lot of memories to preserve the multiple textures.

1.2.1.2 Vertex-based/Patch-based Method

We define an ordinary texture mapping method for generating textures by cropping images from the captured images as the *patch-based method*. We also define a method for calculating a color of each vertex and generate a texture of a patch by interpolating them as the *vertex-based method*.

Patch-Based Method: It can generate high-resolution textures because it maps textures on patches of an object, which are cropped from captured images. This method, however, requires a lot of memories to retain the textures as images. Moreover, discontinuities and jitters appear on the surface of an object without considering continuity of textures of neighboring patches, even though the textures are generated from a single image.

Vertex-Based Method: Since it retains colors of vertices only, little amounts of memories are required. Furthermore, discontinuities never appear on the surface of an object because the vertices of neighboring patches are shared each other, and thus, this method allows to generate smooth textures over the surface. However, high-resolution images are difficult to be generated when a resolution of the shape is insufficient for a resolution of the images.

In Chapter 2, we first describe some problems in viewpoint independent patch-based method that we applied previously, and then propose a method to overcome the problems, that is, a viewpoint dependent vertex-based method.

1.2.2 Versatile Visualization

The versatile visualization of 3D video is a visualization with additional effects by utilizing characteristics of 3D video, that is, geometric information of an object. We further categorize the versatile visualization into the following type:

- ◆ **Geometric directing:** By analyzing the shape of an object, we can generate camerawork for visualizing 3D video, e.g. visualizing the face of an object all the time. Furthermore, by analyzing a motion of an object, we can obtain camerawork that visualizes a traditional dance beautifully, or an action of an athlete intelligibly.
- ◆ **Photometric directing:** It visualizes 3D video by adding or eliminating the effects of lighting. For the photometric directing, reflectance parameters of an object is required. Several methods for estimation of reflectance parameters are proposed using RGB images, a spectrometer, polarization, etc. In this thesis, we utilize the captured multi-view images, and therefore a lighting environment of the scene including the object is required for the reflectance parameter estimation. We describe the background of a lighting environment estimation in Section 1.3, and present effective methods for lighting environment estimation for real scenes in Chapters 3 and 4.

The versatile visualization is one major advantage of 3D video over the IBR method since we can observe the object of 3D video from an arbitrary viewpoint and with an arbitrary appearance.

In the following sections, we review related work of what we call the versatile visualization.

1.2.2.1 Geometric Directing

We define the geometric directing as camerawork generation using object's shape and motion. Movies and TV dramas are created by employing rules of camerawork that was established through long years [Ari76]. A few work for the camerawork generation that based on the rules using computer vision techniques are, however, proposed [HCS96, OMS99, TM02], because camerawork is highly depend on subjectivity of a human that is hard to be analyzed. Moreover, He et al. and Ota et al. propose to generate camerawork in CG world [HCS96, OMS99], and Tokai and Miyagawa propose to generate it utilizing 2D images [TM02]. Thus, 3D video objects generated from real scene are suitable for the study of the camerawork generation, because they have the 3D geometry and are not virtual. In this thesis, we propose algorithms for semi-automatically generating camerawork utilizing the shape of the 3D video object.

1.2.2.2 Photometric Directing

For the photometric directing, three components are required, i.e., shape, reflectance parameter, and lighting environment. Because 3D video has the shape information, other two components are need to be estimated. In general, it is difficult to estimate reflectance parameters under natural lighting environment. The two components are therefore estimated independently. In this thesis, we mainly discuss the acquisition of the two components in order to realize the photometric directing of 3D video.

Reflectance property: Sato et al. propose to estimate object shape and reflectance parameter. They assume the specular component is directly measured by a calibration procedure, and estimate the diffuse component by solving the equation of reflection [SWI97]. Machida et al. propose to estimate reflectance parameters including diffuse and specular components under controlled lighting environment, considering mutual reflections and self-shadows [MYT03]. Hara et al. propose to estimate reflectance parameters and simple lighting environment simultaneously utilizing polarization filters [HNI03]. Tan and Ikeuchi propose to separate reflection components utilizing intensity and chromaticity [TI03]. Though they do not estimate reflectance parameters, the method is effective for analyzing images to estimate the reflection components. Nevertheless, Fei et al. propose to estimate relative reflectance parameters of a convex object under natural lighting environment, assuming the reflection as Lambertian [DOSS04]. They can estimate the reflectance parameters by roughly estimating the lighting environment, because the Lambertian reflection hardly reflects detailed variations of the lighting environment.

In this thesis, we estimate the reflectance parameter of an object assuming as a Lambertian reflection, because the limited accuracy of the shape causes some difficulties for estimation of specular components. The estimation of the reflectance parameters is described in Section 5.3.3.

Lighting environment: We give the detailed descriptions of the lighting environment estimation in the next section.

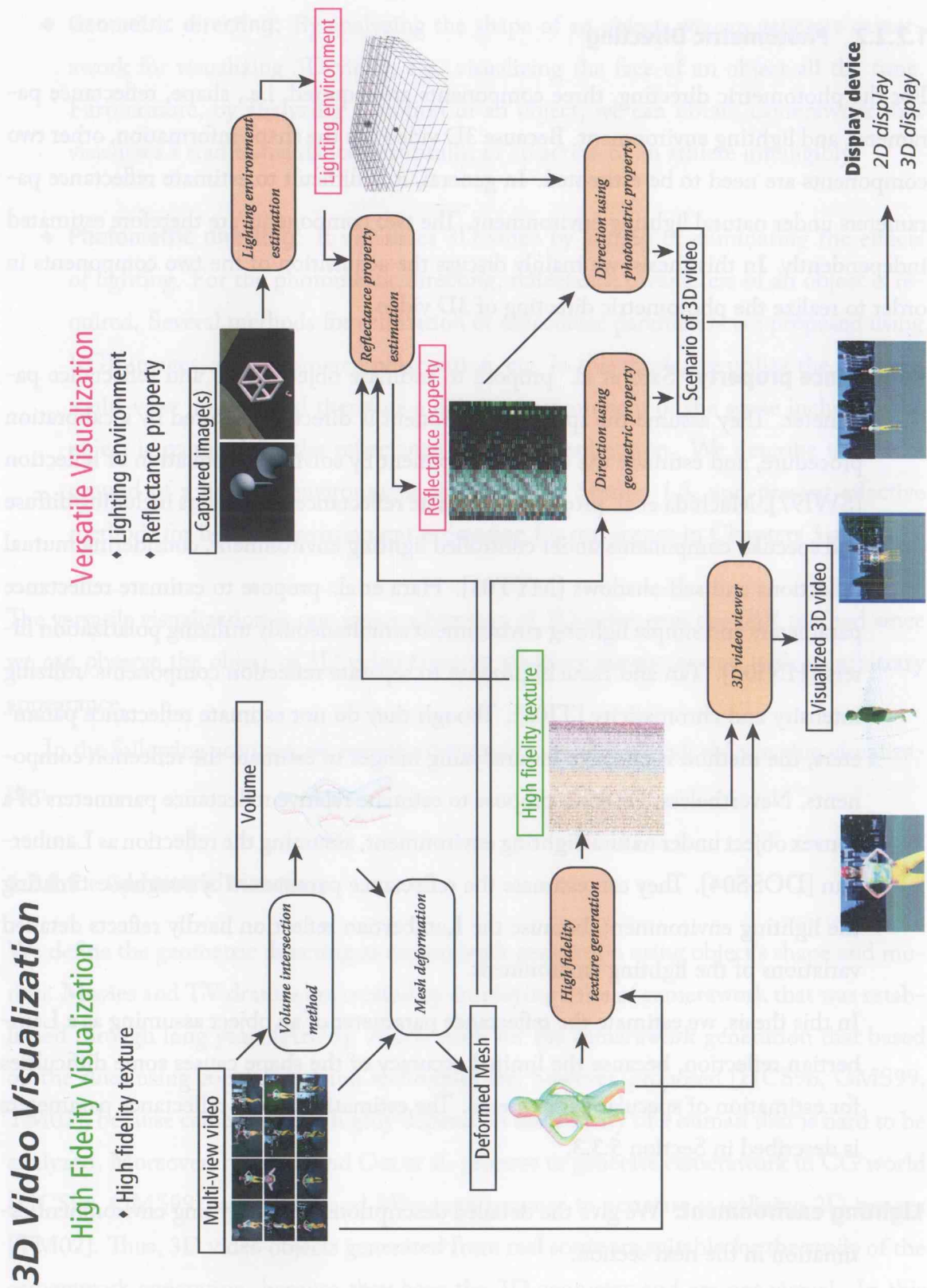


Figure 1.2: Procedure of 3D Video Visualization

1.3 Lighting Environment Estimation

Lighting environment estimation is one of the most important research topics in computer vision. If it is possible to derive the lighting information from an input image of a real scene, it enables us to superimpose virtual objects into the scene with natural shading and shadows by controlling the lighting effects in the generated or captured images accordingly. The applications of the lighting environment estimation are widespread, such as object tracking, face recognition, VR and graphics.

This chapter presents an introduction of Chapter 3 and Chapter 4. We first review related work and then describe our methods for lighting environment estimation—the *skeleton cube* (described in Chapter 3) and the *difference sphere* (described in Chapter 4), especially combination of the two methods.

1.3.1 Related Work

1.3.1.1 Inverse Lighting

Pioneering work includes those of estimating illuminant directions in the framework of shape from shading. Pentland [Pen82] proposed a method for estimating the illuminant direction of a single light source using the Lambertian reflectance model and the maximum-likelihood estimation. Zheng and Chellappa [ZC91] presented a practical approach by enforcing smoothness assumption and a hierarchical implementation. The term, inverse lighting, was used by Marschner and Greenberg [MG97] who proposed a technique for estimating directions and radiant intensities of a few light sources in a scene by a least-squares method for surface intensity functions also assuming Lambertian reflectance model. Some approaches exploit the occluding boundary that puts strong constraints on the light source directions using an object that has (locally) Lambertian surface [NE01, YY91]. Also, some other inverse lighting approaches [WS02, ZY01, ZK02], employ images containing a reference object with known shape and reflectance property, and estimate the lighting environment by analyzing its shading, or highlights.

In fact, as stated earlier, Ramamoorthi and Hanrahan [RH01] and Basri and Jacobs [BJ03] prove that the estimation of lighting environment with the Lambertian reflectance model is ill-posed or numerically ill-conditioned because the Lambertian reflectance model performs low-pass filtering of the lighting environment. Sato et al. [SSI99a] on the other hand propose a method which characterizes the lighting environment by a set of directional

light sources and estimates the radiant intensities of the light sources by analyzing pixel values of cast-shadows on a plane with known/unknown bidirectional reflectance distribution function (BRDF). In the method the radiant intensities can be efficiently estimated even in the case of the Lambertian reflectance model because the information of cast-shadows involves non-linearity and therefore alleviate the drawback due to Lambertian assumption.

Nevertheless, since most of these methods represent the lighting environment as a set of directional light sources, it is difficult to apply them to certain types of real scene, especially to an indoor scene, in which it is required to consider the effects such as attenuation of radiant intensity caused by the positional relationship between the light source and the object. Although a few recent advances [HNI03, PSG01] allow estimation of near light source, it is still a challenging problem to solve for parameters of multiple light sources including their radiant intensity.

1.3.1.2 Lighting Environment Representation

There are several possibilities in representing lighting environment depending on the assumptions as well as the compatibility to the approaches of lighting estimation. Debevec [Deb98] uses a mirror sphere as an omni-directional light probe to measure the radiance distribution. While the estimated lighting environment represents accurate irradiance at the center of the specular sphere, the irradiance do not vary wherever in the scene, and therefore, it is uncertain if the effects of point light source which we wish to deal with are representable.

Sato et al. [SSI99b] estimate the irradiance for each direction using a pair of omni-directional images that are captured with a fish-eye lens. They estimate positions of light sources by matching the highlights on the images, and represent the lighting environment by a polyhedron whose vertices denote light sources and surfaces denote the irradiance. Although this method takes effects derived from the positions into account, radiant intensity of the lighting environment cannot be represented because intensity of the captured images only denotes the irradiance to the points of cameras' projection centers.

Nishino and Nayar propose to utilize eyes in a photograph for recognizing the real world [NN04b, NN04a]. In [NN04a], they propose to estimate the lighting environment with precise analysis of the eyes, the estimated lighting environment is, however, a distribution of infinite point light sources.

1.3.2 Our Methodology: Skeleton Cube and Difference Sphere

As described in the previous section, estimating a lighting environment of a real scene is a difficult problem.

The direct methods, e.g. utilizing a mirror sphere or a fish-eye lens, employ a simple way to represent the lighting environment by a captured image that has a high dynamic range of intensities. The image, however, does not represent *radiant intensity* of the lighting environment, but *irradiance* to the center of the sensor, i.e., the center of the mirror sphere or the optical center of the fish-eye lens. Therefore, the direct method can only deal with a lighting environment as a distribution of infinite light sources, and it does not allow a representation of a lighting environment as a set of near light sources. Even though positions of light sources are estimated from multiple images with a stereo matching method, they are not near light sources because they cannot represent an effect of attenuation depending on distance.

On the other hand, the indirect method—a lighting environment estimation with a reference object—enables us to estimate *radiant intensity* of light sources. Nevertheless, it is difficult to estimate radiant intensity without an assumption of a lighting environment, such as light sources' type, distribution, etc. Moreover, an estimation of complex lighting environment from shading of a reference object is ill-posed or numerically ill-conditioned. Note that radiant intensity cannot be estimated from specular reflection because of the same reason as in the direct method.

According to the above discussion, we propose two types of methods for a lighting environment estimation of real scenes using reference objects, i.e., the *skeleton cube* and the *difference sphere*.

Skeleton cube: In the method utilizing the skeleton cube we estimate lighting environment by analyzing shading and self-shadows of it. The use of skeleton cube is in spirit similar to the above technique of using cast-shadows [SSI99a], but it is very different in that the design of the skeleton cube allows self-shadows to occur. That is, since we have self-shadows under light sources at almost any positions, we just need to examine its inside surfaces (without debating about what cast-shadow to observe from which direction). Also, it is convenient that we do not have to worry about the size of shadows that can be observed in the input image. We first model an approximate distribution of point and area light sources in a scene a priori as *foreground*

lighting and *background lighting*, where the foreground lighting denotes near point light sources and the background lighting denotes ambient lighting, e.g. lighting or reflection from walls or ceiling. We then estimate the radiant intensity of the two sorts of lighting simultaneously. This method requires a model of distribution of light sources but practical for real scenes.

Difference sphere: The method utilizing the difference sphere offers a precise estimation of light sources without prior knowledge of a lighting environment, i.e., light sources' types and positions. The difference sphere is a virtual sphere generated by differencing the radiance on the surfaces of a pair of reference spheres, and therefore its radiance only reflects effects of near light sources whereas effects of infinite light sources are canceled out. We estimate parameters of the light sources by analyzing both geometric and photometric properties of the difference sphere. It is to be noted that the application of the method to a complex lighting environment may not be straight forward because of its estimation algorithm. Note that we can estimate such a complex lighting environment by using the skeleton cube.

Chapter 3 and Chapter 4 present detailed descriptions of the two methods for lighting environment estimation, respectively. Furthermore, we describe the combination of the two methods for practical estimation of the real lighting environment in Section 5.3.2.

1.4 Overview of Thesis

This thesis consists of six chapters.

In Chapter 1, we have presented the background of this thesis; the high fidelity and versatile visualization of 3D video.

Chapter 2 presents a novel texture mapping algorithm to generate high fidelity 3D video. We propose the viewpoint dependent vertex-based method that can generate high fidelity and smooth textures from multi-view images for an object whose shape is limited accuracy. We show the effectiveness of the method by qualitative and quantitative evaluations.

Chapter 3 presents a practical method for a lighting environment estimation with the skeleton cube. The skeleton cube is a small hollow cube whose shadows are casted on itself, i.e. it has self-shadows. The key ideas of our method are 1) the estimation of lighting environment by analyzing self-shadows and shading of the skeleton cube, and 2) the repre-

sentation of lighting environment by plenty of point and area light sources. We verify the advantages of the skeleton cube, demonstrate the performance with CG simulations and an estimation of a real scene.

Chapter 4 presents a precise method for a lighting environment estimation with the difference sphere. The method estimates lighting sources from a single image of a scene that is illuminated by near point light sources, directional light sources and ambient light. We estimate parameters of the light sources by analyzing both geometric and photometric properties of reference spheres. We demonstrate the effectiveness of the entire algorithm with performance evaluations for a CG scene and a real scene.

Chapter 5 presents the versatile visualization of 3D video, which includes the geometric directing and the photometric directing. We also present a method for estimating reflectance parameters of an object utilizing multi-view images and the estimated lighting environment. We demonstrate the possibilities of the geometric and photometric directing with visualized images, and we finally present the 3D video viewer that we have developed for a visualization of 3D video.

Chapter 6 summarizes the thesis and concludes with future work of the high fidelity and versatile visualization of 3D video.

Chapter 2

High Fidelity Visualization of 3D video

2.1 Introduction

In this chapter, we propose a novel texture mapping algorithm for the high fidelity visualization of 3D video. The problem we are going to solve here is how we can generate high fidelity object images from arbitrary viewpoints based on the 3D object shape of limited accuracy. That is, the computed 3D mesh model is just an approximation of the real 3D object shape and include considerable amount of noise.

The input data for this algorithm are

- ◆ A temporal series of 3D mesh data. We prepare two types of 3D mesh data, which are
 - ❖ *Mesh*: It was obtained by applying the discrete marching cubes method [KKI99] to the voxel data.
 - ❖ *D-mesh*: It was obtained by applying the intra-frame deformation to each visual hull individually. This is because, as noted at the end of the previous section, the current inter-frame deformation method cannot cope with global topological structure changes. Since the texture mapping algorithm proposed here conducts rendering frame by frame, such mesh data do not cause any problem even if the mesh topology changes frame by frame.
- ◆ A temporal series of multi-view video data.
- ◆ Camera calibration data for all cameras.

In the following sections, we first describe the system organization for capturing multi-view video. Reviewing the viewpoint independent patch-based method as a naive algorithm, we describe our proposed method, i.e., the viewpoint dependent vertex-based method in detail. In Section 2.5, we show performance of our method, and conclude with discussions.

2.2 3D Video Capturing System Organization

Figure 2.1 illustrates the hardware organization of our real-time active 3D object behavior reconstruction system. It consists of

- ◆ **PC cluster:** 16 node PCs (dual Pentium III 600MHz) are connected through Myrinet, an ultra high speed network (full duplex 1.28Gbps). PM library for Myrinet PC clusters [THIS97] allows very low latency and high speed data transfer, based on which we can implement efficient parallel processing on the PC cluster.
- ◆ **Distributed active video cameras:** Among 16, 12 PCs have fixed-viewpoint pan-tilt-zoom (FV-PTZ) cameras [Mat98], respectively, for active object tracking and image capturing. In the FV-PTZ camera, the projection center stays fixed irrespectively of any camera rotations and zoomings, which greatly facilitates real-time active object tracking and 3D shape reconstruction. Note that this capturing system does not support a hardware trigger and therefore the captured multi-view data are not completely synchronized and include motion blur when a human moves quickly.

2.3 Viewpoint Independent Patch-Based Method

We first implemented a naive texture mapping algorithm, which selects the most “appropriate” camera for each patch and then maps onto the patch the texture extracted from the image observed by the selected camera. Since this texture mapping is conducted independently of the viewer’s viewpoint of 3D video, we call it the Viewpoint Independent Patch-Based Method (VIPBM).

Algorithm (Figure 2.2)

1. For each patch p_i , do the following processing.
2. Compute the locally averaged normal vector V_{lmn} using normals of p_i and its neighboring patches.

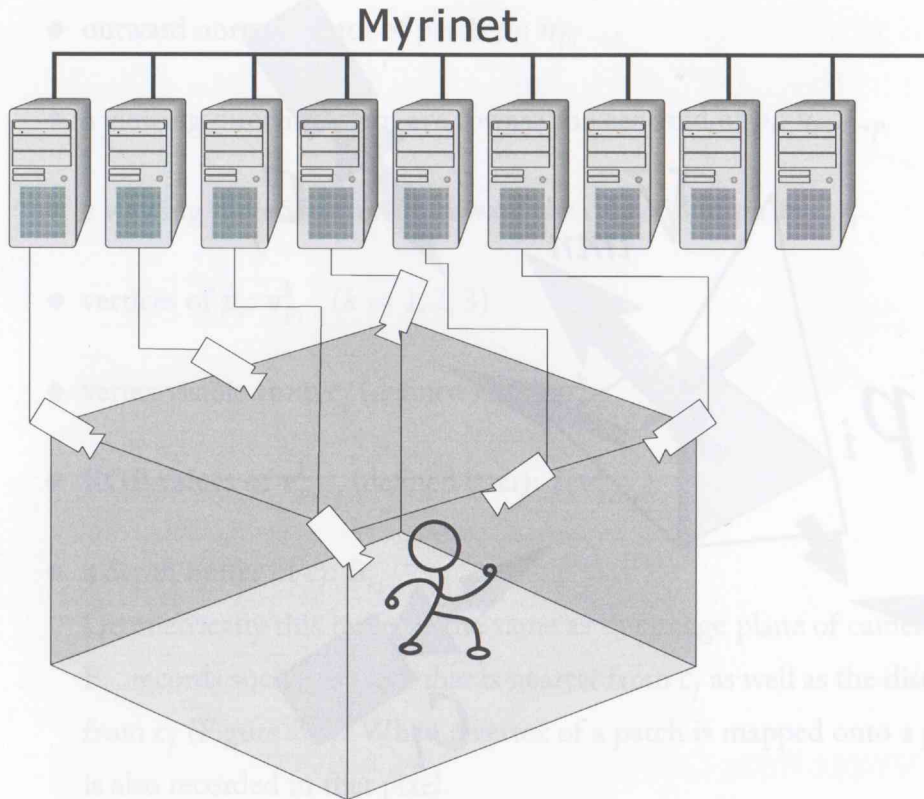


Figure 2.1: *PC cluster for real-time active 3D object behavior reconstruction system.*

3. For each camera c_j , compute viewline vector V_{c_j} directing toward the centroid of p_i .
4. Select such camera c^* that the angle between $V_{l_{mn}}$ and V_{c_j} becomes maximum.
5. Extract the texture of p_i from the image captured by camera c^* .

This method generates a fully textured 3D object shape, which can be viewed from arbitrary viewpoints with ordinary 3D graphics display systems. Moreover, its data size is very compact compared with that of the original multi-view video data.

From the point of view of fidelity, however, the displayed image quality is not satisfying:

1. Due to the rough quantization of patch normals, the best camera c^* for a patch varies from patch to patch even if they are neighboring. Thus, textures on neighboring patches are often extracted from those images captured by different cameras (i.e. viewpoints), which introduces jitters in displayed images.
2. Since the texture mapping is conducted patch by patch and their normals are not accurate, textures of neighboring patches may not be smoothly connected. This introduces jitters at patch boundaries in displayed images.

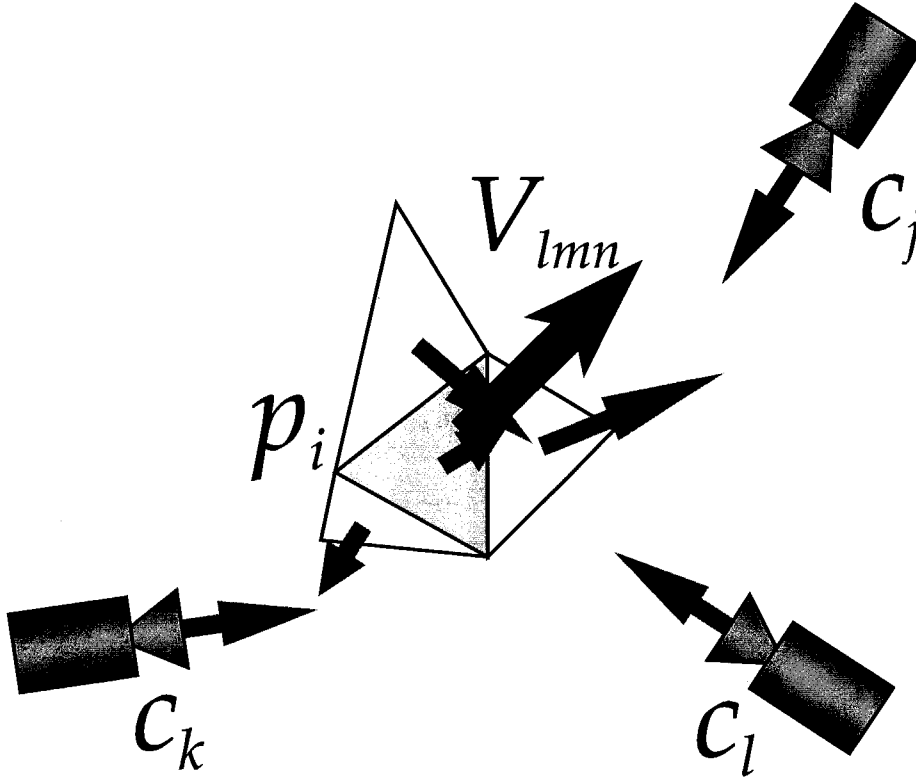


Figure 2.2: *Viewpoint independent patch-based method*

To overcome these quality problems, we developed a viewpoint dependent vertex-based texture mapping algorithm. In this algorithm, the color (i.e. RGB value) of each patch vertex is computed taking into account the viewpoint of a viewer, and then the texture of each patch is generated by interpolating the color values of its three vertices.

2.4 Viewpoint Dependent Vertex-Based Method

2.4.1 Definitions

First of all, we define words and symbols as follows (Figure 2.3), where bold face symbols denote 3D position/direction vectors:

- ◆ a group of cameras: $C = \{c_1, c_2, \dots, c_n\}$
- ◆ a viewpoint for visualization: **eye**
- ◆ a set of surface patches: $P = \{p_1, p_2, \dots, p_m\}$

- ◆ outward normal vector of patch p_i : $\mathbf{n}_{p_i}$
- ◆ a viewing direction from **eye** toward the centroid of p_i : $\mathbf{v}_{eye \rightarrow p_i}$
- ◆ a viewing direction from c_j toward the centroid of p_i : $\mathbf{v}_{c_j \rightarrow p_i}$
- ◆ vertices of p_i : $\mathbf{v}_{p_i}^k$ ($k = 1, 2, 3$)
- ◆ vertex visible from c_j (defined later): $\mathbf{v}_{p_i, c_j}^k$
- ◆ RGB values of $\mathbf{v}_{p_i, c_j}^k$ (defined later): $I(\mathbf{v}_{p_i, c_j}^k)$
- ◆ a depth buffer of c_j : $\mathbf{B}_{c_j}$

Geometrically this buffer is the same as the image plane of camera c_j . Each pixel of $\mathbf{B}_{c_j}$ records such patch ID that is nearest from c_j as well as the distance to that patch from c_j (Figure 2.4). When a vertex of a patch is mapped onto a pixel, its vertex ID is also recorded in that pixel.

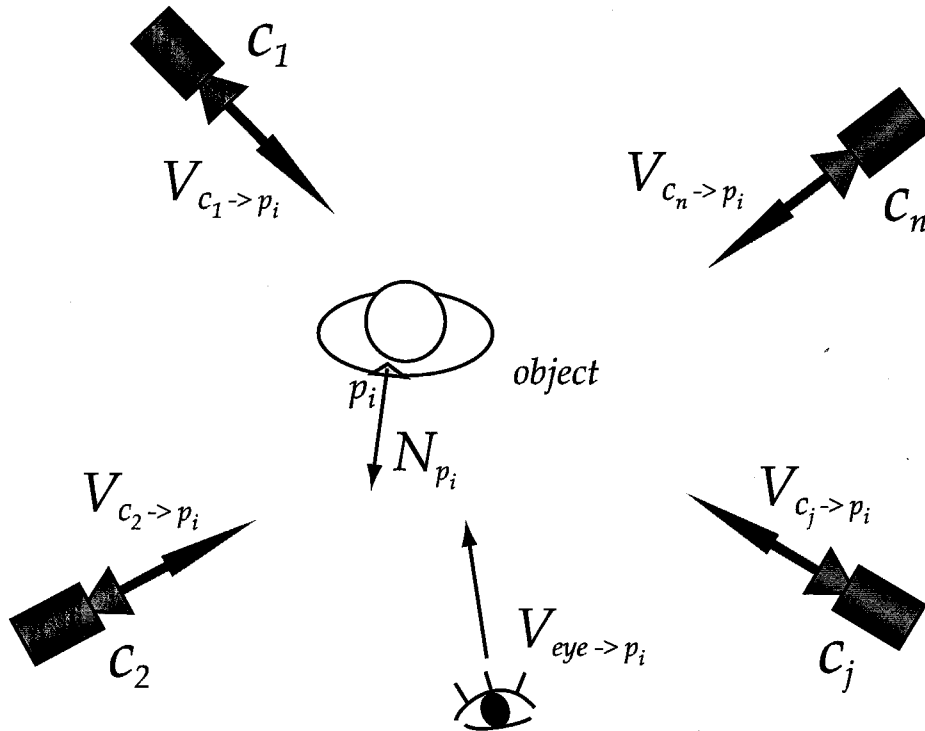
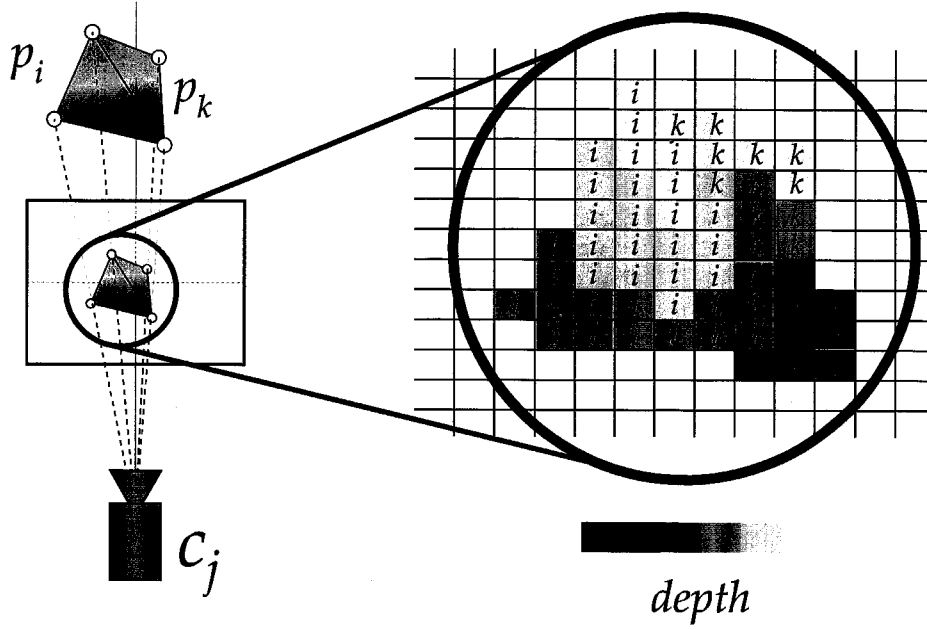


Figure 2.3: Viewpoint and camera position


 Figure 2.4: *Depth buffer*

2.4.2 Visible Vertex from Camera c_j

The vertex visible from c_j , $\mathbf{v}_{p_i, c_j}^k$, is defined as follows.

1. The face of patch p_i can be observed from camera c_j , if the following condition is satisfied.

$$\mathbf{n}_{p_i} \cdot \mathbf{v}_{c_j \rightarrow p_i} < 0 \quad (2.1)$$

2. $\mathbf{v}_{p_i}^k$ is not occluded by any other patches.

Then, we can determine $\mathbf{v}_{p_i, c_j}^k$ by the following process:

1. First, project all the patches that satisfy equation (2.1) onto the depth buffer B_{c_j} .
2. Then, check the visibility of each vertex using the buffer. Figure 2.5 illustrates possible spatial configurations between a pair of patches: all the vertices in type (1) and (2) are visible, while in type (5) three vertices of the occluded patch are not visible. In type (3) and (4), only some vertices are visible.

RGB values $I(\mathbf{v}_{p_i, c_j}^k)$ of the visible vertex $\mathbf{v}_{p_i, c_j}^k$ are computed by

$$I(\mathbf{v}_{p_i, c_j}^k) = I_{c_j}(\hat{\mathbf{v}}_{p_i, c_j}), \quad (2.2)$$

where $I_{c_j}(\mathbf{v})$ shows RGB values of pixel $\mathbf{v}$ on the image captured by camera c_j , and $\hat{\mathbf{v}}_{p_i, c_j}^k$ denotes the pixel position onto which the vertex $\mathbf{v}_{p_i, c_j}^k$ is mapped by the imaging process of camera c_j .

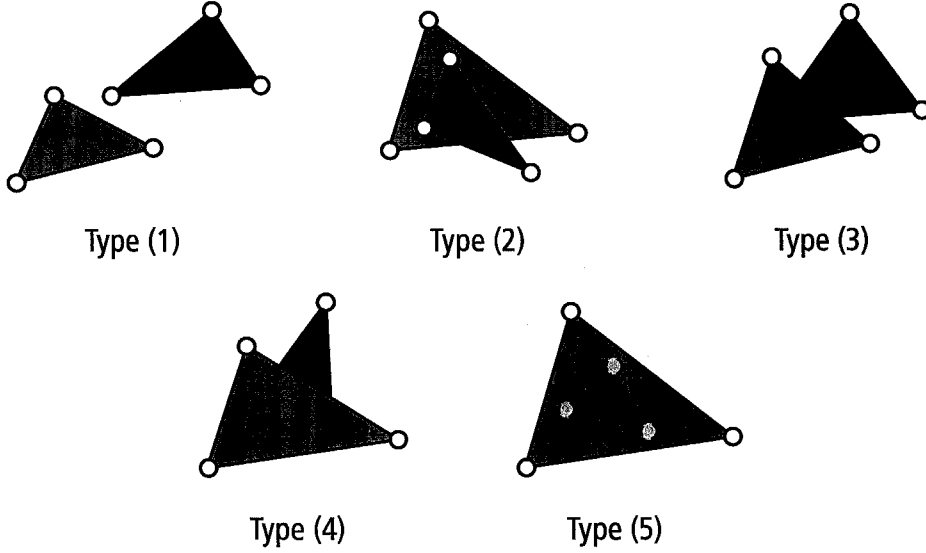


Figure 2.5: *Relations between patches*

2.4.3 Algorithm

1. Compute RGB values of all vertices visible from each camera in $C = \{c_1, c_2, \dots, c_n\}$.
2. Specify the viewpoint **eye**.
3. For each surface patch $p_i \in P$, do 4 to 9.
4. If $\mathbf{v}_{eye \rightarrow p_i} \cdot \mathbf{n}_{p_i} < 0$, then do 5 to 9.
5. Compute weight $w_{c_j} = (\mathbf{v}_{c_j \rightarrow p_i} \cdot \mathbf{v}_{eye \rightarrow p_i})^m$, where m is a weighting factor to be specified a priori.
6. For each vertex $\mathbf{v}_{p_i}^k$ ($k = 1, 2, 3$) of patch p_i , do 7 to 8.
7. Compute the normalized weight for $\mathbf{v}_{p_i}^k$ by

$$\bar{w}_{c_j}^k = \frac{w_{c_j}^k}{\sum_l w_{c_l}^k}. \quad (2.3)$$

Here, if $\mathbf{v}_{p_i}^k$ is visible from camera c_j , then $w_{c_j}^k = w_{c_j}$, else $w_{c_j}^k = 0$.

8. Compute the RGB values $I(\mathbf{v}_{p_i}^k)$ of $\mathbf{v}_{p_i}^k$ by

$$I(\mathbf{v}_{p_i}^k) = \sum_{j=1}^n \bar{w}_{c_j}^k I(\mathbf{v}_{p_i, c_j}^k) \quad (2.4)$$

9. Generate the texture of patch p_i by linearly interpolating RGB values of its vertices. To be more precise, depending on the number of vertices with non-zero RGB values, the following processing is conducted:

- ◆ **3 vertices:** Generate RGB values at each point on the patch by linearly interpolating the RGB values of 3 vertices .
- ◆ **2 vertices:** Compute mean values of the RGB values of the 2 vertices, which is regarded as those of the other vertex. Then apply the linear interpolation on the patch.
- ◆ **1 vertex:** Paint the patch by the RGB values of the vertex.
- ◆ **no vertex:** Texture of the patch is not generated: painted by black for example.

By the above process, an image representing an arbitrary view (i.e from **eye**) of the 3D object is generated.

2.5 Performance Evaluation

To evaluate the performance of the proposed viewpoint dependent vertex-based method (VDVBM), we first compare it with the viewpoint independent patch-based method (VIPBM) qualitatively. Figures 2.6 and 2.7 show images generated by VIPBM and VDVBM, respectively, for the same 3D mesh data of Mesh and D-mesh. First, we can observe the difference between VIPBM and VDVBM, that is, VIPBM introduces many jitters in images, which are considerably reduced by VDVBM. Second, we can observe the difference between Mesh and D-mesh, that is, the quality of VIPBM with D-mesh becomes better because the surface normals are smoothed by the deformation of the 3D mesh, while the results by VDVBM with D-mesh do not show much differences because the textures that are generated by VDVBM do not depend on the surface normals but the viewpoint.

Then, we conducted quantitative performance evaluations with Mesh and D-mesh. That is, we calculate RGB root-mean-square (rms) errors between a real image captured

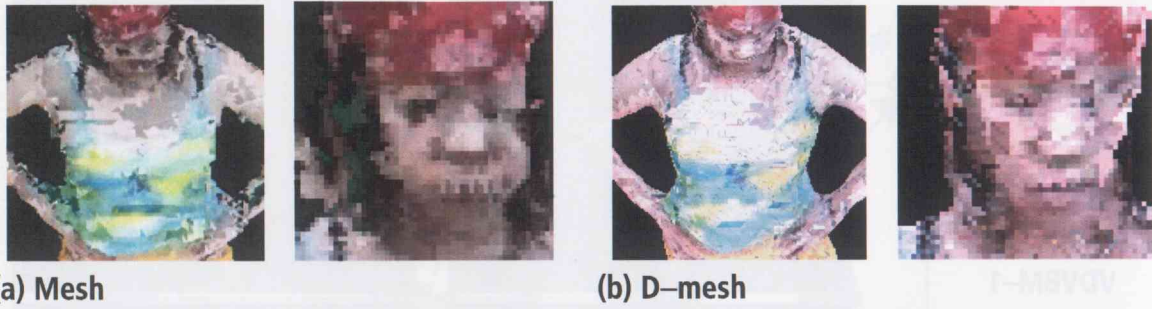


Figure 2.6: *Images generated by the Viewpoint Independent Patch-Based Method*

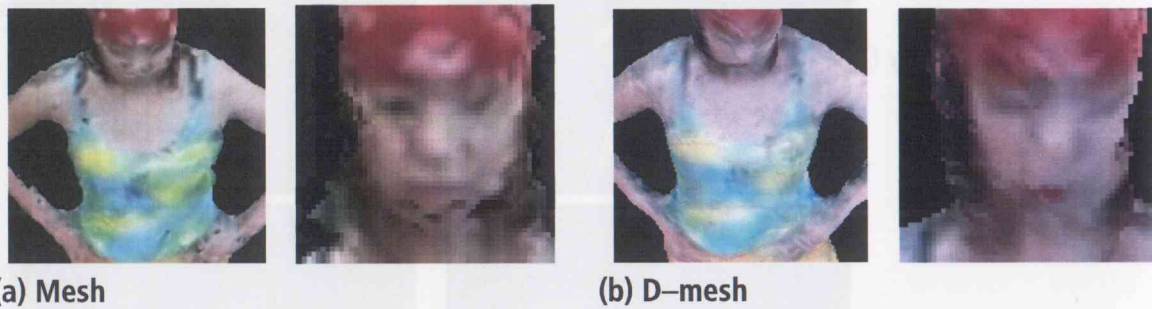


Figure 2.7: *Images generated by the Viewpoint Dependent Vertex-Based Method*

by camera c_j and its corresponding images generated by VIPBM and VDVBM, respectively: in generating the images, the position and direction of camera c_j are used as those of the viewpoint for the 3D video (i.e. **eye** in VDVBM). To evaluate the performance of VDVBM, we employed two methods: VDVBM-1 generates images by using all multi-view video data including real images captured by camera c_j itself, while VDVBM-2 excludes such real images captured by camera c_j . The experiments were conducted under the following settings:

- ◆ camera configuration: Figure 2.9
- ◆ image size: 640×480 [pixel] 24 bit RGB color
- ◆ viewpoint: camera 5
- ◆ weighting factor in VDVBM: $m = 5$

Figure 2.10 and Figure 2.11 illustrate the experimental results, where rms errors for frame 101 to 140 are computed. This figure proves that VDVBM performs better than VIPBM with both Mesh and D-mesh. The superiority of VDVBM and its high fidelity

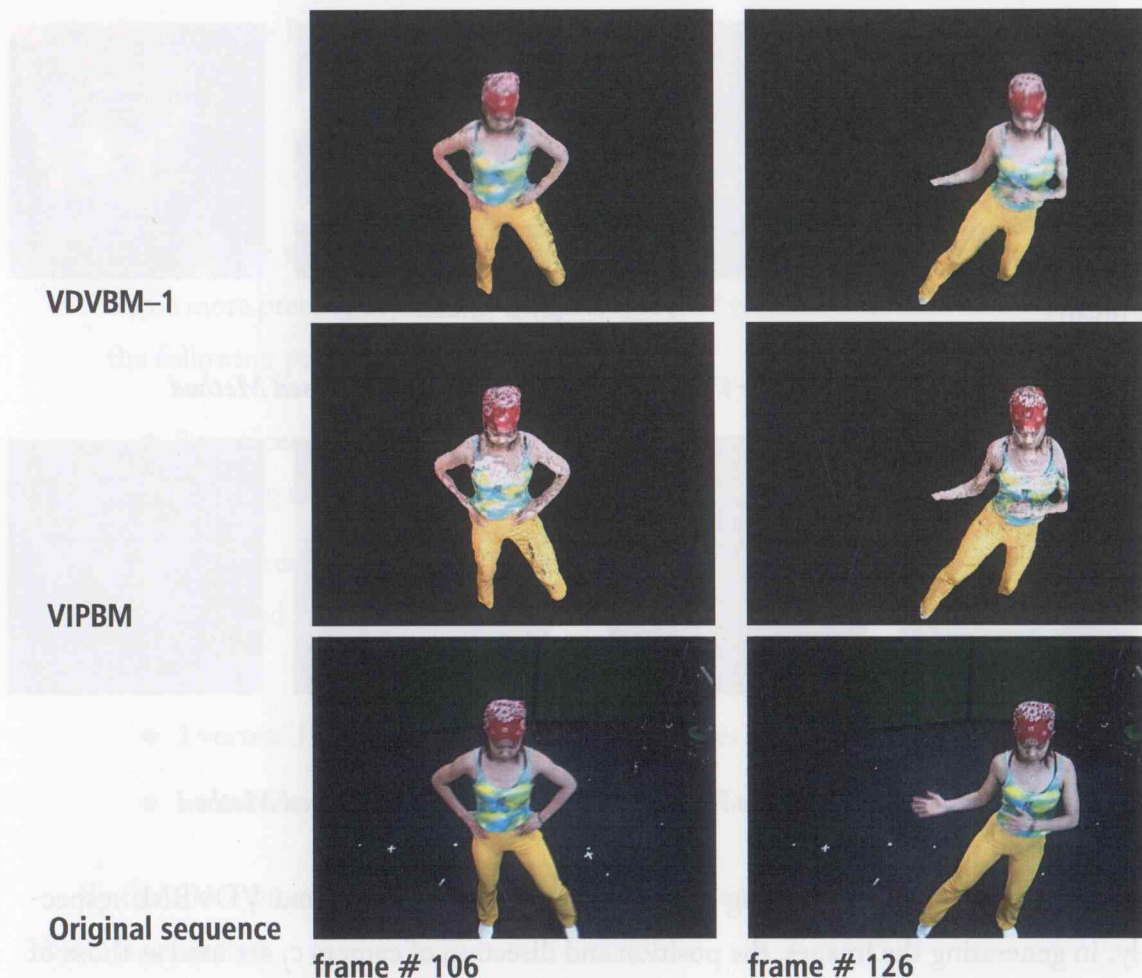


Figure 2.8: *Sample images of generated 3D video with D-mesh*

image generation capability can be easily observed in Figure 2.8, where real and generated images for frames 106 and 126 are illustrated. The rms errors of VIPBM with D-mesh become better because of the reason that we described in the previous evaluation.

Finally, we tested how we can improve the performance of VDVBM by increasing the spatial resolution of patch data. (The all of following examinations are conducted with D-mesh.) Figure 2.12 shows the method of subdividing a patch into three (S3) and six (S6) sub-patches to increase the spatial resolution.

Then, we examine the average side length of a patch on the image plane of each camera by projecting original and subdivided patches onto the image plane. Figure 2.13 shows the mean side length in pixel on the image plane of each camera. Note that since camera 9 is located closer to the 3D object (see Figure 2.9), object images captured by it become larger than those by the other cameras, which caused bumps (i.e. larger side length in pixel) in the

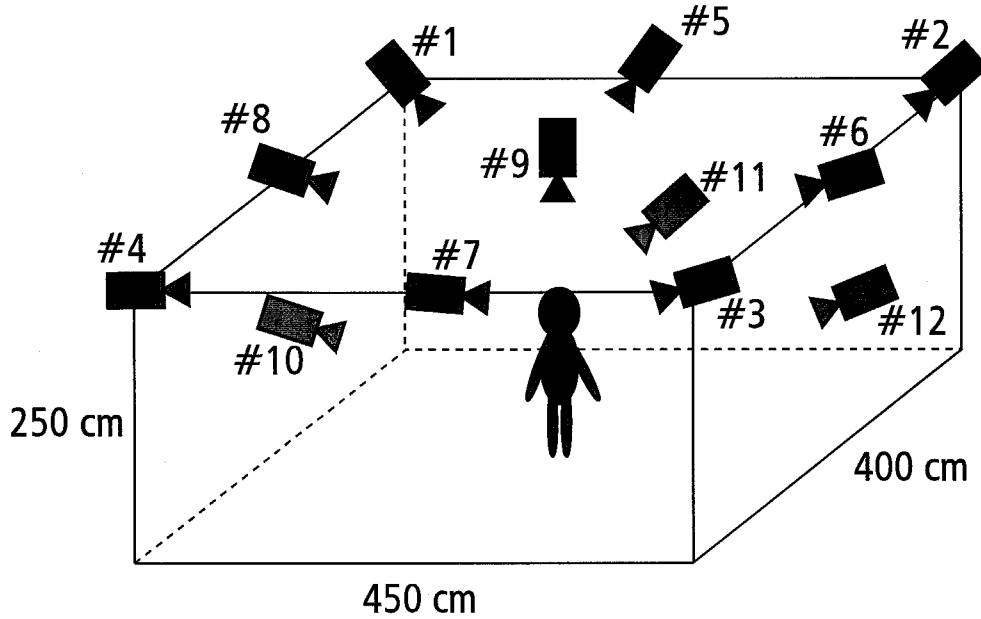


Figure 2.9: *Camera Setting*

graphs in Figure 2.13.

We can observe that the spatial resolution of S6 is approximately the same as that of an observed image (i.e. 1 pixel). That is, S6 attains the finest resolution, which physically represents about 5mm on the object surface. To put this in another way, we can increase the spatial resolution up to the six sub-division, which can improve the quality of images generated by VDVBm.

To quantitatively evaluate the quality achieved by using subdivided patches, we calculated root-mean-square errors between real images and images generated by VDVBm-1 with original, S3, and S6, respectively. The root-mean-square errors are shown in Figure 2.14.

Figure 2.14 shows that subdividing patches does not numerically reduce the errors. The reasons of this observation can be considered as follows. We see that most of the errors arise around the contour of the object and edges of texture (e.g. an edge between skin and clothes, etc.) (Figure 2.15), and they are difficult to be reduced by subdividing patches because they come from motion blur or asynchronization, i.e. capturing the images is not perfectly synchronized. The errors are thus not reduced in total even if they may be decreased in some parts. Fidelity of generated images using subdivided patches, however, is definitely improved (Figure 2.16). Accordingly, subdividing patches is effective from a fidelity point

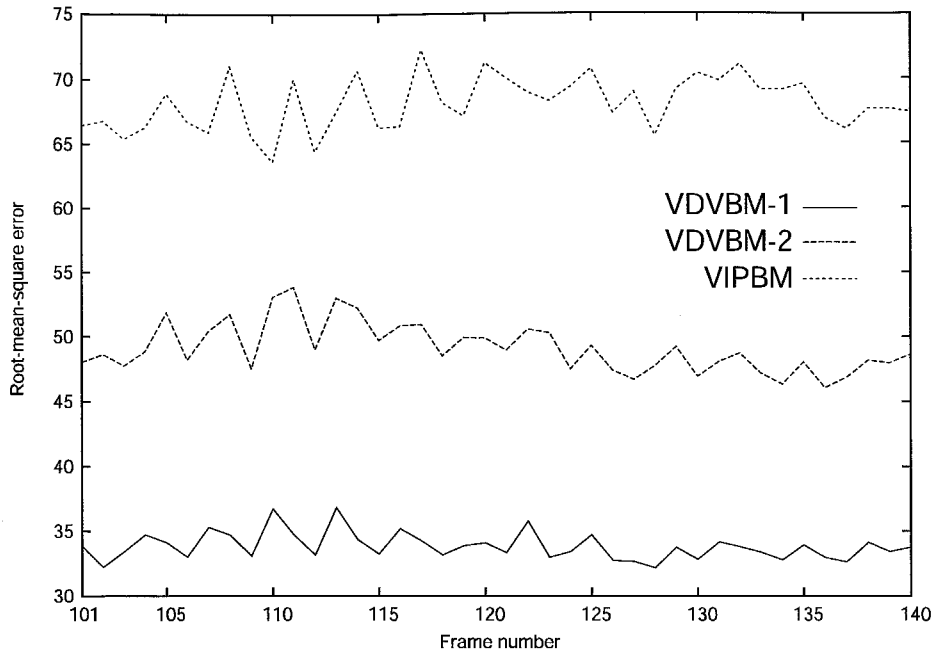


Figure 2.10: Root-mean-square error of RGB value (1) (Mesh)

of view.

Finally, we show examples generated by VDVBM-1 with subdivided patches (S6) viewed from camera 5, 11, and an intermediate point between them (Figure 2.17). Figure 2.17 shows that the images generated by VDVBM look almost real even when they are viewed from the intermediate point of the cameras.

For rendering 3D video data like Figure 2.17, we used a popular PC: (CPU: Xeon 2.2GHz, Memory: 1GB, Graphics Processor: GeForce 4 Ti 4600, Graphics library: DirectX 9.0b) and used the following two stage process:

1. First, compile a temporal sequence of reconstructed 3D shape data and multi-view video into a temporal sequence of vertex lists, where multi-view RGB values are associated with each vertex. It took about 2.8 sec to generate a vertex list for a frame of 3D video.
2. Then, with the vertex list sequence, arbitrary VGA views of the 3D video sequence can be rendered at 6.7 msec/frame. Thus, we can realize real-time interactive browsing of 3D video with a PC. Note also that since we can render a pair of stereo images in real-time (i.e. 14 msec/stereo-pair), we can enjoy eye-popping 3D image interactively with a 3D display monitor.

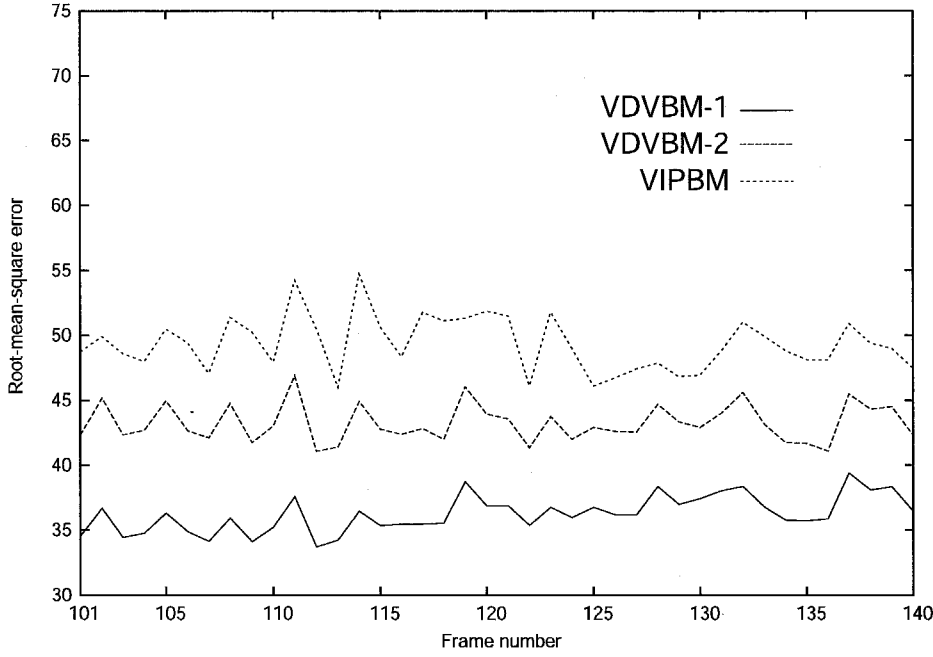


Figure 2.11: *Root-mean-square error of RGB value (1) (D-mesh)*

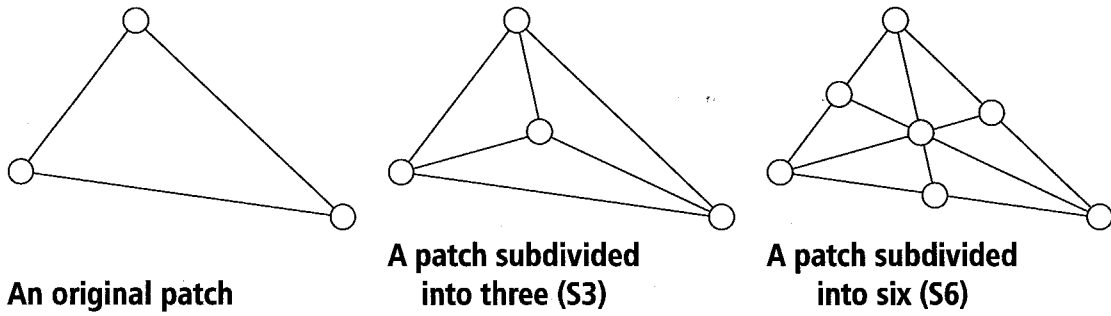


Figure 2.12: *Subdivision of a surface patch*

2.6 Discussions

We have presented an algorithm of rendering high fidelity texture on the reconstructed 3D object surface from multi-view video data, that is, the viewpoint dependent vertex-based method. Our method enables us to reduce jitters in rendered object images even though the limited accuracy of the reconstructed 3D object shape due to the volume intersection method or not completely synchronized multi-view video data.

Experimental results with *Mesb* data demonstrate the effectiveness of our method applying the data of limited accuracy, and results with D-mesh data show that our method can generate more natural looking object images from arbitrary viewpoints.

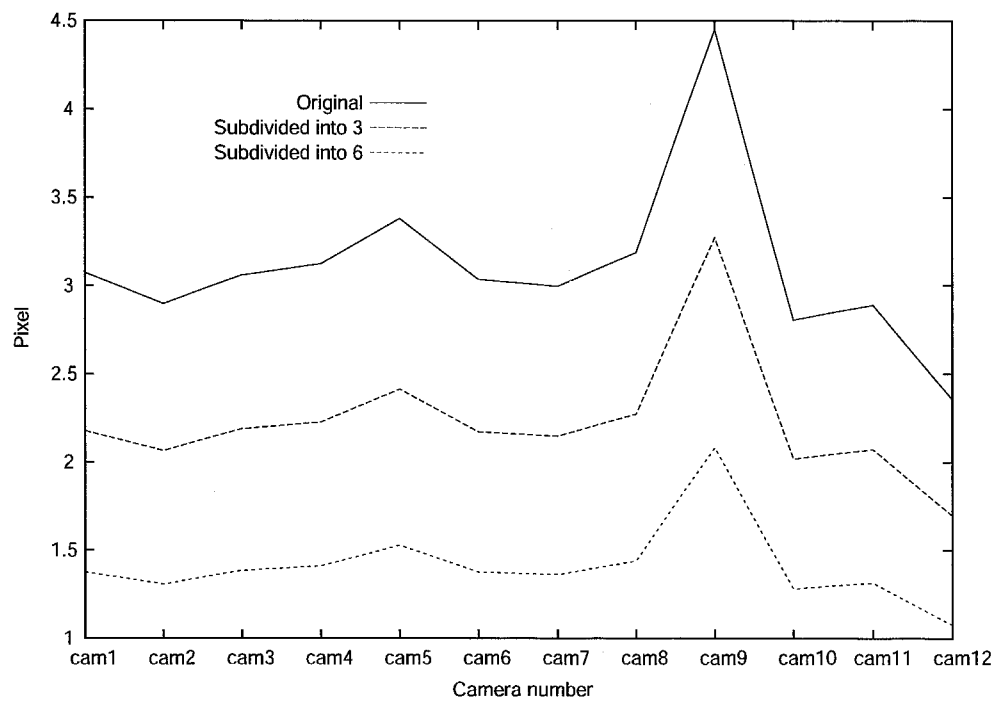


Figure 2.13: Mean side length in pixel on image planes of cameras

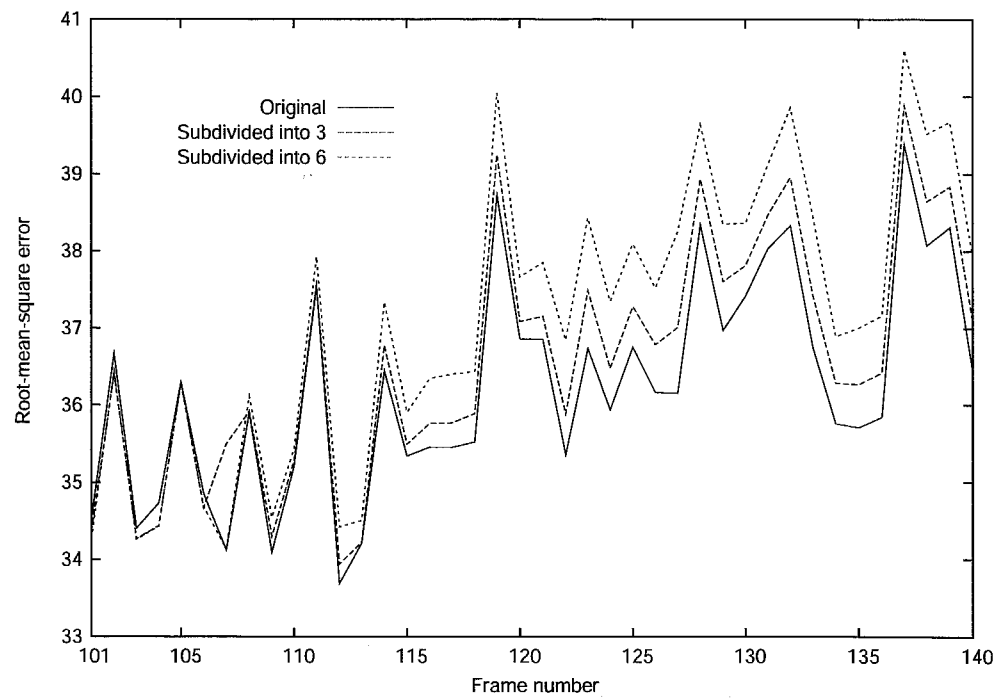


Figure 2.14: Root-mean-square errors of RGB value (2)



Figure 2.15: *Subtraction between a real image and a generated image (frame #106)*

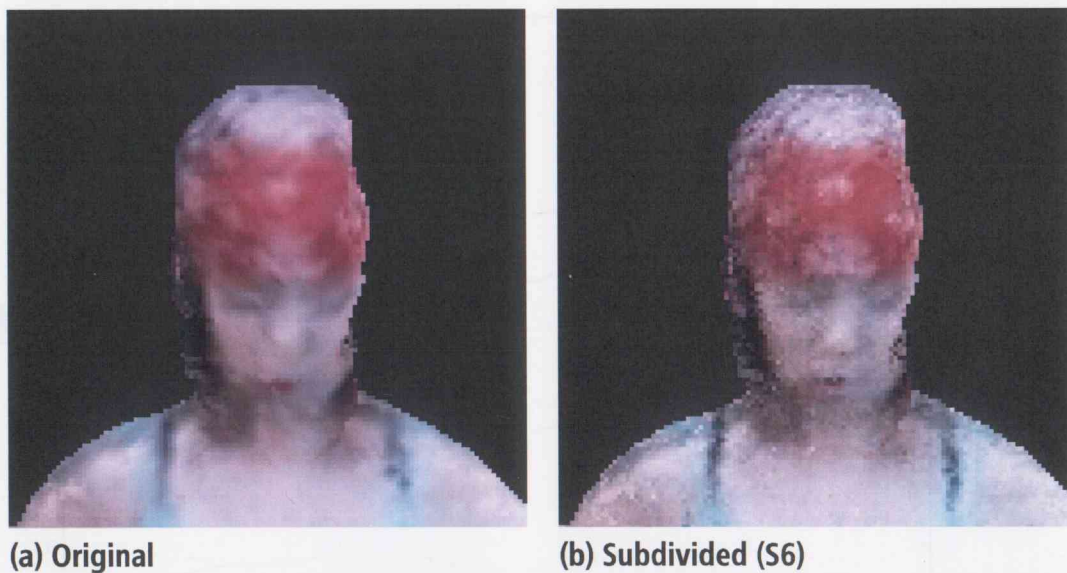


Figure 2.16: *Example images visualized with original and subdivided patches (frame #103)*

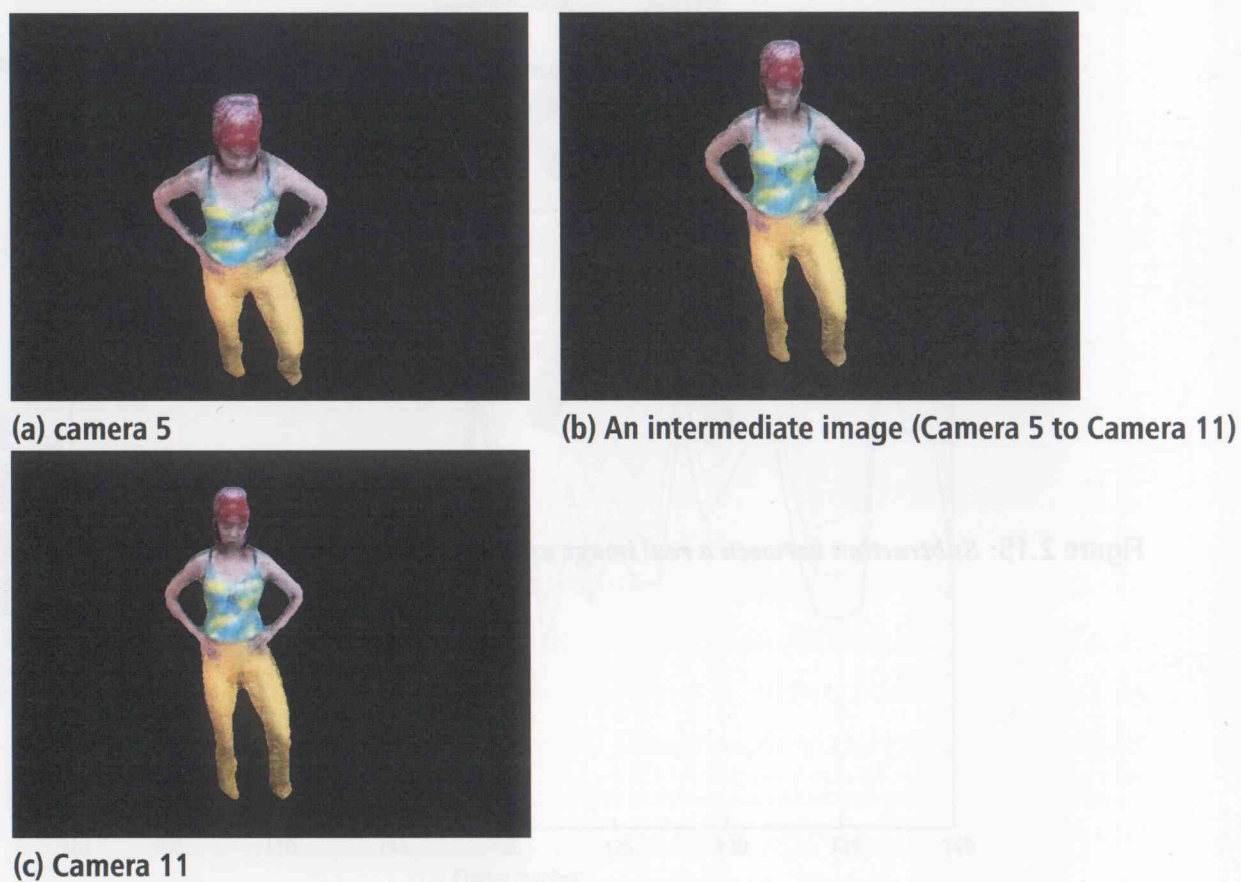


Figure 2.17: *Visualized 3D video with subdivided patches (frame#103)*

Chapter 3

Skeleton Cube for Lighting Environment Estimation

3.1 Introduction

In this chapter, we present a new method for lighting environment estimation with the Skeleton Cube (Figure 3.1). The Skeleton Cube is a small hollow cube, whose shadows are casted on itself—i.e. it has self-shadows. The key ideas of our method are 1) the estimation of lighting environment by analyzing self-shadows and shading of the Skeleton Cube, and 2) the representation of lighting environment by a large set of area light sources and point light sources.

While various approaches for lighting environment estimation have been proposed, the lighting environment is represented either as a set/distribution of distant light sources, or a few near point light sources. Unlike previous approaches, our goal is to develop a framework of inverse lighting for dealing with a number of light sources while taking their proximity and radiant intensity into account. The major advantage of using the skeleton cube is that the self-shadows effectively reflect the complication of lighting in real scene and thereby allows more stable lighting estimation than previous efforts. It should also be noted that the shape of the skeleton cube is quite rational in the sense that it can be used at the same time as an object for camera calibration of imaging geometry since it is a cube of a known size.

In the rest of the chapter, Section 3.2 presents modeling of lighting environment. Section 3.3 introduces the skeleton cube, and Section 3.4 describes in detail our lighting environment estimation. Section 3.5 shows results of verification of the skeleton cube, and

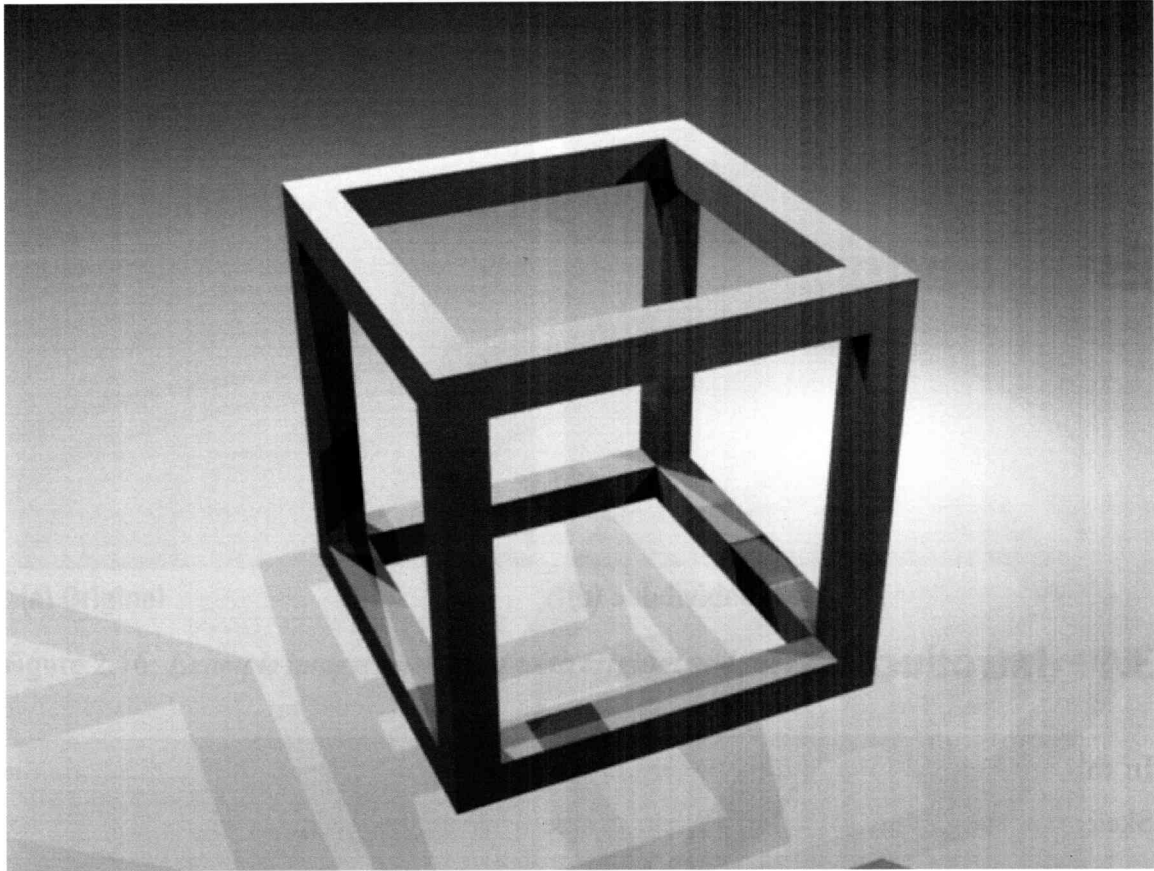


Figure 3.1: *The Skeleton Cube.*

Section 3.6 presents performance evaluations of our method. Finally, Section 3.7 conclude the chapter with discussions.

3.2 Lighting Environment Model

A lighting environment is represented by a distribution of light sources in a space, which have parameters of a type of a light source, position, direction, and radiant intensity. The distribution and the type of a light source are specified a priori, and therefore a lighting environment estimation is a task to estimate the radiant intensity of each light source in the environment. In this section, we first introduce a photometric model and a type of light sources that are utilized for representing the lighting environment, and then we describe an effective arrangement of the light sources.

3.2.1 Photometric Model

First of all, we give short descriptions of words in radiometry for describing a photometric model in the following [Oku04]:

- ◆ **Radiant flux:** Emitted energy of a light source per unit time with respect to wavelength λ . It is measured in units in watts [W], and written by $\Phi(\lambda)$.
- ◆ **Irradiance:** Radiant power incident per unit area upon a surface. It is measured in units of watts per square meter [W/m²]. Irradiance, $E(\lambda)$, is given by

$$E(\lambda) = \frac{d\Phi(\lambda)}{dS}. \quad (3.1)$$

- ◆ **Radiant intensity:** Radiant power per unit solid angle. It is measured in units of watts per steradian. [W/sr]. Radiant intensity, $L(\lambda)$, is given by

$$L(\lambda) = \frac{d\Phi(\lambda)}{d\omega}. \quad (3.2)$$

- ◆ **Radiance:** Reflected energy from a surface. It is measured in units of watts per steradian per square meter [W/(sr · m²)]. When we observe an object, we sense the radiance of a surface on the object. Radiance, $I(\lambda)$, is given by

$$I(\lambda) = \frac{dL(\lambda)}{dS \cos \delta} = \frac{d^2\Phi(\lambda)}{d\omega dS \cos \delta'} \quad (3.3)$$

where δ denotes the angle between the viewing direction and the surface orientation.

3.2.2 Types of Light Sources

We utilize two types of light sources for representing a lighting environment, one is a point light source and the other is an area light source.

These light sources are assumed as *near light sources*—i.e. they have an effect depending on their positions and attenuation by distance between a light source and a surface point of an object. Since we demand to estimate a lighting environment in a small space such as studios instead of open spaces, near light sources are highly required to represent such lighting environments.

We utilize area light sources as well as point light sources for representing a lighting environment. While an area light source enables us to realize rich expressions in shading

38 Chapter 3. Skeleton Cube for Lighting Environment Estimation

and shadowing, it is hardly utilized for a lighting environment estimation. The main reason is that it is difficult to model an area light source solvable. In this chapter, we present a practical method to model an area light source and estimate its parameters as well as a point light source.

We here summarize the two types of light sources in the following:

Point light source (Figure 3.2): It is a light source that emits radiant flux to all direction from an infinitesimal point. Figure 3.2 shows a relationship between radiant intensity of a point light source and irradiance of minute surface dS . Given radiant intensity of a point light source that illuminates the surface from distance r , the irradiance is given by

$$E(\lambda) = \frac{d\Phi(\lambda)}{dS} = \frac{d\omega}{dS} L(\lambda) = \frac{\cos \theta}{r^2} L(\lambda), \quad (3.4)$$

where we utilize the following equation of solid angle:

$$d\omega = \frac{dS \cos \theta}{r^2},$$

where θ denotes an angle between the surface orientation of dS and the direction from dS to the point light source.

Equation (3.4) indicates that the irradiance attenuates depending on the distance between the position of a point light source and the surface point of an object.

Area light source (Figure 3.3): It is a model of a light source that emits radiant flux to all direction above the surface. To simplify the model, we represent the distance between an area light source and a surface point of an object by the distance between the centroid of the light source and the surface point of the object. We assume that the area light source has uniform radiant intensity over the area. Figure 3.3 shows the relationship between the radiant intensity of an area light source and the irradiance of minute surface dS . Given an area light source that has an area of S_L and distance r , the irradiance is given by

$$E(\lambda) = \frac{d\Phi(\lambda)}{dS} = \frac{\cos \theta}{r^2} \int_{S_L} L'(x, \lambda) dx, \quad (3.5)$$

where $L'(x, \lambda)$ denotes radiant intensity per unit area of the area light source at surface point x . θ denotes an angle between the surface orientation of dS and the direction

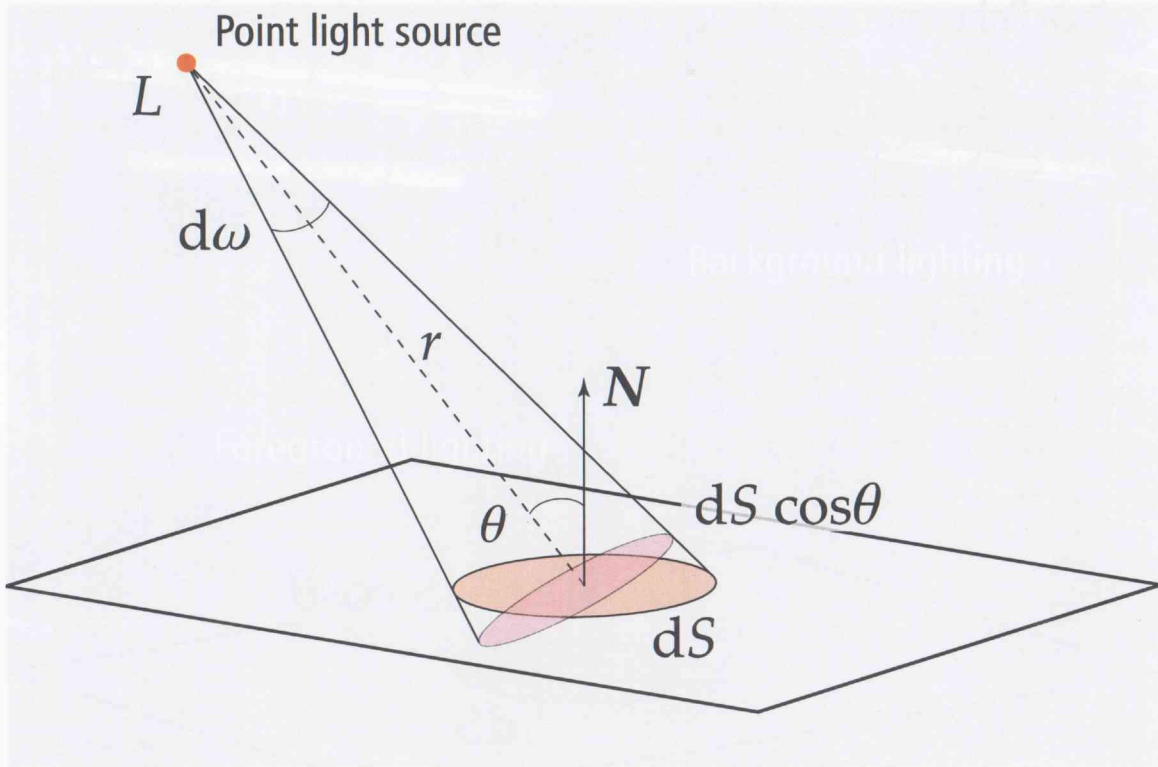


Figure 3.2: *A point light source and an illuminated surface*

from dS to the area light source. Assuming the radiant intensity is uniform over the area, we can obtain the following formula:

$$\int_{S_L} L'(x, \lambda) dx = L'(\lambda) S_L \cos \varphi = L(\lambda) \cos \varphi, \quad (3.6)$$

where φ denotes the angle between the orientation of the area light source and the direction from the area light source to ds . We thus obtain the irradiance by

$$E(\lambda) = \frac{\cos \theta}{r^2} L(\lambda) \cos \varphi. \quad (3.7)$$

3.2.3 Arrangement of Light Sources

To represent a lighting environment, we arbitrarily arrange a large number of light sources in a space. It enables us to represent variable lighting environment, but also increases the computational costs. We therefore have to consider to represent the lighting environment with a minimum number of light sources. Furthermore, we have to take reflected lighting from illuminated wall or floor into account as well as light sources such as lamps, fluorescent

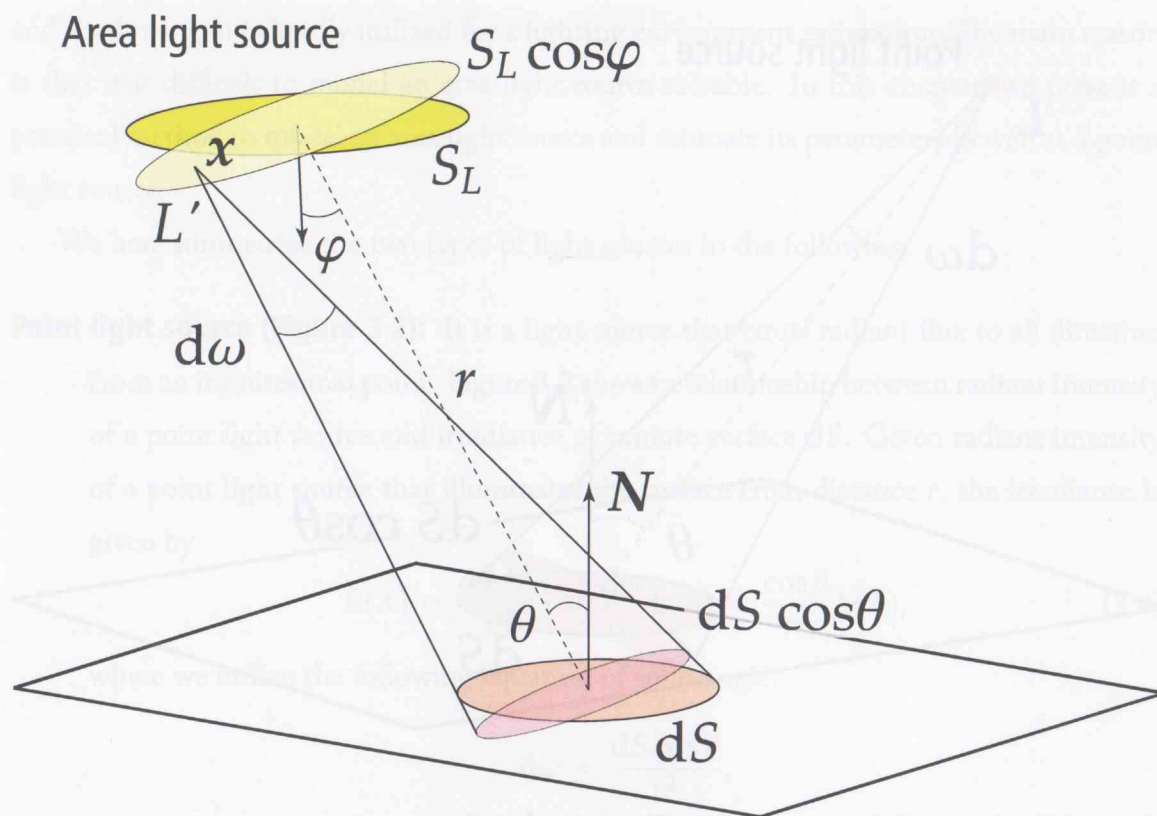


Figure 3.3: *An area light source and illuminated surface*

tubes, etc., because we mainly consider a lighting environment in an indoor space such as studios, rooms, etc. where we cannot ignore the effects of reflection.

Accordingly, we separate a lighting environment into two components: structure of a studio (walls, a ceiling, and a floor)—i.e. **background lighting**, and actual light sources (lamps, spot lights, etc.)—i.e. **foreground lighting**. By utilizing background and foreground lighting, we can decrease the number of light sources without losing arbitrariness of the representation. In the following section, we describe the details of these lighting.

3.2.3.1 Background Lighting

We assume that background lighting surrounds a space in which we capture images of an object. To represent the background lighting for a studio that is surrounded by walls, we can simply arrange area light sources in such a way that roughly model the geometry of the studio. We show an example of the background lighting in Figure 3.4. While the level of details varies according to required accuracy, in most cases, a cube is sufficient to represent

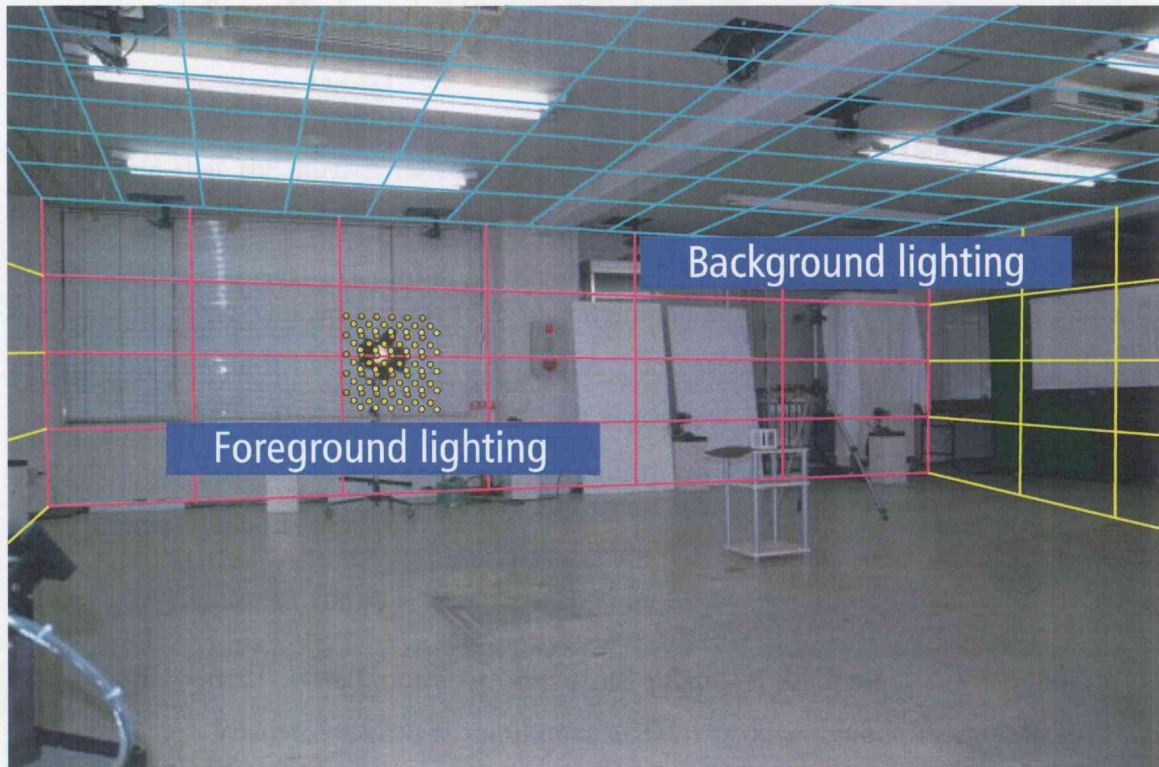


Figure 3.4: *An example of the background lighting and the foreground lighting*

The background lighting is represented by a distribution of area light sources. Each rectangle indicates an area light source, and resolution of the background lighting is arbitrarily specified. The foreground lighting is represented by a distribution of point light sources that represents the actual light source.

the shape. Furthermore, given fluorescent light tubes on the ceiling, we can represent the background lighting as a set of area light sources including the fluorescent light tubes.

Hence, we can represent the background lighting arbitrary with area light sources. In addition, for constructing a lighting environment of an open space, large dome-shaped background lighting is suitable.

3.2.3.2 Foreground Lighting

We define that foreground lighting consists of ordinary light sources such as spot lights, lamps, etc., and represent it by a distribution of point light sources. We need to know not precise positions of the light sources but possible spaces where the light sources can exist. By arranging point light sources in the space and estimating radiant intensity of all of them, we can represent an actual light source by a set of the point light sources. We show an example

of distribution of point light sources for the foreground lighting in Figure 3.4.

3.3 Skeleton Cube

3.3.1 Geometric Features

Our basic strategy is to estimate the parameters of light sources by analyzing self-shadows as well as the shading on the surface of a reference object. Given that we can utilize a reference object of known shape, we design the skeleton cube, a hollow cube which casts shadows to its inner surface, as shown in Figure 3.1. The design is on the basis of the two requirements that are inconsistent to each other. Namely, the shape should be simple while some complexity is desirable:

- ◆ **Simplicity:** Simple shape is suitable for the computational costs. Then, a large portion of the surface should be observable in sampling the surface intensity.
- ◆ **Complexity:** The shape needs to be complex to some extent so that cast shadows occur under variable lighting conditions.

We came up with the skeleton cube as an object that satisfies above requirements. That is, under light sources at almost any positions in a scene, it cast shadows on its inside surfaces and the shadows can be observed from any viewpoints. In rare settings certain light sources might not be probeable with the skeleton cube depending on the configurations of the light sources and the viewing direction. It is in such a case that the self-shadow on the inner surface of the skeleton cube that is casted by the light sources happens to be occluded¹. However, without losing the practicality of our estimation, we could take pictures from more than one different viewpoints, from which we can see all part of the surfaces of the skeleton cube in order to handle the lighting estimation even in such situation.

The skeleton cube can also be used for a reference object for geometric calibration of cameras, which is an ordinary method by matching corresponding points in a captured image and the model although we do not go into the details of the geometric calibration.

¹There may be also an argument that a light source inside the skeleton cube is unprobeable, but we simply ignore the case since a light source should not be inside unless we deliberately place the cube and a light source in such an exceptional way.

3.3.2 Photometric Features

3.3.2.1 Reflectance Model

We assume that the surface reflection of the reference object is described by simplified Torrance-Sparrow model [TS67, IS91] which can represent both diffuse and specular reflection, and it especially describes the specular reflection with physical properties. It is in this sense more general than other models such as Phong reflectance model [Pho75]. In this thesis, we consider that the influence of inter-reflection is ignorable and concentrate our discussion on the first reflection.

See Figure 3.5 that shows the geometry of reflection, including the viewpoint, V , and the location of light source, $\mathcal{L}$. With the simplified Torrance-Sparrow model, observed radiance at minute surface $\mathbf{x}$, $I(\mathbf{x})$, is described as

$$I(\mathbf{x}) = (k_d R_d + k_s R_s) L_{\mathcal{L}}, \quad (3.8)$$

where L is the radiant intensity of light source $\mathcal{L}$. R_d and R_s denote the diffuse and specular component of a bidirectional reflectance distribution function, respectively, whereas k_d and k_s are the weighting coefficients of them. Though $I(\mathbf{x})$ and L are functions with respect to wavelength, we utilize the functions at three wavelengths for red, green and blue. For the sake of a simple description, we do not denote it explicitly in the following description.

Assuming the diffuse component is represented as Lambertian—i.e. the radiance of the diffuse component is proportional to the irradiance of the surface that is given by a light source, we have $R_d = N \cdot L / r^2$, where r denotes the distance between light source L and point $\mathbf{x}$. The specular component, R_s , is represented as

$$R_s = \frac{1}{N \cdot V} \exp \left[-\frac{(\cos^{-1}(N \cdot H))^2}{2\sigma^2} \right], \quad (3.9)$$

where σ denotes the surface roughness.

3.3.2.2 Radiance of Skeleton Cube

The radiance of point $\mathbf{x}$ on the skeleton cube in a lighting environment can be formulated as

$$I(\mathbf{x}) = \sum_{i=1}^N \mathcal{M}(\mathbf{x}, \mathcal{L}_i) (k_d R_d + k_s R_s) L_{\mathcal{L}_i}, \quad (3.10)$$

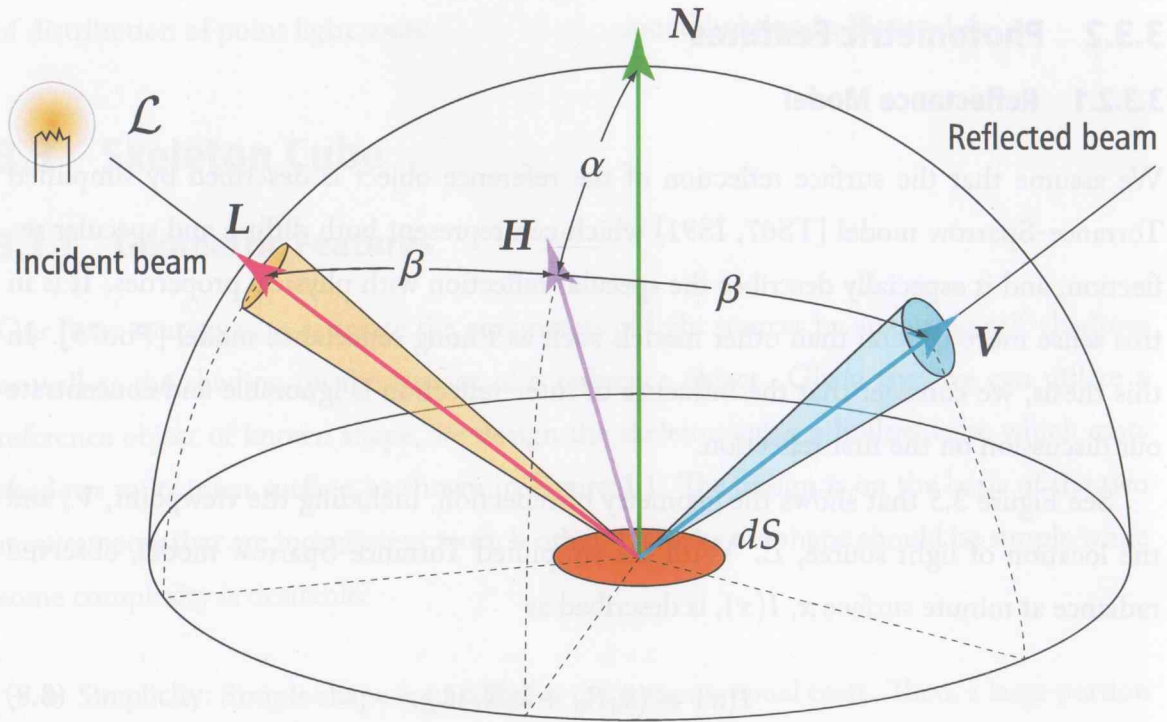


Figure 3.5: The geometry of reflection

V is unit vector in direction of the viewer, L is unit vector in direction of the light source, N is unit surface normal, and H is unit angular bisector of V and L .

where N is the number of point light sources, and $\mathcal{M}(x, \mathcal{L}_i)$ is a mask term that encodes the self-shadow of the skeleton cube. The mask term is determined by the type of light sources, which are described as following.

Point light source: It indicates whether point x is illuminated by light source $\mathcal{L}_i$ or not—i.e. $\mathcal{M}(x, \mathcal{L}_i) = 1$ if light source $\mathcal{L}_i$ illuminates point x and $\mathcal{M}(x, \mathcal{L}_i) = 0$ otherwise.

Area light source: It indicates the rate of occlusion of the light source—i.e. $\mathcal{M}(x, \mathcal{L}_i) = 1$ if the whole area of light source $\mathcal{L}_i$ illuminates point x , and $\mathcal{M}(x, \mathcal{L}_i) = 0$ if the area light source dose not illuminates point x at all. We describe the procedure of calculating the mask term in the following:

step 1: Render an image of area light source $\mathcal{L}_i$ viewed from point x assuming a situation *without* the skeleton cube, and count the size of area light source $\mathcal{L}_i$ in the image (Figure 3.6a). We denote the size as S .

step 2: Render an image of area light source $\mathcal{L}_i$ viewed from point x according to the practical situation *with* the skeleton cube, and count the size of area light source $\mathcal{L}_i$ in the image (Figure 3.6b). We denote the size as S' .

step 3: Calculate the mask term by

$$\mathcal{M}(\mathcal{L}_i, x) = S' / S. \quad (3.11)$$

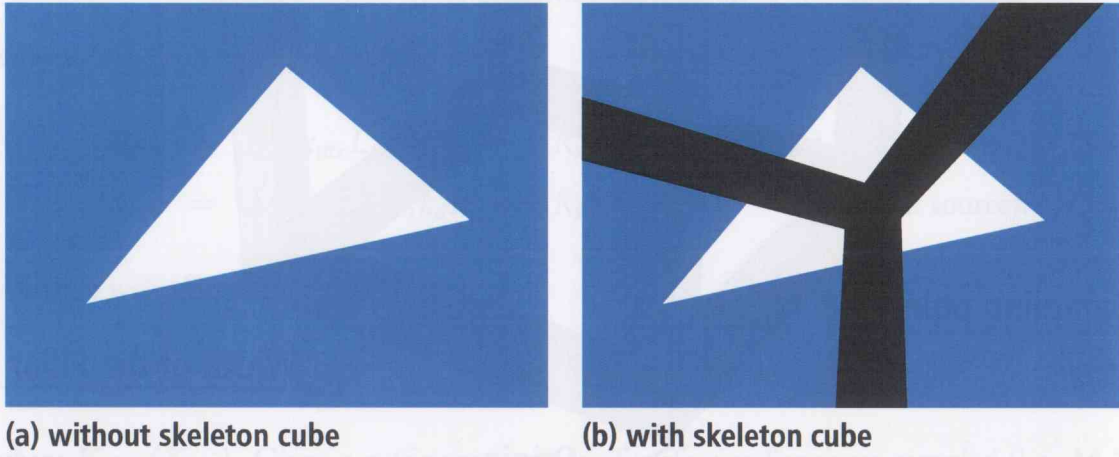


Figure 3.6: *Appearance of an area light source viewing from a point on an inner surface of the skeleton cube*

The white triangle and the black polygon represent the area light source and the skeleton cube, respectively, from which are viewed from a point on an inner surface of the skeleton cube.

3.4 Lighting Environment Estimation

In this section, we formulate the estimation of lighting environment preceded by the definition of the lighting environment and the description of our setup.

3.4.1 Sensing System

Figure 3.7 shows the setup regarding the skeleton cube. Observing the radiance at the sample points that are on the inner surface, we estimate the lighting environment based on the following premises.

Coordinate System: We define the origin of the world coordinate system at the center of the skeleton cube in such a way that each axis of it is aligned parallel to the side of the

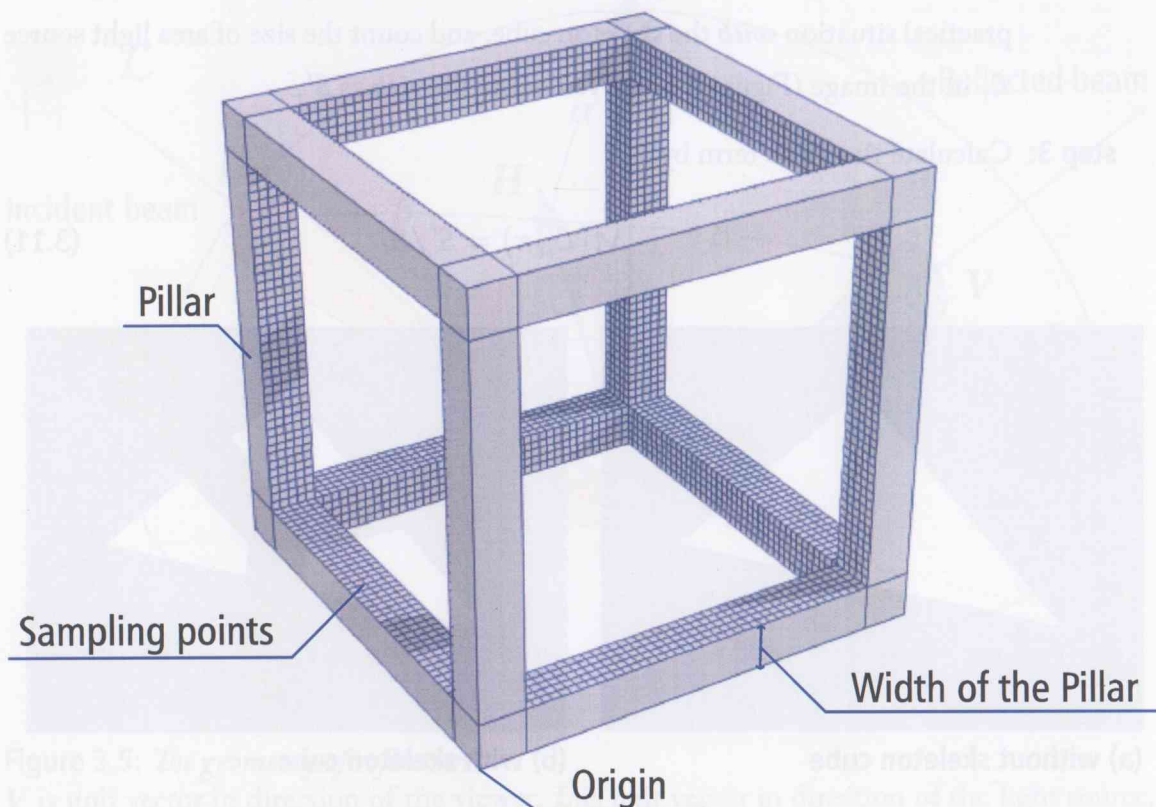


Figure 3.7: *The geometry of the skeleton cube*

Sampling points are at the lattice positions on the inner surface of the cube.

skeleton cube. We locate the lighting environment and cameras that capture images in the world coordinate system.

Lighting environment: The distribution range of the light sources is assumed a priori apart from what is described in Section 3.2.

Skeleton cube: The size is known. The properties of the surface reflection is also available by the estimation that is described in Section 3.6.2.1.

Camera: The cameras that capture images of the scene including the skeleton cube are calibrated with the skeleton cube so that we know the relationship of corresponding points in the scene and in the image.

3.4.2 Computational Algorithm

When we sample M points on the inner surface of the skeleton cube for observing the radiance, we have a matrix representation based on Equation (3.10)

$$\begin{bmatrix} I(x_1) \\ I(x_2) \\ \vdots \\ I(x_M) \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1N} \\ K_{21} & K_{22} & \cdots & K_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ K_{M1} & K_{M2} & \cdots & K_{MN} \end{bmatrix} \begin{bmatrix} L_{\mathcal{L}_1} \\ L_{\mathcal{L}_2} \\ \vdots \\ L_{\mathcal{L}_N} \end{bmatrix}, \quad (3.12)$$

where

$$K_{mn} = \mathcal{M}(x_m, L_n) (k_d R_d + k_s R_s) \cdots \cdots \cdots (\text{Point light source}), \quad (3.13)$$

$$K_{mn} = \mathcal{M}(x_m, L_n) (k_d R_d + k_s R_s) \cos \varphi \cdots \cdots \cdots (\text{Area light source}). \quad (3.14)$$

We then write Equation (3.12) simply as

$$I = KL, \quad (3.15)$$

where $K = (K_{mn})$. Given a sufficient number of surface radiance are sampled (i.e. $M \gg N$), it is possible to solve Equation (3.15) for L by

$$L = K^+ I, \quad (3.16)$$

where K^+ denotes a pseudo-inverse matrix of K . Namely, we can solve for the radiant intensities in vector $[L_{\mathcal{L}_1}, L_{\mathcal{L}_2}, \dots, L_{\mathcal{L}_N}]^T$.

3.5 Verifying Utilities of Skeleton Cube

We verify the skeleton cube as a reference object for lighting environment estimation. In Section 3.5.1 we investigate the effectiveness of our lighting environment estimation using self-shadows by analyzing matrix K that constitutes Equation (3.12). In Section 3.5.2 we examine the occurrence of self-shadows depending on the positions of point light sources.

3.5.1 Evaluating Effectiveness of Lighting Environment Estimation with Self-shadows

Because lighting environment estimation using Lambertian shading on an object is an ill-posed or numerically ill-conditioned problem, in our inverse rendering, we take specular

reflection into account on one hand, and on the other hand expect the non-linear factor by self-shadows, $\mathcal{M}$ term in Equation (3.13) and Equation (3.14), to help making the problem better-conditioned. Here, we study the relevancy of our lighting environment estimation with self-shadows by comparing matrix $\mathbf{K}$ in Equation (3.13) and Equation (3.14), for the cases with and without $\mathcal{M}$ term. We generate matrix $\mathbf{K}$ in respective cases by the following conditions,

◆ **Lighting environment:**

- ❖ **Point light source:** A set of point light sources on 3D grid points ranging from $(-1500, -1000, 0)$ to $(1500, 1000, 2500)$ with spacing of 500 as shown in Figure 3.8a. The number of the light sources is 209.
- ❖ **Area light source:** A set of area light sources on the ceiling and the walls ranging from $(-1500, -1000, 0)$ to $(1500, 1000, 2500)$ as shown in Figure 3.8b. The number of the light sources is 200.

◆ **Camera position:** (450, 450, 450).

◆ **Skeleton cube:**

- ❖ **Size:** Cube 100 on a side. The width of pillar 10.
- ❖ **Reflectance property:** $k_d = 1.0\text{E}+9$, $k_s = 0.8\text{E}+2$, and $\sigma=0.3$.
- ❖ **Number of the sampling points from a viewpoint:** 2293.

We visualize the generated matrix $\mathbf{K}$ in the case of the point/area light source and with/without $\mathcal{M}$ term (Figure 3.9 and Figure 3.10). The rows of matrix $\mathbf{K}$ is sorted according to the categories of surface normals of the skeleton cube while the row and the column correspond to sampling points and light sources, respectively. The color denotes the value of entries, i.e., red for low, blue for high, and white for 0 values. Moreover, we apply the singular value decomposition to matrix $\mathbf{K}$ in the case of the point/area light source and with/without $\mathcal{M}$ term, in order to verify the rank of the matrix by calculating the contributions of each singular value to the entire sum (Figure 3.11 and Figure 3.12). We discuss the results of visualized matrix $\mathbf{K}$ and contributions of singular values in the following.

◆ **Visualized matrix $\mathbf{K}$ (Figure 3.9 and Figure 3.10)**

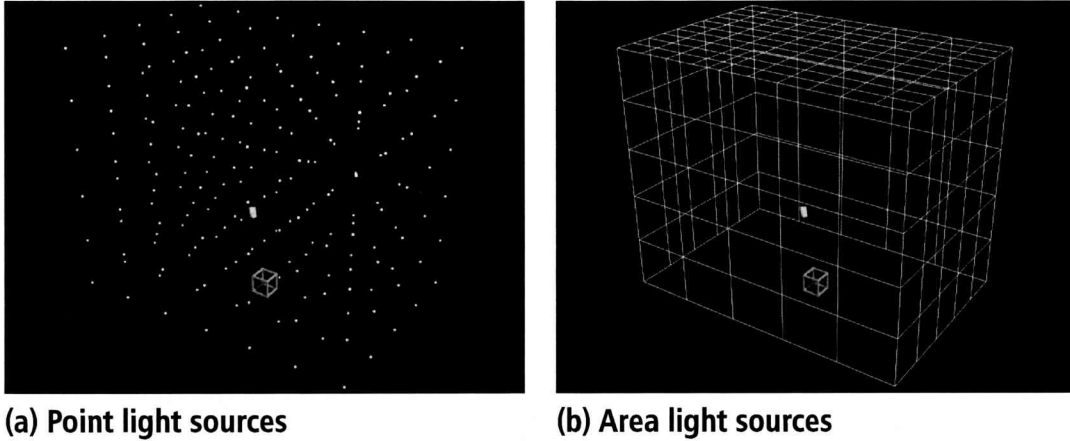


Figure 3.8: *Distribution of light sources*

The points and the rectangles in the images illustrate point and area light sources, respectively. The skeleton cube and the camera are arranged in each lighting environment.

- ❖ **Effect of $\mathcal{M}$ term:** With $\mathcal{M}$ term as a consequence of self-shadows, matrix K contains variations of the rows as observed in Figure 3.9a and Figure 3.10a since different light sources are visible from different sample points even when they have an identical surface normal. Without $\mathcal{M}$ term, on the other hand, variations of the rows are roughly in three categories, and it illustrates that the rank of matrix K without $\mathcal{M}$ term is degenerated close to rank three. This is because the number of the sorts of surface normals that are viewable from a single viewpoint is at most three, in other words, the rows within each categories are quite similar except for the difference due to specularities or the distance to the light sources, which causes rank deficiency. We later discuss the rank of matrix K in details.
- ❖ **Difference between the types of light sources:** Matrix K derived from the area light sources shows more color variations than that from the point light sources because of the effects thanks to the continuous value of $\mathcal{M}$ term and $\cos \phi$ in Equation (3.14). Although there are differences in the color variations or appearance of white specks due to the $\mathcal{M}$ term, the visualized matrices of both types of the light sources show similar tendencies.
- ◆ **Contributions of singular values of matrix K (Figure 3.11 and Figure 3.12)**
 - ❖ **Effect of $\mathcal{M}$ term:** We compare matrix K in the two cases (with or without $\mathcal{M}$ term) by analyzing their rank in terms of the contributions of the major singular

values. In order to solve Equation (3.16) stably, a certain rank of matrix K is required. As shown in Figure 3.11 and Figure 3.12, however, matrix K without $\mathcal{M}$ term degenerates close to rank three, which is observable in contributions of the singular values that falls down drastically after the third singular value. With $\mathcal{M}$ term, singular values show relatively higher contributions after the third singular value, and it proves that matrix K with $\mathcal{M}$ term has a higher rank, which is more desirable for solving Equation (3.16).

- ❖ **Difference between the types of light sources:** The major difference between the point light source and the area light source is that the contribution of the first singular value in the case of matrix K of area light sources (see Figure 3.12) indicates a higher value than that of point light sources (see Figure 3.11). That is, the rank of the matrix derived from area light sources is lower than that from point light sources. Nevertheless, the contributions after the third singular value of matrix K with $\mathcal{M}$ term keep higher values, and therefore, it illustrates that an estimation using self-shadows is also effective for the area light sources, as well as the point light source.

Overall we have seen that the self-shadows effectively reflect the complication of the lighting environment, and as the result it is beneficial for computing the radiant intensities of the light sources with higher stability.

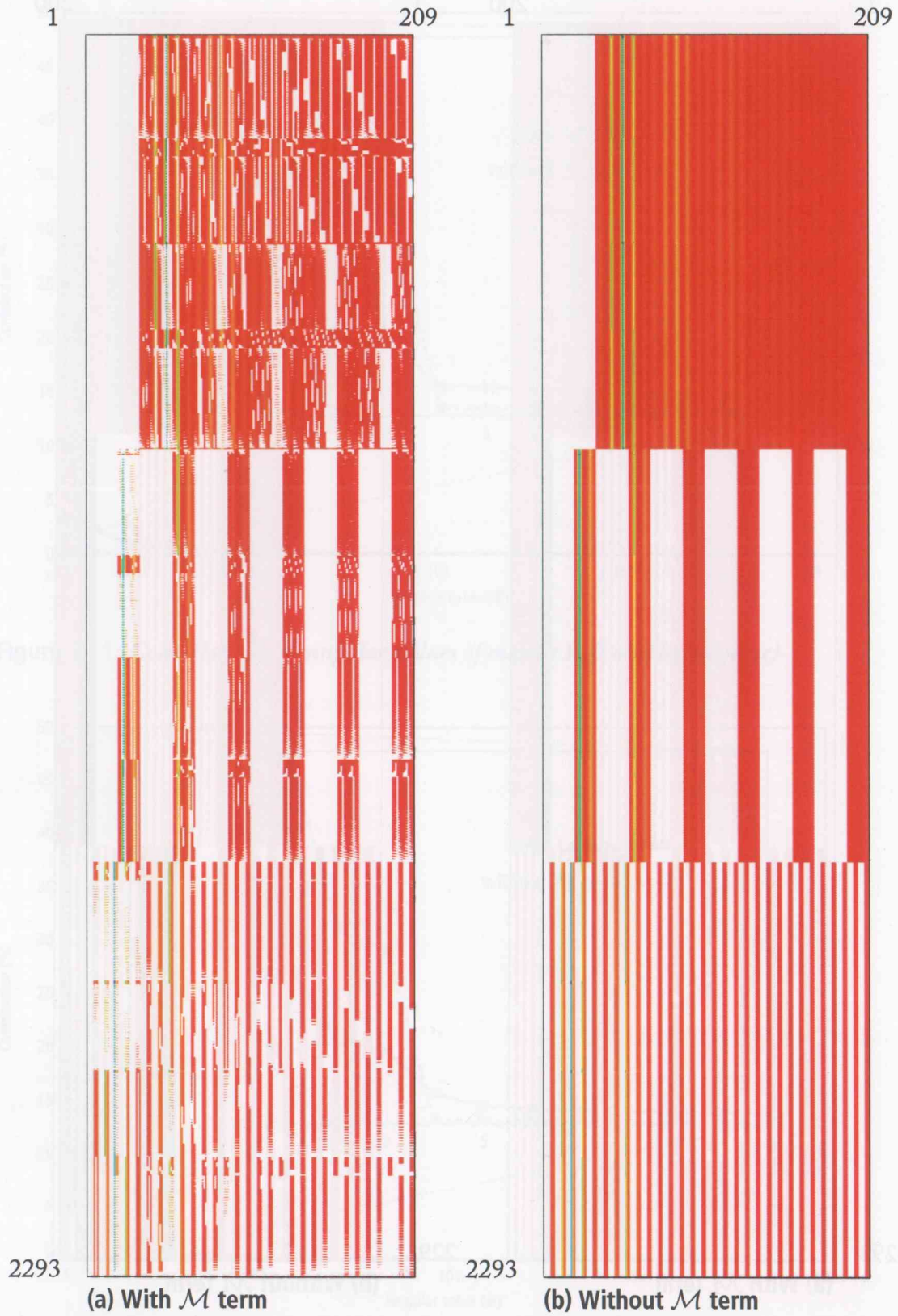


Figure 3.9: Visualized matrix K (Point light source)

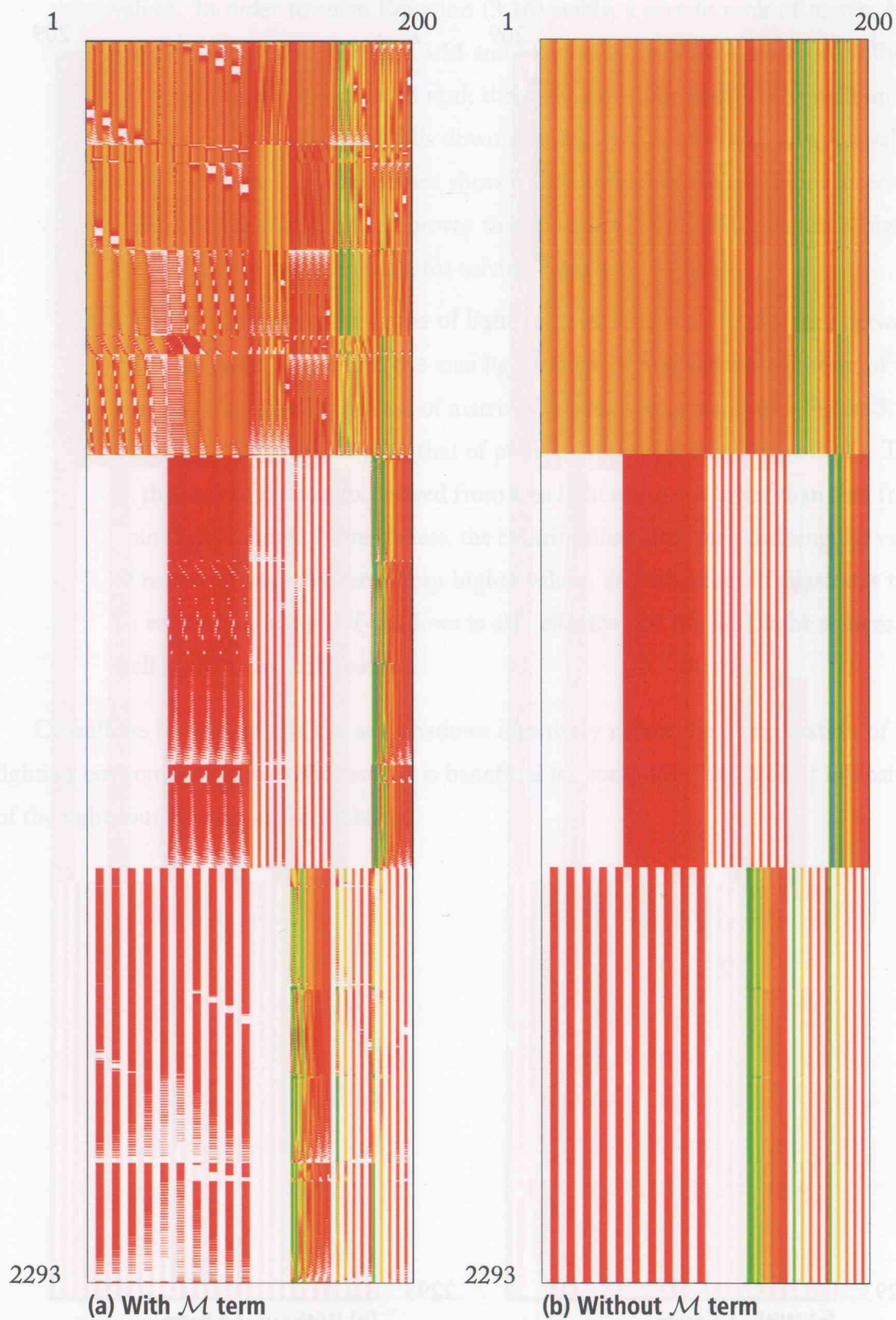


Figure 3.10: Visualized matrix K (Area light source)

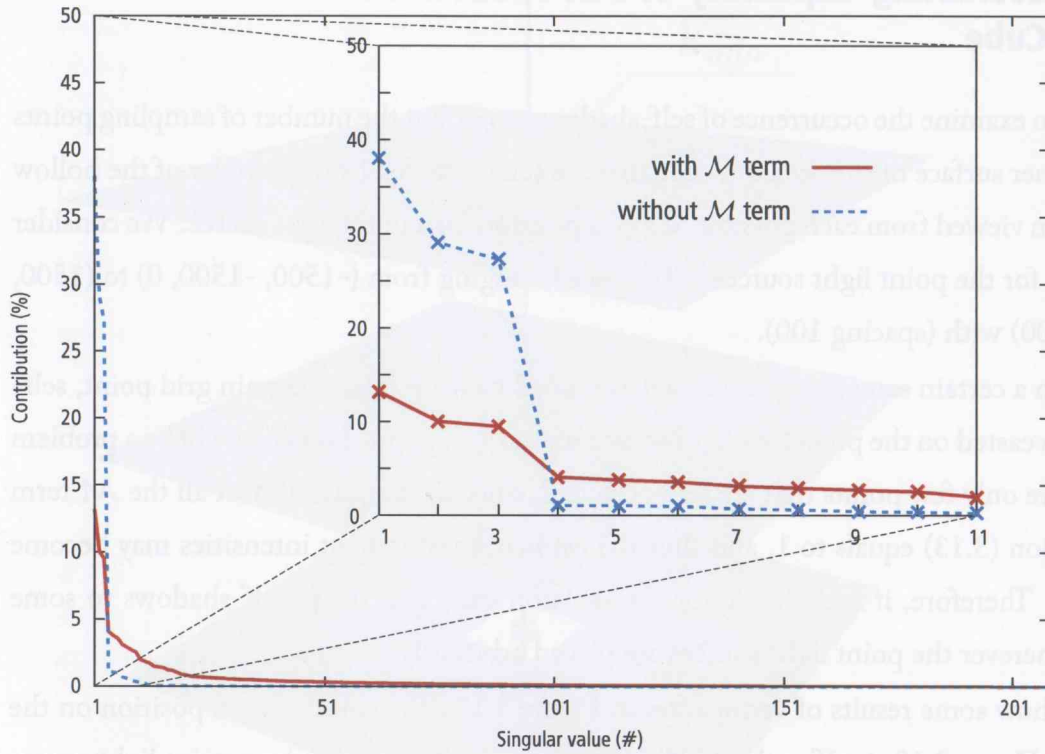


Figure 3.11: Contributions of singular values of matrix K (Point light source)

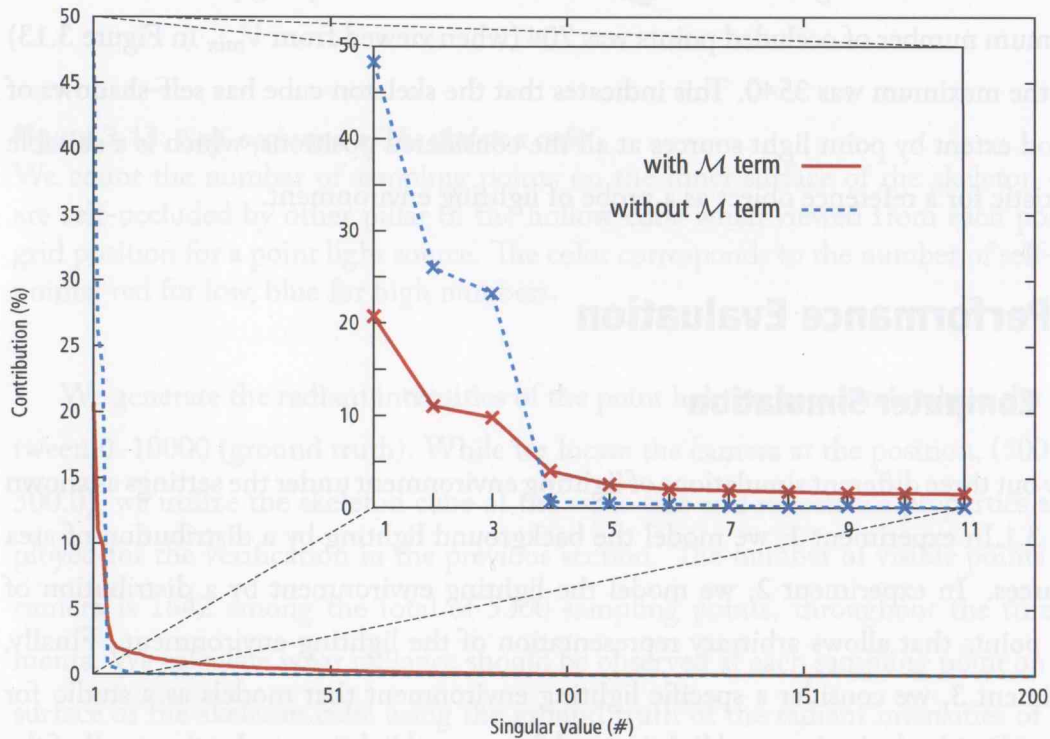


Figure 3.12: Contributions of singular values of matrix K (Area light source)

3.5.2 Evaluating Capability of Self-Shadow Generation of Skeleton Cube

In order to examine the occurrence of self-shadows, we count the number of sampling points on the inner surface of the skeleton cube that are self-occluded by other pillar of the hollow cube when viewed from each possible 3D grid position for a point light source. We consider the space for the point light sources to be placed, ranging from $(-1500, -1500, 0)$ to $(1500, 1500, 3000)$ with (spacing 100).

When a certain sampling point is self-occluded viewing from a certain grid point, self-shadow is casted on the point by a light source on the grid point. Hence, it will be a problem if there are only few points that are self-occluded, since in that case almost all the $\mathcal{M}$ term in Equation (3.13) equals to 1, and thus the estimation of radiant intensities may become unstable. Therefore, it is desirable for the skeleton cube that it has self-shadows to some extent wherever the point light sources are placed arbitrarily.

We show some results of verification in Figure 3.13. The color of each position on the planes in Figure 3.13 signifies the possible number of self-occlusion by a point light source that is placed on that position of the place. The color denotes the number, i.e. red denotes a low, and blue denotes a high number. For the total number of sampling points, 5300, even the minimum number of occluded points was 704 (when viewed from V_{min} in Figure 3.13) whereas the maximum was 3540. This indicates that the skeleton cube has self-shadows of some good extent by point light sources at all the considered positions, which is a suitable characteristic for a reference object as a probe of lighting environment.

3.6 Performance Evaluation

3.6.1 Computer Simulation

We carry out three different simulations of lighting environment under the settings as shown in Table 3.1. In experiment 1, we model the background lighting by a distribution of area light sources. In experiment 2, we model the lighting environment by a distribution of 3D grid points that allows arbitrary representation of the lighting environment. Finally, in experiment 3, we consider a specific lighting environment that models as a studio for capturing 3D video by background lighting and foreground lighting, and estimate all of the light sources that form the lighting environment.

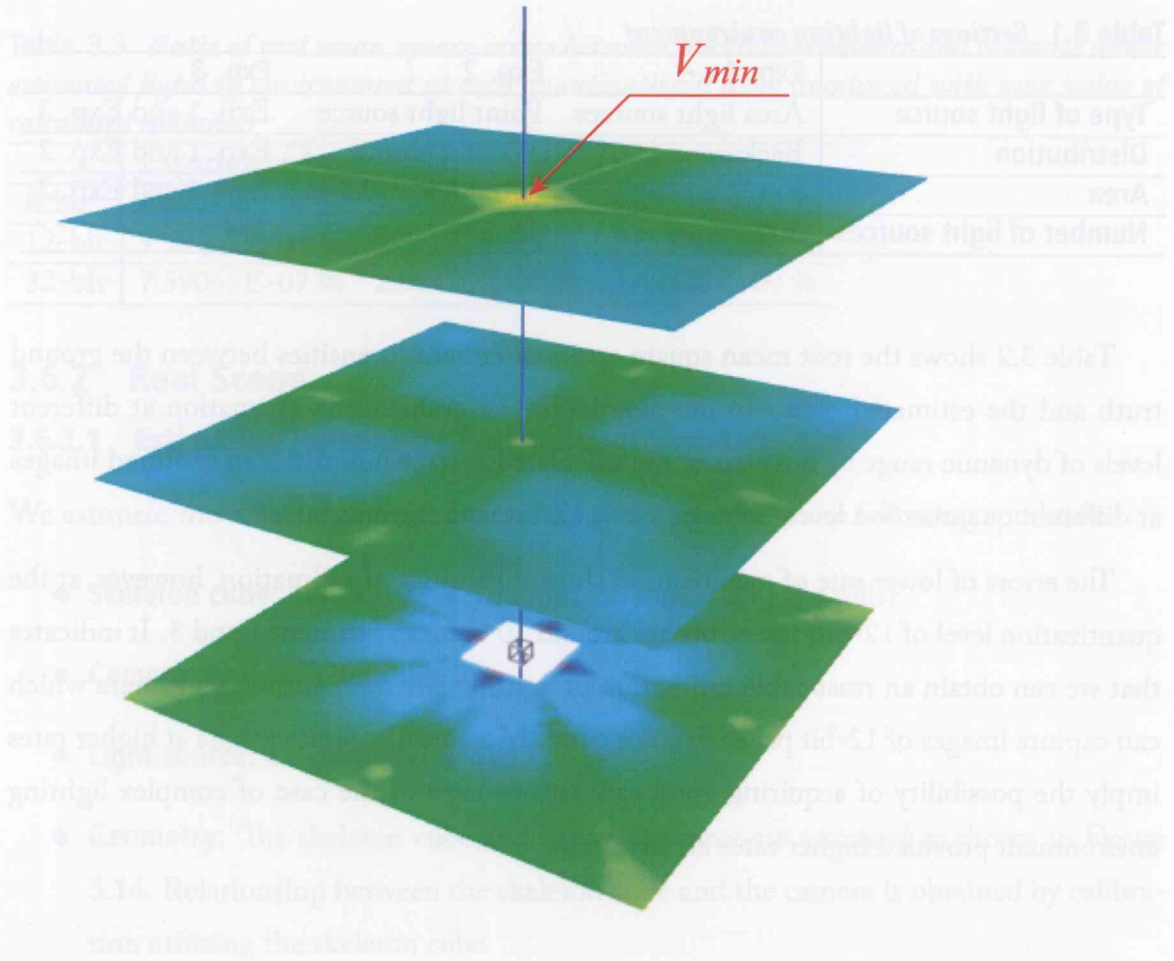


Figure 3.13: Self-occlusion of the skeleton cube

We count the number of sampling points on the inner surface of the skeleton cube that are self-occluded by other pillar of the hollow cube when viewed from each possible 3D grid position for a point light source. The color corresponds to the number of self-occluded points: red for low, blue for high numbers.

We generate the radiant intensities of the point light sources randomly in the range between 0–10000 (ground truth). While we locate the camera at the position, (500.0, 500.0, 500.0), we utilize the skeleton cube of the same size and reflectance properties as we employed for the verification in the previous section. The number of visible points from the camera is 1642 among the total of 5300 sampling points, throughout the three experiments. We calculate what radiance should be observed at each sampling point on the inner surface of the skeleton cube using the ground truth of the radiant intensities of the point light sources. Then, we inversely estimate the radiant intensity with the calculated surface radiance and investigate how close the estimation is to the ground truth.

Table 3.1: Settings of lighting environment

	Exp. 1	Exp. 2	Exp. 3
Type of light source	Area light sources	Point light source	Exp. 1 and Exp. 2
Distribution	Background	3D grid points	Exp. 1 and Exp. 2
Area	$8 \times 9 \times 2.5$ m	$0.5 \times 0.5 \times 0.5$ m	Exp. 1 and Exp. 2
Number of light sources	300	64	364

Table 3.2 shows the root mean square errors of radiant intensities between the ground truth and the estimated ones. In our simulation we evaluate the estimation at different levels of dynamic range by quantizing the calculated surface intensities in captured images at different quantization level, namely, 8-bit, 12-bit, and 32-bit.

The errors of lower rate of quantization show difficulties of estimation, however, at the quantization level of 12-bit, the errors are around 10 % for experiment 1 and 3. It indicates that we can obtain an reasonable estimation of lighting environment with a camera which can capture images of 12-bit per each color band. Moreover, accurate results at higher rates imply the possibility of acquiring good estimations even in the case of complex lighting environment provided higher rates are available.

Table 3.2: Root mean square errors of estimated radiant intensities at each quantizational level (Max intensity: 10000)

	Experiment 1	Experiment 2	Experiment 3
8-bit	5.6888E+03	8.0826E+04	5.5465E+03
12-bit	1.0474E+03	3.7732E+03	1.0335E+03
32-bit	1.3607E-02	2.8895E-01	3.8754E-01

Table 3.3 shows the ratio of root mean square errors between the calculated radiance and radiance under the estimated lighting environment at each quantizational level. That is, it illustrates possibilities of relighting with estimated lighting environment. The result shows quite reasonable results even though in lower rate of quantization. The main reason of this is that variations of radiant intensity of light sources do not cause much effects to radiance of an object that does not have strong specular reflections. Hence, our method enables us to obtain naturally relighted objects which do not have strong specular reflections over the surface such as mirror-like objects.

Table 3.3: *Ratio of root mean square errors between calculated radiance and radiance under estimated lighting environment at each quantizational level (compared with max value of calculated radiance)*

	Experiment 1	Experiment 2	Experiment 3
8-bit	8.02518E-01 %	1.30026E-01 %	3.84320E+00 %
12-bit	9.33367E-01 %	9.35727E-03 %	3.81730E+00 %
32-bit	7.59055E-07 %	2.01420E-07 %	3.45526E+00 %

3.6.2 Real Scene

3.6.2.1 Estimating Reflectance Parameter of Skeleton Cube

We estimate the reflectance parameter of the skeleton cube in the following conditions:

- ◆ **Skeleton cube:** $100 \times 100 \times 100$ mm, 10 mm size of the pillar
- ◆ **Camera:** Nikon D70 (Table 3.4)
- ◆ **Light source:** a light bulb (40 W)
- ◆ **Geometry:** The skeleton cube and the light source are arranged as shown in Figure 3.14. Relationship between the skeleton cube and the camera is obtained by calibration utilizing the skeleton cube.
- ◆ **Lighting:** We arrange one light bulb in a studio, whose position is estimated a priori. We regard the radiant intensity of the light bulb, L_L , as 1, that is, radiant intensity that is estimated by our method denotes relative value to the light bulb. If an estimation of absolute value of radiant intensity is required, it can be calculated by measuring the radiant intensity of the light bulb with a radiometer and multiplying the estimated value on the measured radiant intensity.
- ◆ **White balance:** We assume that the color of the skeleton cube is pure white, i.e., white balance of observed radiance simply reflects white balance of light sources.

Table 3.4: *Specifications of Nikon D70 [Nik04]*

Type of Camera	Single-lens reflex digital camera
Image Sensor	RGB CCD, 23.7×15.6 mm
Image Size	3008×2000 pixel
Data Format	NEF: 12-bit lossless compression

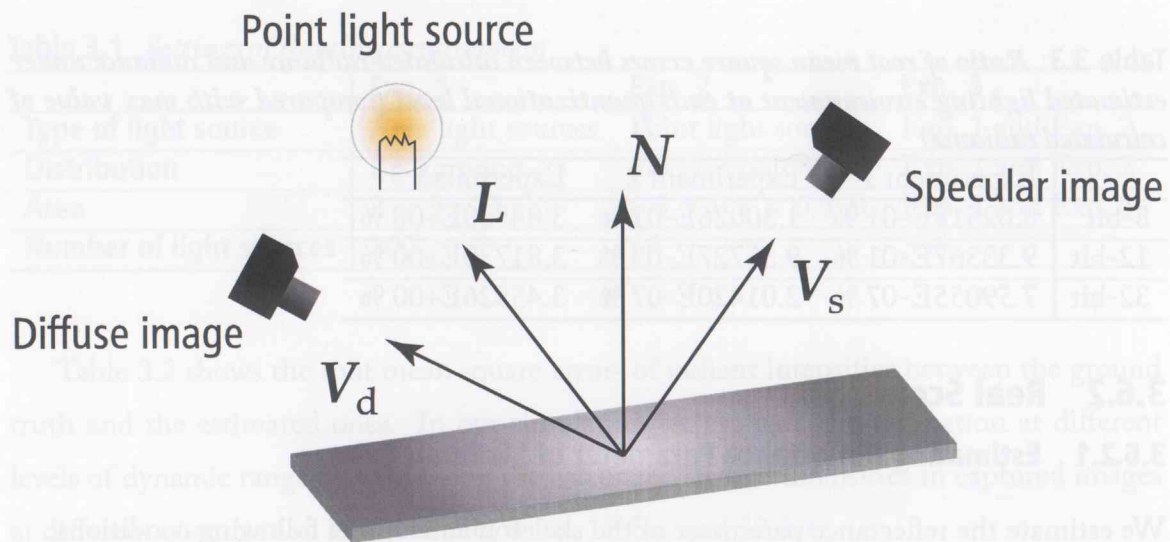


Figure 3.14: Relationship between camera and light source

With the above condition, we estimate the light position and the reflectance parameter by the following procedure:

◆ **Light Position:**

1. Pick up specific points of a cast shadow of the skeleton cube on a graph paper.
2. Calculate lines in 3D space with relationships between the points and the skeleton cube (Figure 3.15).
3. Calculate the intersecting point of the lines, and the point indicates the position of the light bulb. Lines in 3D space, however, do not generally intersect a single point, and moreover the light bulb is not a point but a volume. We therefore define the nearest point to all of the lines as the position of the light bulb.

◆ **Reflectance parameter:**

1. Capture two images of the skeleton cube. One is captured from a position at which we cannot observe strong specular reflection—i.e., a position at which the viewing direction is similar to the light direction, and we denote the image as a *diffuse image*. Another is captured from a position at which we can observe strong specular reflection—i.e., a position at which the viewing direction is similar to the direction of the specular reflection, and we denote the image as

a *specular image*. An example of the relationship between the camera and light source is shown Figure 3.14.

2. Sample the radiance on the surface of the skeleton cube in both of the images. We denote the radiance at point $\mathbf{x}_i$ on the surface of the skeleton cube from the diffuse image as $I_d(\mathbf{x}_i)$, and from the specular image as $I_s(\mathbf{x}_i)$. Note that we select the sampling points where on self-shadows is observed—e.g., the top surface of the skeleton cube.
3. Estimate the diffuse component of the reflectance parameter, k_d , by

$$k_d = \frac{r^2 I_d(\mathbf{x}_i)}{N(\mathbf{x}_i) \cdot \mathbf{L}}, \quad (3.17)$$

where r denotes the distance between the direction of the light bulb and $\mathbf{x}_i$, $N(\mathbf{x}_i)$ the surface normal, and $\mathbf{L}$ unit vector from $\mathbf{x}_i$ to the light bulb. We determine k_d by the least square method of a large number of sampling points.

4. Estimate the specular component, σ and k_s .

Given the radiance of an arbitrary pair of sampling points, we have

$$I_s(\mathbf{x}_i) - I_d(\mathbf{x}_i) = \frac{k_s}{N(\mathbf{x}_i) \cdot \mathbf{V}_s} \exp \left[-\frac{(\cos^{-1}(N(\mathbf{x}_i) \cdot \mathbf{H}))^2}{2\sigma^2} \right], \quad (3.18)$$

where i denotes a label of the sampling point, i.e., $i = \{1, 2\}$, and $\mathbf{V}_s$ the position of the camera that captures the specular image. Remember that we assume the specular reflection is not observed in the diffuse image. We first estimate σ by eliminating k_s from Equation (3.18), that is, we take the ratio of two equations for $\mathbf{x}_1$ and $\mathbf{x}_2$ in Equation (3.18) as

$$\frac{I_s(\mathbf{x}_1) - I_d(\mathbf{x}_1)}{I_s(\mathbf{x}_2) - I_d(\mathbf{x}_2)} = \frac{N(\mathbf{x}_2) \cdot \mathbf{V}_s}{N(\mathbf{x}_1) \cdot \mathbf{V}_s} \cdot \frac{\exp(-sA_1)}{\exp(-sA_2)}, \quad (3.19)$$

where

$$s = \frac{1}{2\sigma^2}, \quad (3.20)$$

$$A_i = (\cos^{-1}(N(\mathbf{x}_i) \cdot \mathbf{H}))^2. \quad (3.21)$$

Calculating natural logarithms of both hands of Equation (3.19), we have

$$s = \frac{\ln NV - \ln I'}{A_1 - A_2}, \quad (3.22)$$

where

$$NV = \frac{N(x_2) \cdot V_s}{N(x_1) \cdot V_s'} \quad (3.23)$$

$$I' = \frac{I_s(x_1) - I_d(x_1)}{I_s(x_2) - I_d(x_2)}. \quad (3.24)$$

We determine s by the least square method of a large number of sampling points, and then we obtain σ from s . Finally, we obtain k_s by

$$k_s = \frac{[I_s(x_i) - I_d(x_i)] \cdot N(x_i) \cdot V_s}{\exp \left[-\frac{(\cos^{-1}(N(x_i) \cdot H))^2}{2\sigma^2} \right]}. \quad (3.25)$$

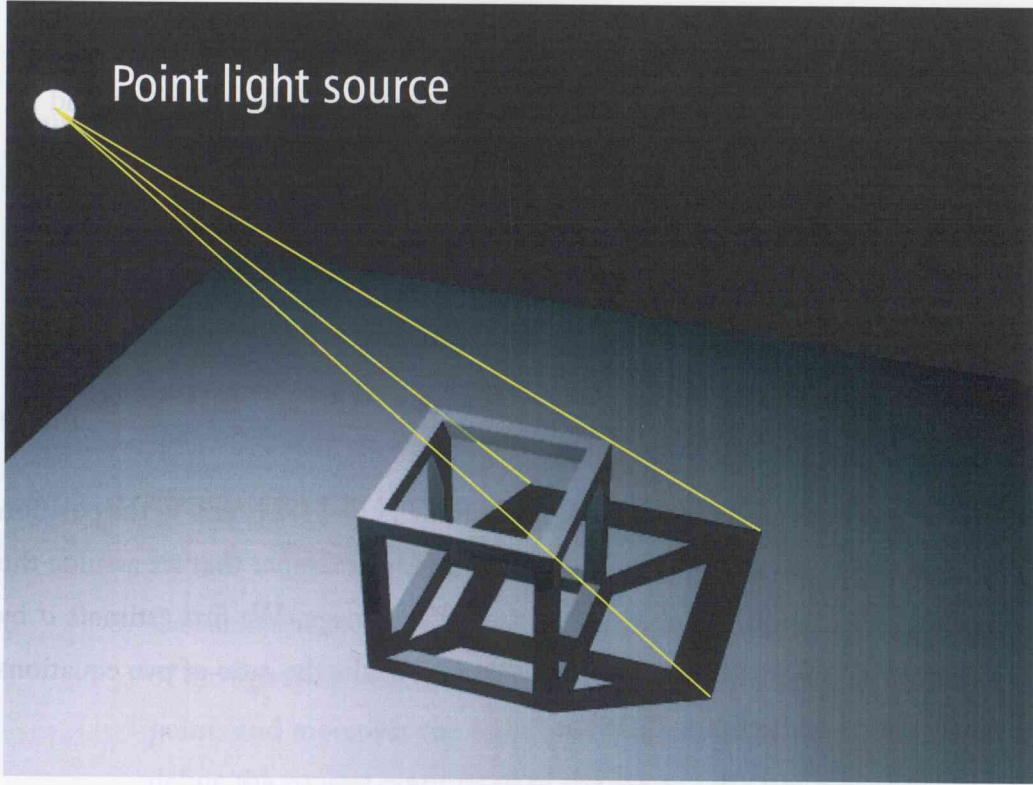
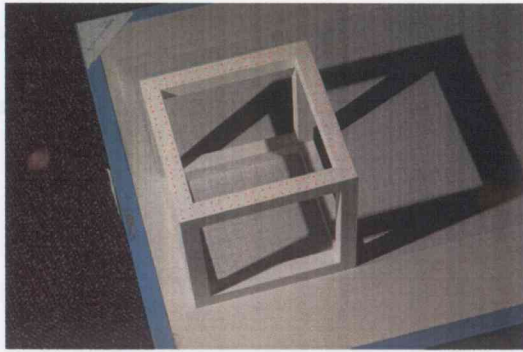


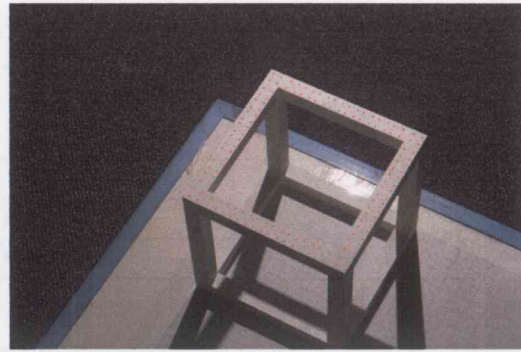
Figure 3.15: Estimation of the position of a light source

The yellow lines are estimated by pointing specific points of the shadow on the floor, and the intersection of the lines indicates the position of the light source.

We show the images for reflectance parameter estimation in Figure 3.16 and the estimated parameters in Table 3.5. We also show square errors between the radiance calculated with the estimated parameter and pixel values in the input images in Table 3.6. They indicate that the estimated parameters fit the reflection model appropriately.



Diffuse image



Specular image

Figure 3.16: Sampling points for reflectance parameter estimation
The red points indicate the sampling points.

Table 3.5: Estimated reflectance parameters

	R	G	B
k_d	1.4220E+9	1.3601E+9	1.1041E+9
k_s	8.1007E+2	7.7726E+2	7.6115E+2
σ	3.4635E-1	3.5010E-1	2.9101E-1

3.6.2.2 Lighting Environment Estimation of Real Scene

We estimate lighting environment of a real scene in the following conditions:

- ◆ **Coordinate system:** The origin of the scene coordinate system is defined on the floor, and the origin of the lighting environment is defined at the left lower corner of the skeleton cube. The each face of the skeleton cube is aligned to be parallel to the axis of the scene coordinate system.
- ◆ **Skeleton cube:** $100 \times 100 \times 100$ mm size, 10 mm pillar. The left lower corner of the skeleton cube is aligned at (697.35, 330.02, 587.98) in the scene coordinate system.
- ◆ **Camera:** Nikon D70. The estimated position of the optical center is (211, -4, 1413) in the scene coordinate system.
- ◆ **Lighting:** 14 fluorescent light tubes and one spot light (Figure 3.17)
- ◆ **Lighting environment model:** We distribute 385 area light sources as the background lighting, and 125 point light sources as the foreground lighting, which are shown in Figure 3.18.

Table 3.6: *Root-mean-square errors of radiance between the pixel values of captured image and the values calculated with the estimated parameters*

The percentages in brackets denote the ratio of the error to the maximum value of the radiance.

	R	G	B
Image 1	2.2494E+2 (5.09%)	2.3213E+2 (5.44%)	1.7270E+2 (4.98%)
Image 2	2.2078E+2 (3.97%)	2.1548E+2 (4.04%)	2.2208E+2 (4.80%)

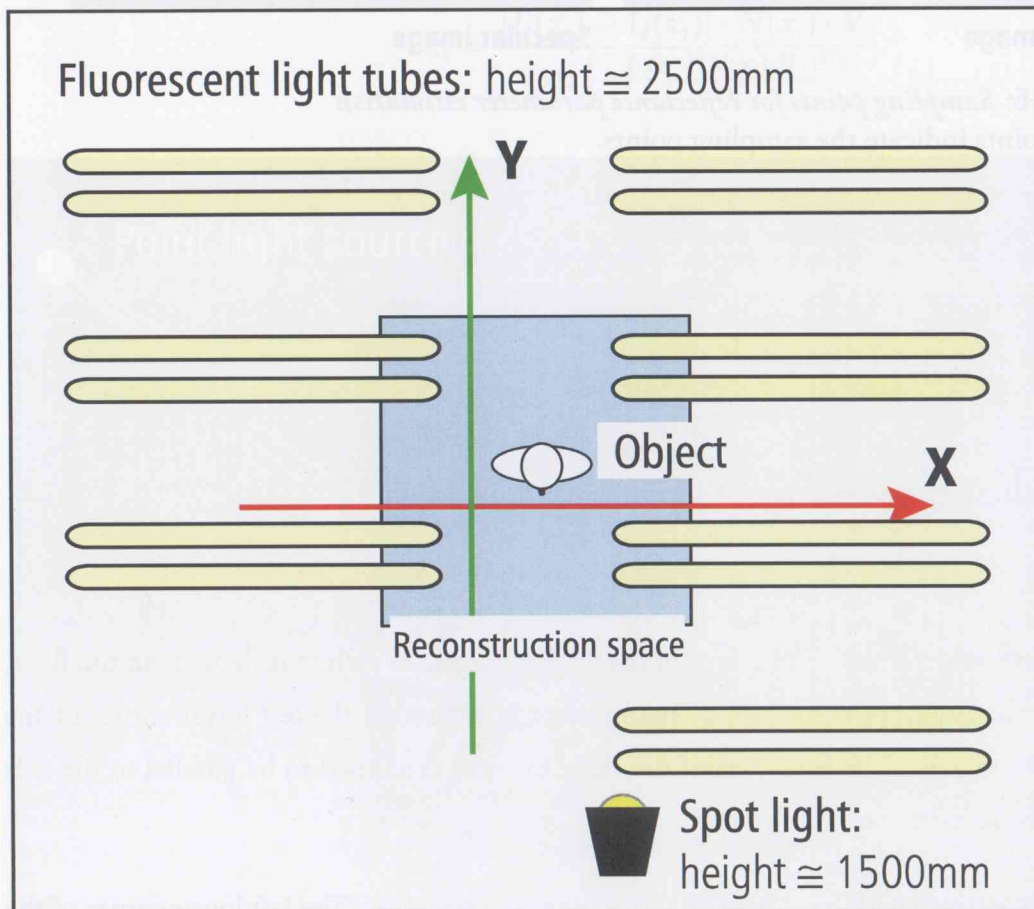


Figure 3.17: *Lighting condition*

Figure 3.19 shows images of the skeleton cube in the studio, and Figure 3.20 images that are captured by the camera which is used for lighting environment estimation. We first calibrate the position and the direction of the skeleton cube in the scene coordinate system with the images in Figure 3.19, and then calibrate the camera with the skeleton cube. We can thus relate a 3D point in the scene to the corresponding pixel of a surface point on the skeleton cube in the images that are captured by the camera.

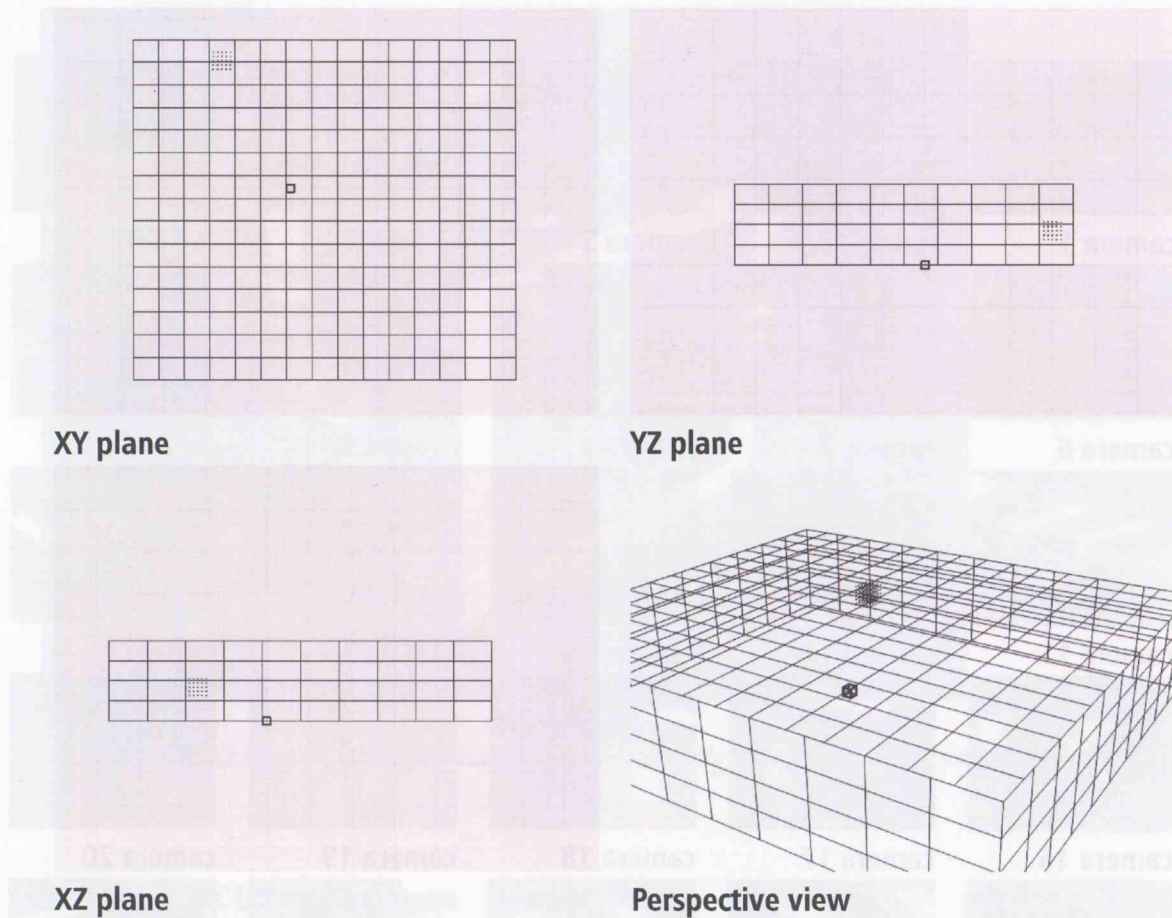


Figure 3.18: *Arrangement of skeleton cube and light sources*

Figure 3.21 shows virtually illuminated skeleton cube under the estimated lighting environment. These images illustrate that shading and shadows of the skeleton cube are naturally represented as the captured images, and the lighting environment in the studio is estimated appropriately.

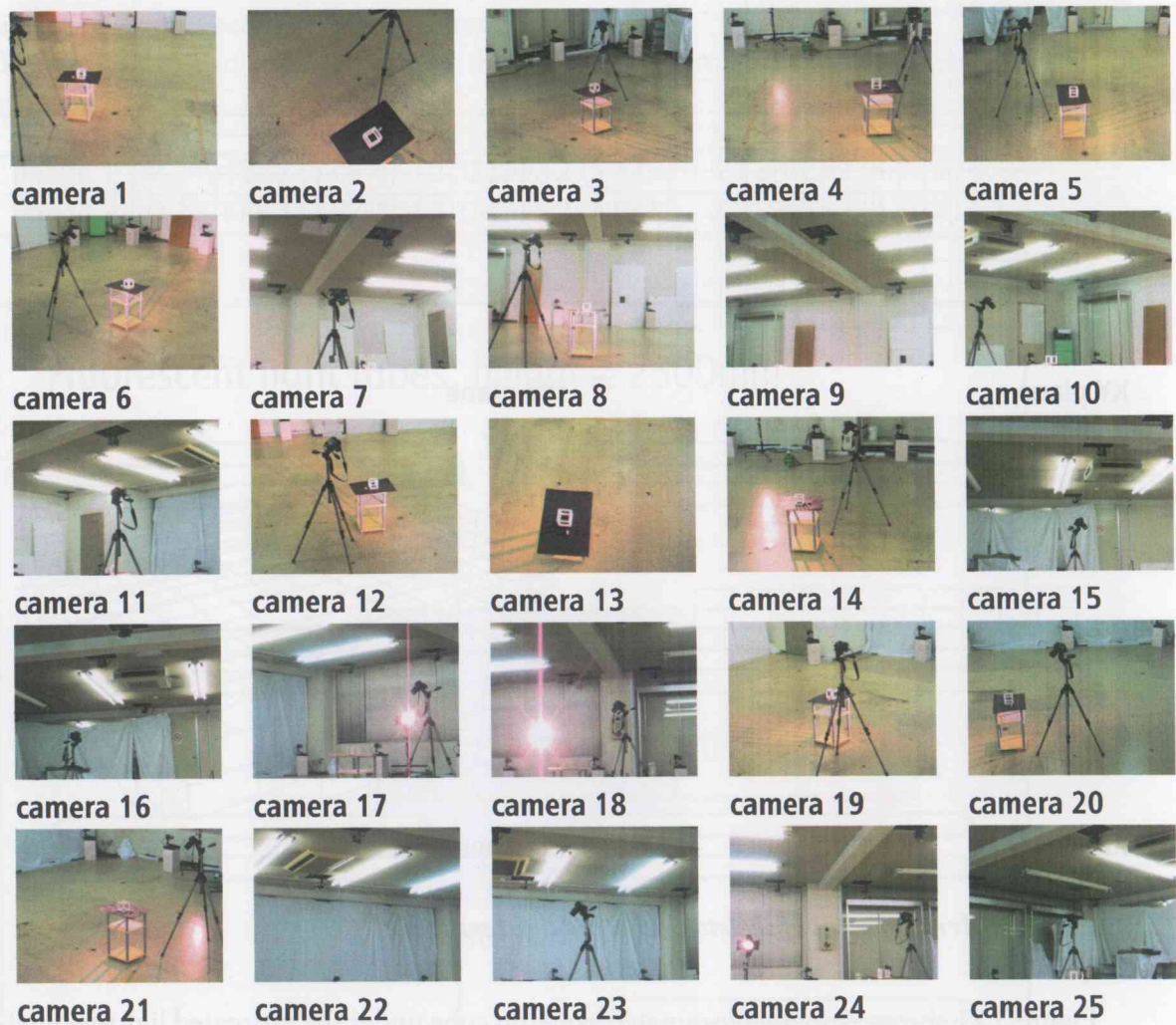


Figure 3.19: Images of the skeleton cube and a digital camera in a studio

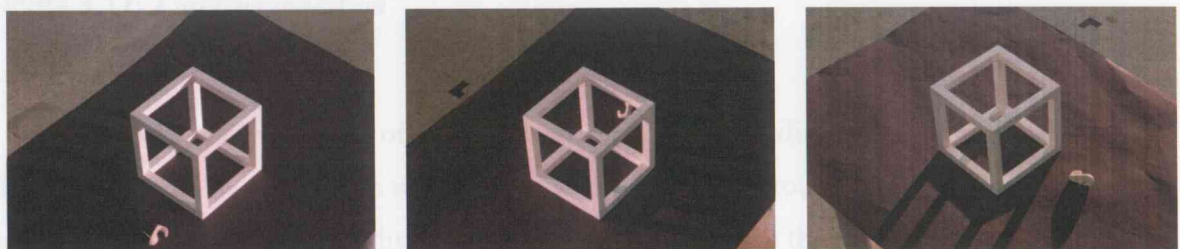


Figure 3.20: Images for lighting environment estimation captured by Nikon D70

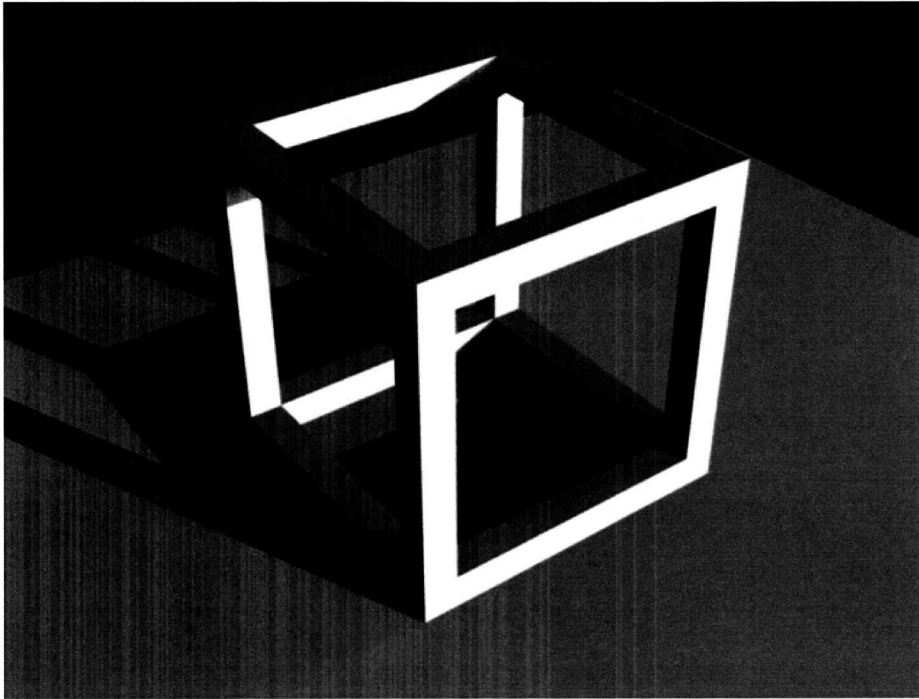


Figure 3.21: *Skeleton cube illuminated by estimated lighting environment*

3.7 Discussions

We have introduced a practical method for the estimation of lighting environment that is represented by a large set of light sources. We estimate the radiant intensities of the light sources using shading and self-shadows of the *skeleton cube*, which we propose as a reference object featuring both simplicity and complexity for a probe of lighting environment. The skeleton cube can also be utilized as a reference object for geometric calibration for cameras. That is, it allows us to calibrate both of the geometry and photometry in a scene. We have presented its effectiveness with experimental results of computer simulations and a real scene.

We have also presented a model for representing lighting environment as a combination of *foreground* and *background lighting*, which are composed of a set of point light sources and area light sources, respectively. Utilizing our practical modeling and estimation method of area light sources as well as point light sources, we can estimate a lighting environment of real scenes effectively. Our method enables us to represent shading and shadowing in which the effects of near light sources have to be considered, which were difficult to realize by previous approaches.

Chapter 4

Difference Sphere: An approach to Near Light Source Estimation

4.1 Introduction

In this chapter, we present a theoretical approach for estimating parameters of light sources, i.e., ambient light, directional light sources, and in particular near point light sources including their radiant intensity.

We propose to employ a pair of reference spheres as light probes and introduce the difference sphere that we acquire by differencing the radiance of two image regions of the reference spheres. Since the effect by directional light sources and ambient light is eliminated by differencing, the key advantage of considering the difference sphere is that it enables us to estimate near point light sources including their radiance, which has been difficult to achieve in previous efforts where only distant directional light sources were assumed.

We also show that analysis of gray level contours on spherical surfaces facilitates separate identification of multiple combined light sources and is well suited to the difference sphere. Once we estimate near point light sources with the difference sphere, we update the input image by eliminating their influence and then estimate other remaining light sources, that is, directional light sources and ambient light.

In the rest of the chapter, we define the lighting and reflectance model in Section 4.2. While introducing the notion of difference sphere, we discuss its characteristics in Section 4.3. Section 4.4 provides the description of our algorithm of light source estimation followed by evaluations of our proposed method in Section 4.5. Finally we conclude the chapter with discussions in Section 4.6.

4.2 Model Definitions

4.2.1 Assumptions

For our light source estimation we employ a pair of spheres with known size, which we call *reference spheres*, and assume a bidirectional reflectance distribution function as Lambertian. We place the spheres in a way that they do not occlude or cast shadows to each other. We also assume that the camera for capturing images and the reference spheres are accurately calibrated. We then deal with measured image irradiance which we in this paper refer to as image intensity. In other words, we assume that the transformation function from the radiance to the pixel gray value is spatially uniform and linear with zero bias so that the scale factor is equal to 1. We consider the mutual reflection between the spheres as minor and ignorable.

4.2.2 Coordinate Systems

As illustrated in Figure 4.1 we consider the *scene coordinate system* (X, Y, Z) and the *sphere coordinate system* (x, y, z) . Given a reference sphere, A , we align each axis of the sphere coordinate system, x_A , y_A , and z_A , parallel to X , Y , and Z axis of the scene coordinate system, respectively. We also utilize *surface coordinate system* (θ, ϕ) in order to specify angular positions of spherical surface points.

4.2.3 Lighting Environment

As already mentioned, we deal with three types of light sources, i.e., near point light source, distant directional light source, and ambient light. In the following descriptions we denote the reflectance coefficient of the reference sphere as η .

Near point light source: As shown in Figure 4.2, at point X , let $\alpha_p(X, P)$ be an angle between the surface normal and the line from the light source. The radiance, $I_p(X; P)$, is then given by

$$I_p(X; P) = \eta L_p \max[\cos \alpha_p(X, P), 0] / d(X, P)^2, \quad (4.1)$$

where L_p denotes radiant intensity of the point light source, P the position of it, and $d(X, P)$ the distance between the light source and X .

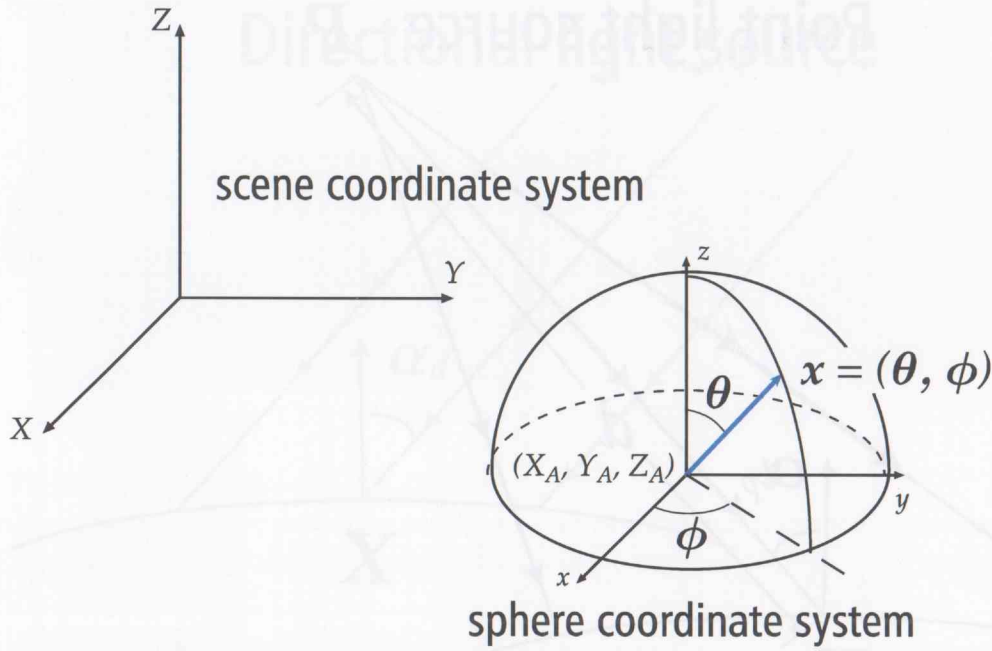


Figure 4.1: *Coordinate systems.*

Directional light source: As shown in Figure 4.3, let $\alpha_d(\mathbf{X})$ be an angle between the surface normal at point $\mathbf{X}$ and the direction of the light source. With the radiant intensity of the directional light source, L_d , the radiance, $I_d(\mathbf{X})$, is given by

$$I_d(\mathbf{X}) = \eta L_d \max[\cos \alpha_d(\mathbf{X}), 0]. \quad (4.2)$$

Ambient light: It provides constant light for a scene. We consider it as bias in this paper. If the scene were illuminated by ambient light L_a alone, the radiance, I_a , of the reference sphere would simply be

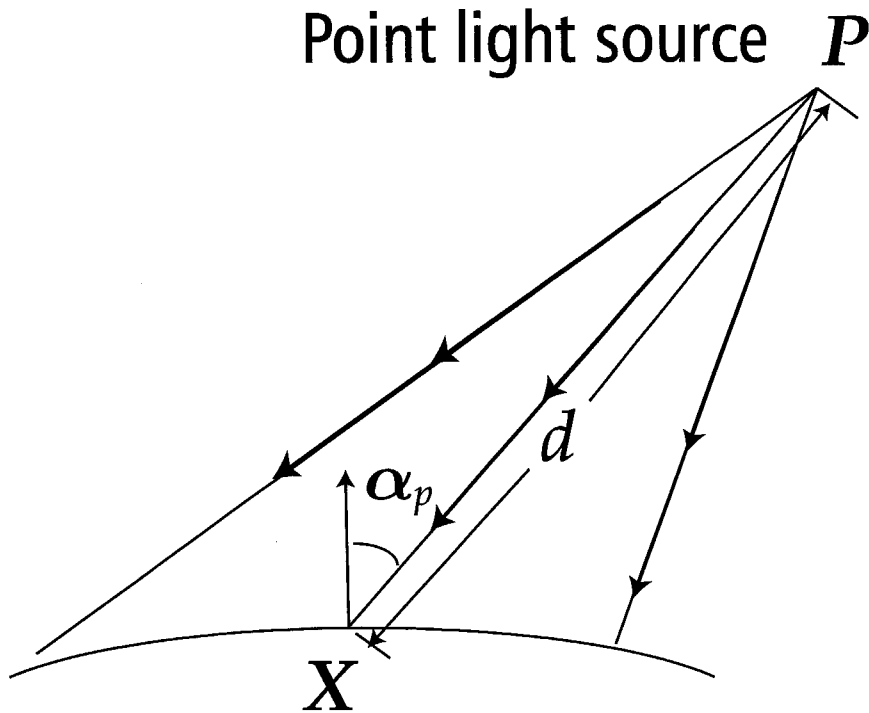
$$I_a = \eta L_a. \quad (4.3)$$

4.2.4 Radiance

Under the above described light sources, the total radiance can be modeled in general as

$$I(\mathbf{X}) = \sum_{i=1}^s I_p^{[i]}(\mathbf{X}; \mathbf{P}[i]) + \sum_{j=1}^t I_d^{[j]}(\mathbf{X}) + I_a, \quad (4.4)$$

where s and t are (unknown) numbers of point light sources and directional light sources, respectively, i and j are indices of them, and $\mathbf{P}[i]$ is the 3D position of the i -th point light source in the scene coordinate system.

Figure 4.2: *Point light source*

4.3 Characteristics of Single and Difference Sphere

We use the term, *single sphere*, interchangeably with reference sphere to explicitly distinguish it from *difference sphere*, that is, geometric and photometric characteristics are identical to the reference sphere. From the single sphere, we can derive characteristics for light source estimation, which we describe the following sections.

Estimating light sources in a complex lighting environment, however, is difficult because it becomes harder to detect features from radiance of a surface of the single sphere under such an environment. We thus introduce a notion of *difference sphere*, which we acquire by differencing two image regions of reference spheres.

In the following sections, we describe the characteristics of the single and difference sphere in detail.

4.3.1 Radiance

4.3.1.1 Single Sphere

Let us first consider a single sphere, A , with shading in the scene and formulate the radiance on the surface of the sphere in the sphere coordinate system.

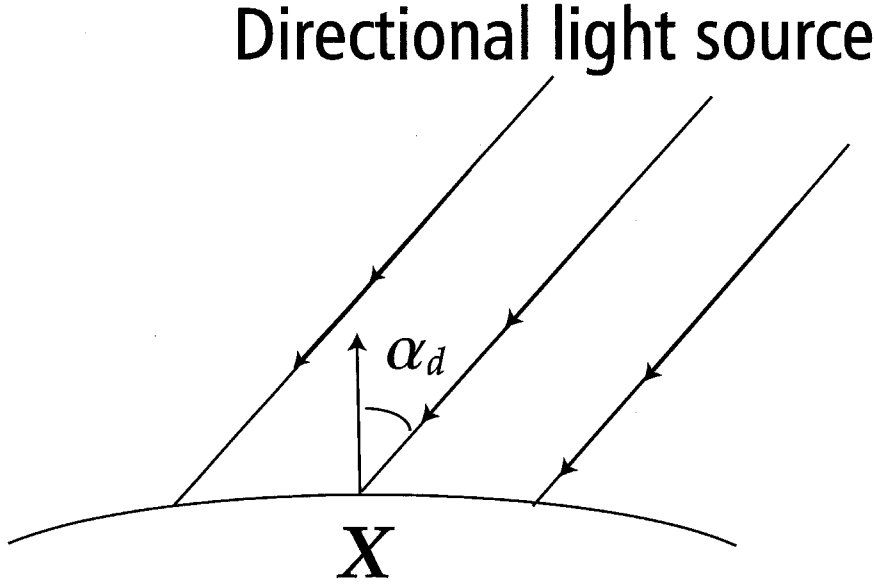


Figure 4.3: *Directional light source*

Let x_A represent a point on the surface of single sphere A , and $\mathcal{F}_A(x_A)$ the 3D position of x_A in the scene coordinate system. Representing the radiance, $I(X)$, in equation (4.4) by $I_A(x_A)$, and thus replacing X with $\mathcal{F}(x_A)$, the constituents of it according to equations (4.1)-(4.3) are given by

$$I_p^{[i]} = \eta_A L_p^{[i]} \max[\cos \alpha_p^{[i]}(\mathcal{F}_A(x_A), P[i]), 0] / (D(\mathcal{F}_A(x_A), P[i]))^2, \quad (4.5)$$

$$I_d^{[j]} = \eta_A L_d^{[j]} \max[\cos \alpha_d^{[j]}(\mathcal{F}_A(x_A)), 0], \quad (4.6)$$

$$I_a = \eta_A L_a, \quad (4.7)$$

where η_A is the diffuse coefficient, $L_p^{[i]}$ and $L_d^{[j]}$ denote the radiant intensity of the i -th point light source and the j -th directional light source, respectively. Note that both $\cos \alpha_p$ and $\cos \alpha_d$ accompany an index of light source accordingly.

4.3.1.2 Difference Sphere

We virtually generate a difference sphere, “ $A - B$ ”, from a pair of reference spheres, A and B , that has the following properties.

- ◆ **Geometry:** The location and the radius of the difference sphere is inherited from those of single sphere A .

- ◆ **Photometry:** Let $I_A(\theta, \phi)$ and $I_B(\theta, \phi)$ denote the radiance of single spheres, A and B , respectively. The radiance of difference sphere, $A - B$, is ¹

$$I_{A-B}(\theta, \phi) = I_A(\theta, \phi) - I_B(\theta, \phi). \quad (4.8)$$

From equations (4.4) - (4.7) and (4.8) the radiance, $I_{A-B}(\mathbf{x}_{A-B})$, at point $\mathbf{x}_{A-B}$ on the surface of difference sphere $A - B$ is defined as follows:

$$\begin{aligned} I_{A-B}(\mathbf{x}_{A-B}) &= I_A(\mathbf{x}_A) - I_B(\mathbf{x}_B) \\ &= \sum_{i=1}^s I_p^{[i]}(\mathcal{F}_A(\mathbf{x}_A); P[i]) - \sum_{i=1}^s I_p^{[i]}(\mathcal{F}_B(\mathbf{x}_B); P[i]) \end{aligned} \quad (4.9)$$

where surface coordinates of $\mathbf{x}_{A-B}$, $\mathbf{x}_A$, and $\mathbf{x}_B$ on their corresponding spheres are all equivalent. Equation (4.9) is due to the fact that the differencing operation in equation (4.8) eliminates lighting effects caused by all the directional light sources and ambient light. That is, the illumination on a difference sphere is caused only by point light sources.

Further, since the differencing operation generally gives rise to negative radiance as well as positive ones, we could interpret the surface radiance of a difference sphere as if they were independently generated by *positive* and *negative* point light sources. Thus, a difference sphere is virtually illuminated by *positive* and *negative* point light sources, each of which introduces positive and negative radiance.

Figure 4.4 illustrates a difference sphere that is generated by a pair of single spheres, and show the positive and the negative radiance. In the subsequent analysis of surface radiance for estimating point light sources, negative radiance can be treated just equally as positive radiance without a loss of generality. Moreover, it turns out that we only need to analyze either positive radiance or negative one for estimating the point light sources since the influence of the light sources basically appear in both of them. However, note that the acquired lighting parameters in case of analyzing negative radiance should be interpreted in the coordinate system whose origin is at the center of single sphere B , instead of sphere A . Although it does not really matter which one to choose in the example of Figure 4.4, it is in general sensible to choose the radiance that represents larger amount of radiant energy.

¹Note that in the shading analysis that is described later, $I_{A-B}(\theta, \phi)$ becomes undefined for those (θ, ϕ) where the corresponding surface points on single spheres A or B cannot be observed by a camera.

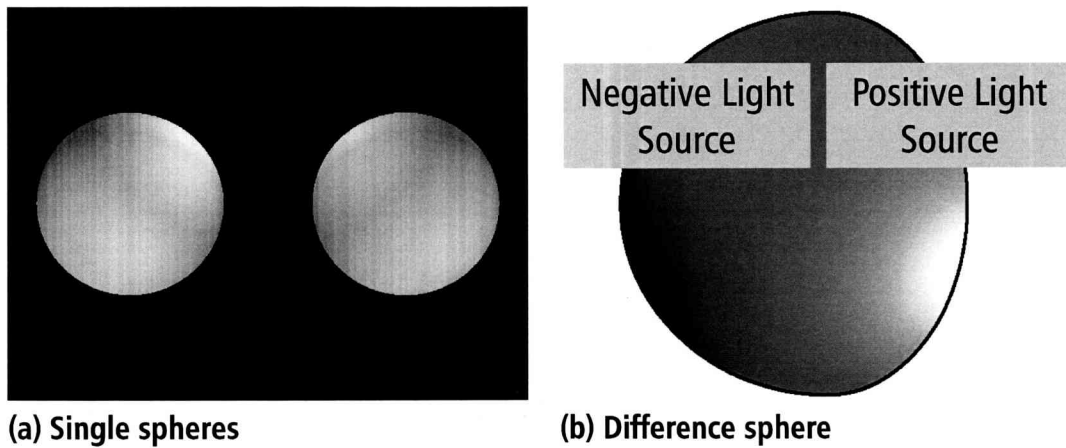


Figure 4.4: *Single spheres and difference sphere*

The gray-level of (b) shows the radiance of the difference sphere; the bright/dark area reflects positive/negative values. The intensities of the image are emphasized for display reason.

4.3.2 Surface Analysis

4.3.2.1 Classification of Surfaces

Let us investigate the characteristics of the sphere surface while separating it into three categories² depending on the types of illuminating light sources (see Figure 4.5). They are,

- ◆ **S-surface:** The surface illuminated by a single light source (and the ambient light). In particular, we call *S-surface* that is illuminated by a single point light source S_p -*surface* and that by a single directional light source S_d -*surface*.
- ◆ **M-surface:** The surface illuminated by multiple point and/or directional light sources.
- ◆ **A-surface:** The surface illuminated by ambient light alone.

Now, we put our focus on the characteristics of *S-surface* in order to estimate parameters of light sources. We will see that the *S-surface* has such features as described below.

- ◆ **Feature 1:** A set of points on *S-surface* with identical radiance forms an arc on a plane in 3D scene. We call the plane *feature plane*.
- ◆ **Feature 2:** The surface normal of the feature plane denotes the direction of the point/directional light source which illuminates the *S-surface*.
- ◆ **Feature 3:** A *radiance ratio* of a group of feature planes in an identical *S-surface* characterizes the radiant intensity of the light source.

² Completely dark surface due to shadow is set aside.

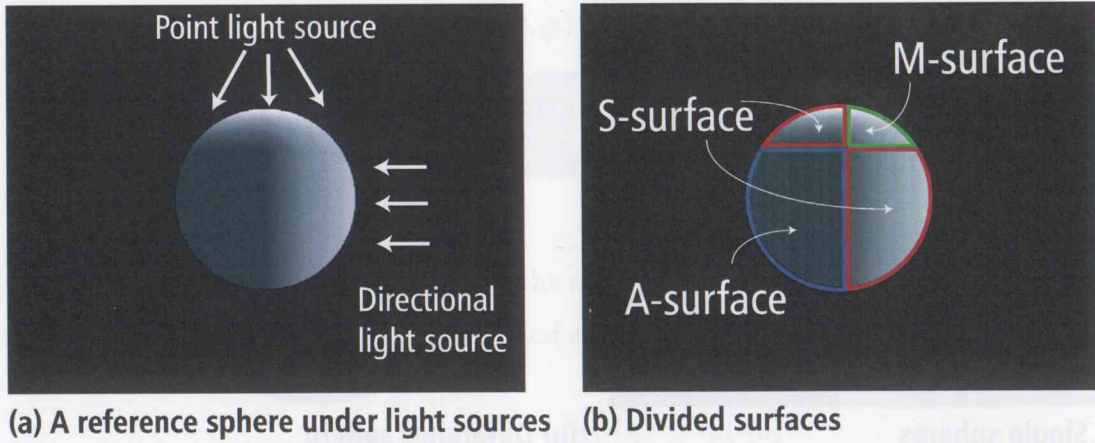


Figure 4.5: Surface classification

In the remainder of the section, we explain the above features in detail. Preceded by the analyses in the cases of a single point light source and a single directional light source which show the mechanisms of our light source estimation, we derive how the difference sphere simplifies the procedure of light source estimation. Note that we additionally define the following notations.

- ◆ $O = (O_X, O_Y, O_Z)$: The center of a reference sphere.
- ◆ $P = (P_X, P_Y, P_Z)$: The light source position.
- ◆ $l = (l_X, l_Y, l_Z)$: The direction of the light source.
- ◆ r : The radius of a reference sphere.

4.3.2.2 S_d -surface of Single Sphere

The bold line in Figure 4.6 illustrates a relationship between a directional light source and a single sphere. We show a proof of **Feature 1** and **Feature 2** of S_d -surface in the following:

Proof. Radiance of points that have an identical value on S_d -surface is given by

$$I_d = \eta L_d N \cdot l, \quad (4.10)$$

where N denotes a normal vector of point $\mathbf{x}$ on the sphere, and l denotes a direction of a light source that illuminates the S_d -surface, i.e., $l = (l_X, l_Y, l_Z)$. We rewrite N with $\mathbf{X} = \mathcal{F}(\mathbf{x}) = (X, Y, Z)$ and O as

$$N = \frac{\mathbf{X} - O}{r}. \quad (4.11)$$

With Equation (4.11), we rewrite Equation (4.10) and obtain as the following equation

$$l \cdot X - l \cdot O - \frac{I_d r}{\eta L_d} = 0. \quad (4.12)$$

Equation (4.12) indicates that points that have an identical radiance are on an identical plane whose surface orientation is $l = (l_X, l_Y, l_Z)$.

Assuming that lighting direction l is identical to the z-axis (we do not lose a generality by this assumption, because we can arbitrarily define a coordinate system), we have $l = (0, 0, 1)$. Substituting l for Equation (4.12), we obtain

$$Z = O_Z + \frac{I_d r}{\eta L_d}. \quad (4.13)$$

Moreover, since X is a point on the sphere, we have

$$(X - O_X)^2 + (Y - O_Y)^2 + (Z - O_Z)^2 = r^2. \quad (4.14)$$

With Equation (4.13) and (4.14), we finally obtain

$$(X - O_X)^2 + (Y - O_Y)^2 = r^2 - \left(\frac{I_d r}{\eta L_d} \right)^2. \quad (4.15)$$

Thus a set of points on S_d -surface with identical radiance forms an arc on a feature plane in 3D scene (**Feature 1**), and the surface normal of the feature plane denotes the direction of a directional light source (**Feature 2**). $\square$

Note that Equation (4.12) tells that all feature planes that is defined in an S_d -surface are parallel to each other. Furthermore, a group of feature planes with different radiance characterizes the radiant intensity of the directional light source illuminating the S_d -surface. Since the radiance of the group of feature planes varies according to $\beta_d(x)$, we describe the radiance with $\beta_d(x)$ instead of $\alpha_d(x)$. That is, the radiance, $I(x)$, at x on an S_d -surface is given by

$$I(x) = \eta L_d \cos \alpha_d(x) + \eta L_d. \quad (4.16)$$

We rewrite this equation with $\beta_d(x)$ as

$$I(x) = \eta L_d \cos \beta_d(x) + \eta L_d, \quad (4.17)$$

where $\beta_d(x) = \alpha_d(x)$. Selecting points on any three independent feature planes, we have the radiance ratio (**Feature 3**)

$$\frac{I(x_1) - I(x_2)}{I(x_1) - I(x_3)} = \frac{I'(x_1) - I'(x_2)}{I'(x_1) - I'(x_3)} \quad (4.18)$$

where $I'(x) = \cos \beta_d(x)$.

To summarize the analysis on S -surface, if the radiance ratio is given by Equation (4.18), we can deduce that the S -surface is an S_d -surface and thus can obtain L_d and L_a from Equation (4.17).

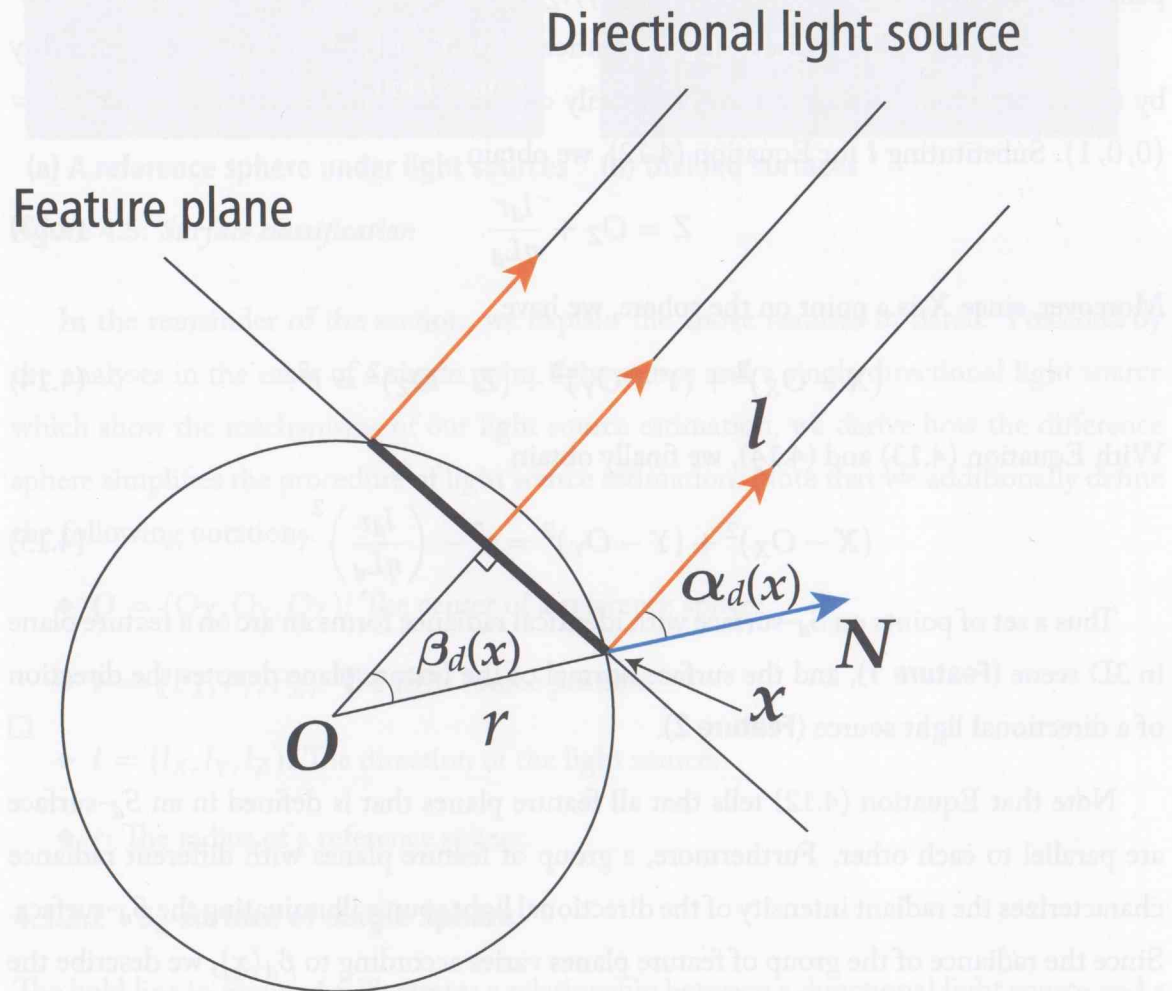


Figure 4.6: Relationship between a single directional light source and a reference sphere

4.3.2.3 S_p -surface of Single Sphere

Figure 4.7 shows a relationship between a point light source and a single sphere. We first show a proof of **Feature 1** and **Feature 2** of S_p -surface in the following:

Proof. As shown in Figure 4.7, a distance between a point light source and points on the S_p -surface that have an identical radiance is denoted as $d(\mathcal{F}(x), P)$. The radiance is given

by

$$I_p = \eta L_p N \cdot l' / d(\mathcal{F}(x), P)^2, \quad (4.19)$$

where $l' = (P - X) / d(\mathcal{F}(x), P)$.

We rewrite Equation (4.19) with X and l as

$$I_p = \frac{\eta L_p}{d(\mathcal{F}(x), P)^2} \frac{(X - O)}{r} \frac{d_p l - X}{d(\mathcal{F}(x), P)}, \quad (4.20)$$

where d_p denotes the distance between P and O . Expanding Equation (4.20), we have

$$l \cdot X - l \cdot O - \frac{1}{d_p} \left(r^2 - d(\mathcal{F}(x), P)^3 \frac{I_p r}{\eta L_p} \right) = 0. \quad (4.21)$$

Since the distance between the light source and the points that have the identical radiance is constant, Equation (4.21) indicates that the points are on an identical plane whose surface orientation is $l = (l_X, l_Y, l_Z)$.

Assuming that lighting direction l is identical to the z-axis, i.e., $l = (0, 0, 1)$, we have

$$Z = O_Z + \frac{1}{d_p} \left(r^2 + d(\mathcal{F}(x), P)^3 \frac{I_p r}{\eta L_p} \right). \quad (4.22)$$

Moreover, since X is a point on the sphere, with Equation (4.22) and (4.14), we finally obtain

$$(X - O_X)^2 + (Y - O_Y)^2 = r^2 - \left[\frac{1}{d_p} \left(r^2 + d(\mathcal{F}(x), P)^3 \frac{I_p r}{\eta L_p} \right) \right]^2. \quad (4.23)$$

Thus a set of points on S_p -surface with identical radiance forms an arc on a feature plane in 3D scene (**Feature 1**), and the surface normal of the feature plane denotes the direction of a light source (**Feature 2**). $\square$

The radiance, $I(x)$, at x on a feature plane is

$$I(x) = \eta L_p \cos \alpha_p(\mathcal{F}(x), P) / d(\mathcal{F}(x), P)^2 + \eta L_a, \quad (4.24)$$

where $\cos \alpha_p(\mathcal{F}(x), P)$ and $d(\mathcal{F}(x), P)$ can be formulated as

$$\cos \alpha_p(\mathcal{F}(x), P) = \frac{d_p^2 - r^2 - d(\mathcal{F}(x), P)^2}{2rd(\mathcal{F}(x), P)}, \quad (4.25)$$

$$d(\mathcal{F}(x), P) = \sqrt{d_p^2 + r^2 - 2rd_p \cos \beta_p(x, l)}. \quad (4.26)$$

By substituting the above equations into Equation (4.24), we have

$$I(\mathbf{x}) = \eta L_p \frac{d_p \cos \beta_p(\mathbf{x}, l) - r}{\left[d_p^2 + r^2 - 2rd_p \cos \beta_p(\mathbf{x}, l) \right]^{\frac{3}{2}}} + \eta L_a. \quad (4.27)$$

Equation (4.27) indicates that the radiance depends on unknown L_p , L_a , and d_p . However, the ambient light term can be eliminated by subtracting the radiance at points on any two feature planes. Besides, the lighting effect of the point light source due to L_p can also be canceled by selecting yet another point on any other feature plane and computing the radiance ratio (**Feature 3**),

$$\frac{I(\mathbf{x}_1) - I(\mathbf{x}_2)}{I(\mathbf{x}_1) - I(\mathbf{x}_3)} = \frac{I'(\mathbf{x}_1) - I'(\mathbf{x}_2)}{I'(\mathbf{x}_1) - I'(\mathbf{x}_3)} \quad (4.28)$$

where

$$I'(\mathbf{x}) = \frac{d_p \cos \beta_p(\mathbf{x}, l) - r}{\left[d_p^2 + r^2 - 2rd_p \cos \beta_p(\mathbf{x}, l) \right]^{\frac{3}{2}}}.$$

Solving Equation (4.28), we can theoretically obtain d_p , and thereby L_p and L_a from Equation (4.27). Nevertheless, the solution to d_p is not indeed straightforward, which indicates the difficulty in estimating the point light sources solely by a single sphere.

4.3.2.4 S_p -surface of difference Sphere

S-surface of a difference sphere has similar characteristics as does S_p -surface of a single sphere, except that the factor of ambient light is precluded. That is, radiance $I(\mathbf{x})$ at $\mathbf{x}$ on a difference sphere is

$$I(\mathbf{x}) = \eta L_p^* \cos \alpha_p(\mathbf{x}, \mathbf{P}) / d(\mathcal{F}(\mathbf{x}), \mathbf{P})^2, \quad (4.29)$$

where L_p^* denotes the radiant intensity of a positive or negative point light source. Thus the radiance ratio is given by

$$\frac{I(\mathbf{x}_1)}{I(\mathbf{x}_2)} = \frac{I'(\mathbf{x}_1)}{I'(\mathbf{x}_2)} \quad (4.30)$$

where

$$I'(\mathbf{x}) = \frac{d_p \cos \beta_p(\mathbf{x}, l) - r}{\left[d_p^2 + r^2 - 2rd_p \cos \beta_p(\mathbf{x}, l) \right]^{\frac{3}{2}}}.$$

Solving Equation (4.30) analogously, we can obtain d_p , and then L_p^* from Equation (4.29). It can be seen that a more practical solution is allowed in the case with difference sphere and it is much simpler than the case of solving Equation (4.28) with single sphere.

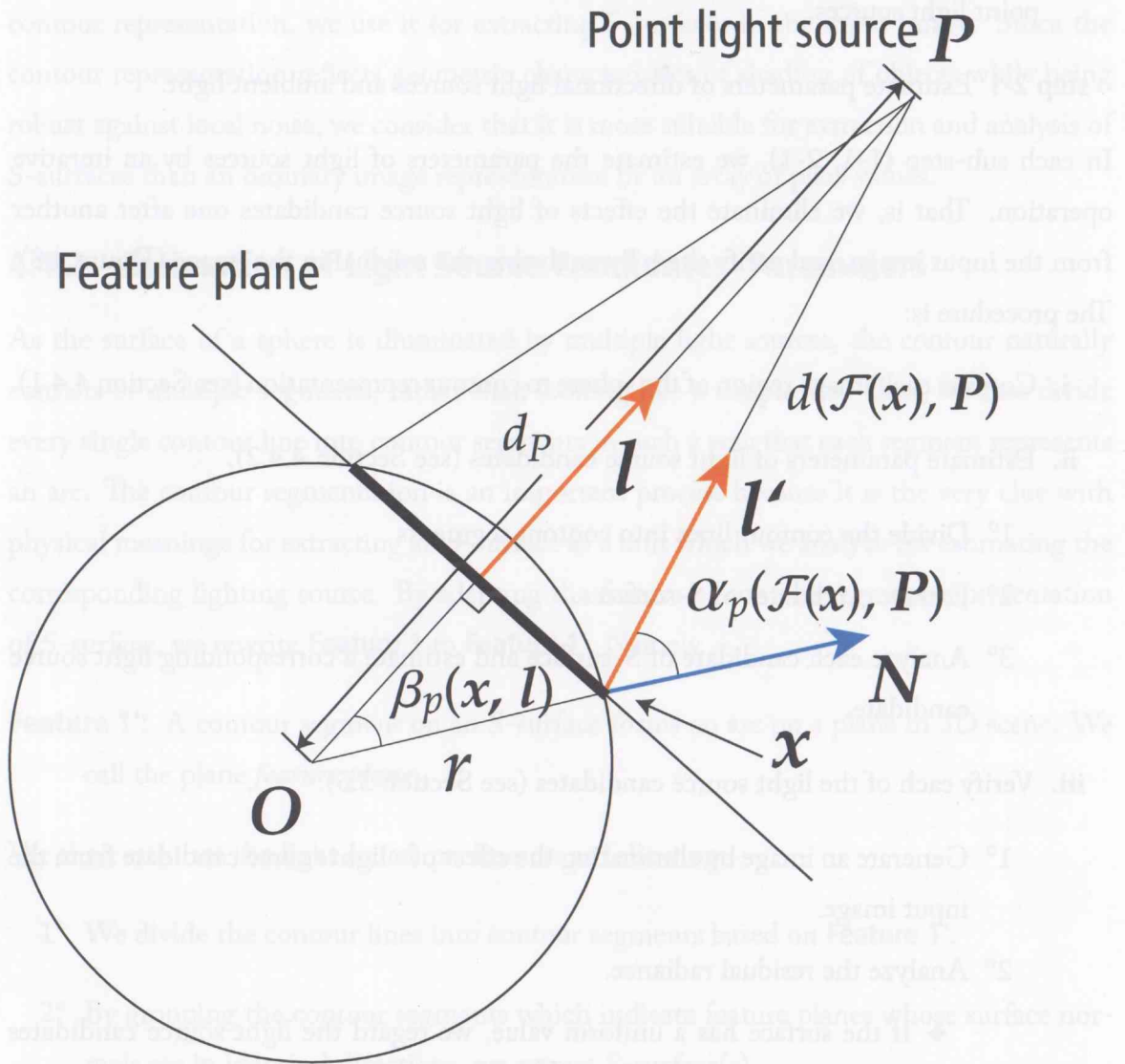


Figure 4.7: Relationship between a single point light source and a reference sphere.

4.4 Algorithm of Light Source Estimation

Based on the above discussions we propose a twofold algorithm for light source estimation as following:

step 0 Capture an image.

step 1 Generate an image of a difference sphere

step 1-1 Estimate parameters of point light sources

step 2 Update the input image by eliminating the lighting effects that is due to the estimated

point light sources.

step 2-1 Estimate parameters of directional light sources and ambient light.

In each sub-step (1-1, 2-1), we estimate the parameters of light sources by an iterative operation. That is, we eliminate the effects of light source candidates one after another from the input image, and verify them by analyzing the residual in the image (Figure 4.8). The procedure is:

- i. Convert each image region of the sphere to contour representation (see Section 4.4.1).
- ii. Estimate parameters of light source candidates (see Section 4.4.2).
 - 1° Divide the contour lines into contour segments
 - 2° Extract candidates of S -surfaces.
 - 3° Analyze each candidate of S -surface and estimate a corresponding light source candidate.
- iii. Verify each of the light source candidates (see Section 3.5).
 - 1° Generate an image by eliminating the effect of a light source candidate from the input image.
 - 2° Analyze the residual radiance.
 - ❖ If the surface has a uniform value, we regard the light source candidates that have been employed for the generation of the input image as correct and terminate the procedure.
 - ❖ If there exists a surface with negative values (*negative surface*), the light source candidate is judged to be incorrect.
 - ❖ Otherwise, update the input image to the image that is generated by **iii-1°** and go to **i**.

We now describe the details of the processes in sub-steps **i** to **iii**.

4.4.1 Region Representation with Gray-Level Contour Lines

The contour representation can be described as a geographic map where the radiance levels of pixels are regarded as height at corresponding locations [AK97]. After obtaining the

contour representation, we use it for extracting S -surfaces in the input image. Since the contour representation reflects geometric characteristics of shading of objects while being robust against local noise, we consider that it is more suitable for extraction and analysis of S -surfaces than an ordinary image representation by an array of pixel values.

4.4.2 Estimation of Light Source Candidates' Parameters

As the surface of a sphere is illuminated by multiple light sources, the contour naturally consists of multiple segments, rather than looking like a simple arc. Thus, we first divide every single contour line into contour segments in such a way that each segment represents an arc. The contour segmentation is an important process because it is the very clue with physical meanings for extracting an S -surface as a unit which we analyze for estimating the corresponding lighting source. By adapting the feature of contour segment representation of S -surface, we rewrite **Feature 1** to **Feature 1'**. Namely,

Feature 1': A contour segment on an S -surface forms an arc on a plane in 3D scene. We call the plane *feature plane*.

We then estimate the light source parameters as following:

- 1° We divide the contour lines into contour segments based on **Feature 1'**.
- 2° By grouping the contour segments which indicate feature planes whose surface normals are in identical directions, we extract S -surface(s).
- 3° We estimate the parameters of the light sources according to **Feature 2** while calculating the radiance ratio to determine the radiant intensity by **Feature 3**.

4.4.3 Verification of Estimated Light Source Candidates

We verify the estimated light source candidates by analyzing images that are generated by eliminating their possible effects from the input image. That is, if the residual has negative values³, we can determine that the estimation is not correct.

Suppose a reference sphere that is illuminated by three directional light sources (Figure 4.8). While there are four candidates of S -surface in the input image, correct S -surfaces are

³In **step 1**, the residual should be zero. In **step 2**, the residual can also take a uniform value, which is regarded as the effect of ambient light.

C_2 and C_4 , and the others are M -surfaces. Although it is not possible to identify S -surfaces among the candidates only by the iso-radiance contours, we can identify an M -surface by analyzing the residual that is generated by eliminating the effects of the light source candidates. That is, we can identify that C_3 is an M -surface since the *negative surface* appears in the third figure from the left in the first verification, which is generated by eliminating the effect of the light source candidate from C_3 . Continuing the procedure iteratively with the updated images, we find that three paths give the correct estimations. As these paths allow estimations of identical parameters of the same light sources, the lighting environment is correctly estimated.

For general lighting setting, as long as at least one correct S -surface exists, we can estimate the corresponding light source and continue the procedure by eliminating its effect from the input image accordingly.

As we described above, our method investigates all of the possible paths, and determines the path that indicates the correct set of light sources. That is, it is a kind of try-and-error approaches which are required to consider an accumulation of errors or a selection of estimation paths, i.e., a global optimization. To obtain an optimal set of the light sources, we evaluate all combinations of the estimated light sources in the successful paths.

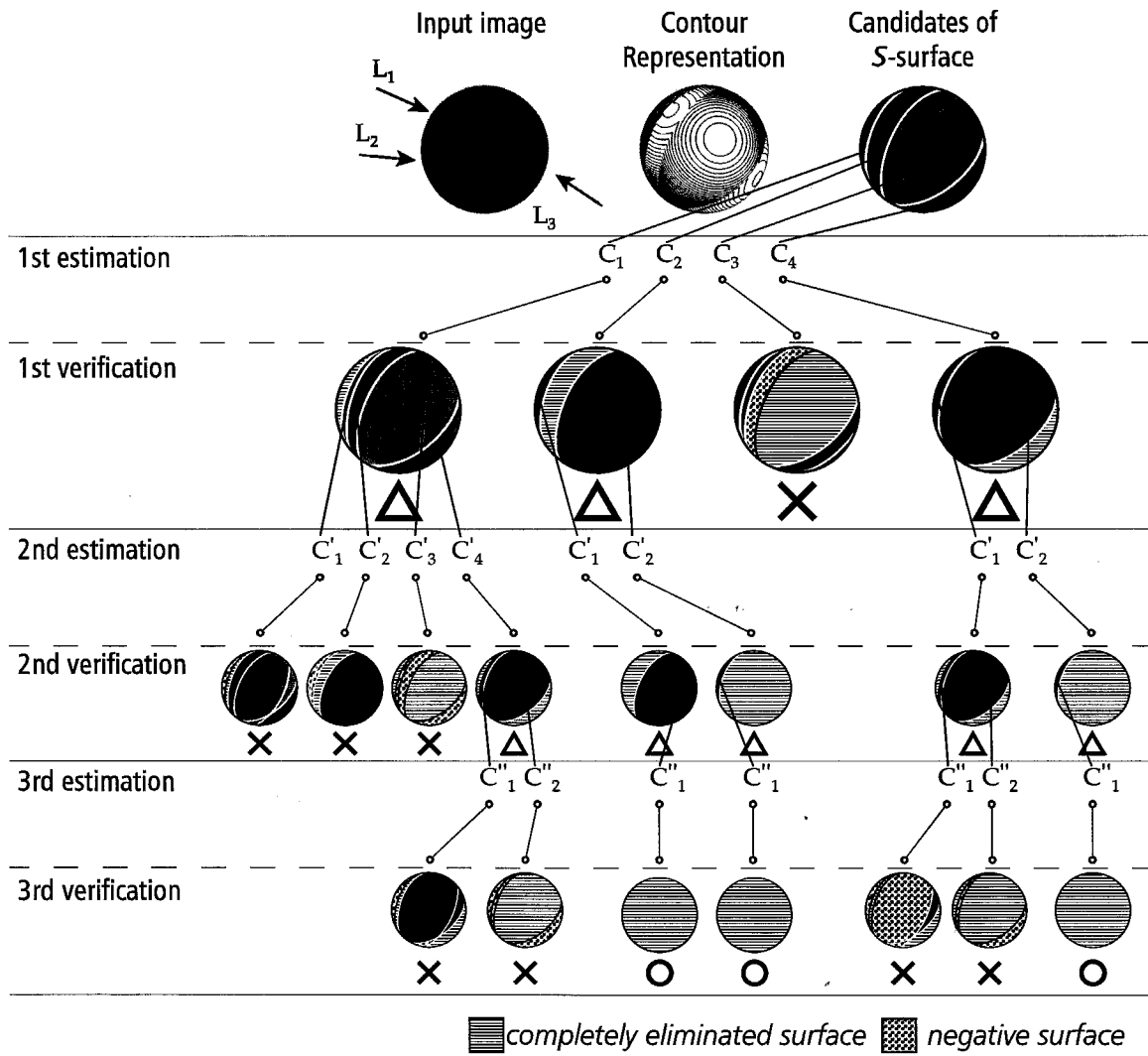


Figure 4.8: *An example of an estimation flow*

'L' denotes the light sources, and 'C' denotes S-surface candidates. ○ and × signify correct and incorrect estimations, respectively, and △ an intermediate estimation.

4.5 Performance Evaluation

We demonstrate the effectiveness of our algorithm with a CG image and a real image. The CG image is rendered with OpenGL while Gaussian noise with a variance of 6.55 is added to it.

4.5.1 CG Simulation

Figure 4.9 exemplifies some results including the contour analysis on difference sphere. The input spheres are set to be illuminated by a point light source, two directional light sources, and ambient light. Each figure shows:

- (a) Input image.
- (b) Difference sphere generated from (a).
- (c) Grouped contour segments and contour lines of the region that has positive values.
- (d) Grouped contour segments and contour lines of the region that has negative values.
- (e) Intermediate image — generated by eliminating the lighting effect by the estimated point light source from the input image.
- (f) Grouped contour segments and contour lines of (e).
- (g) Result of elimination of lighting effects.

The intermediate image (e) shows the situation where the lighting effect by the point light source is eliminated and the remaining effect by directional light sources and ambient light is present. We then consider the image as a new input and estimate the parameters of light sources with the reference spheres which we regard as two reference spheres. (f) shows that each single sphere reflects one directional light source, respectively, whereas (g) is a result of eliminating the effect by the light sources.

Table 4.1 shows the estimated parameters as the result. Light source 1 is best estimated whereas the accuracy tends to relatively decline as the estimation proceeds due to accumulation of errors. However, it can be seen that the overall performance is quite reasonable as the first trial of estimating both the positions and the radiant intensities of light sources.

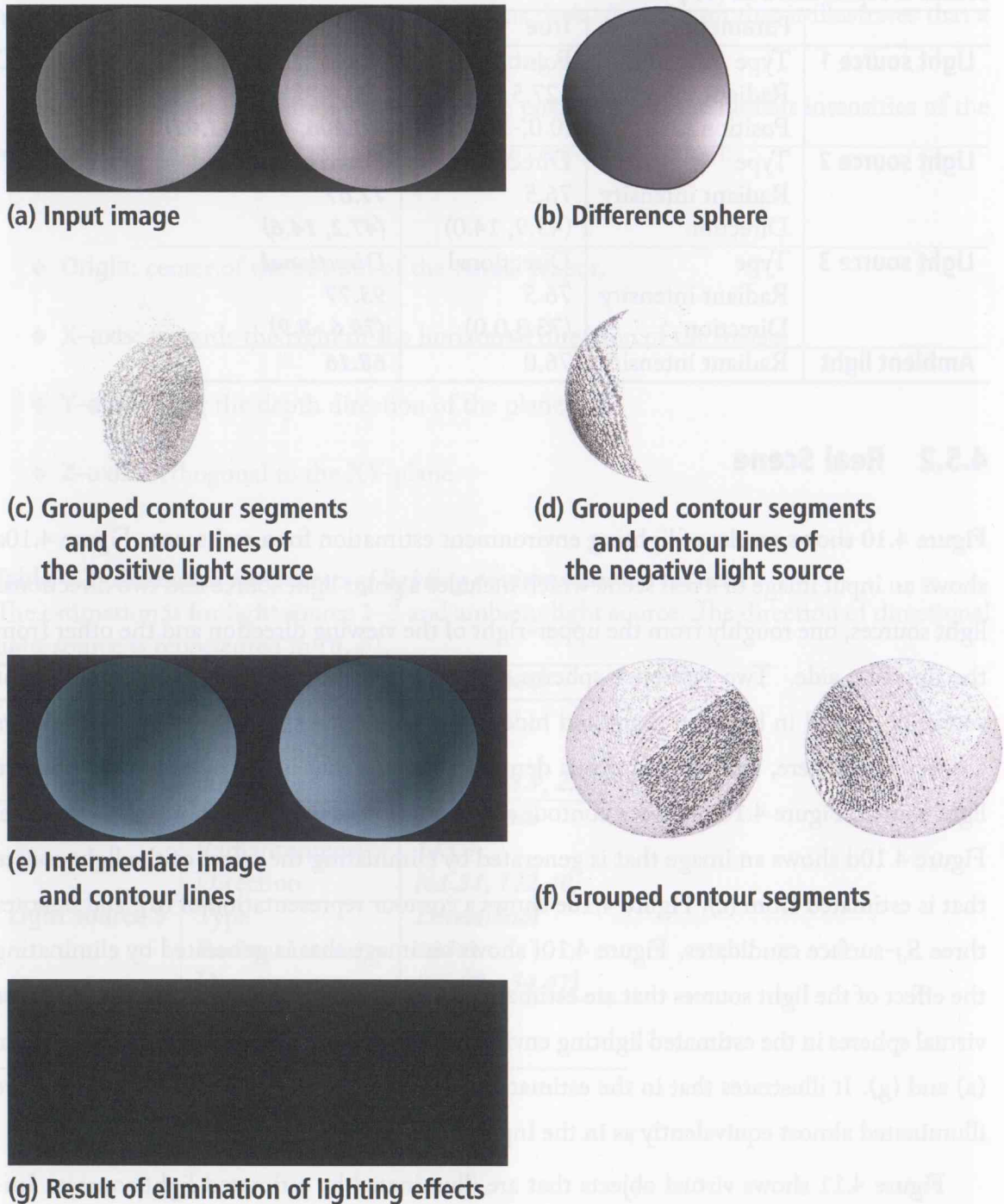


Figure 4.9: *Procedures of light source estimation.*

Table 4.1: Estimated parameters of lighting environment in CG image

The estimation is for light source 1–3 and ambient light. The direction of directional light source is represented in (θ, ϕ) .

	Parameter	True	Estimated
Light source 1	Type	Point	<i>Point</i>
	Radiant intensity	127.5	<i>120.4</i>
	Position	(0.0, -2.0, -1.0)	<i>(-0.006, -1.99, -1.01)</i>
Light source 2	Type	Directional	<i>Directional</i>
	Radiant intensity	76.5	<i>77.67</i>
	Direction	(45.9, 14.0)	<i>(47.2, 14.6)</i>
Light source 3	Type	Directional	<i>Directional</i>
	Radiant intensity	76.5	<i>93.77</i>
	Direction	(73.3, 0.0)	<i>(70.6, -8.9)</i>
Ambient light	Radiant intensity	76.0	<i>68.16</i>

4.5.2 Real Scene

Figure 4.10 shows results of lighting environment estimation for a real scene. Figure 4.10a shows an input image of a real scene which includes a point light source and two directional light sources, one roughly from the upper-right of the viewing direction and the other from the opposite side. Two reference spheres are placed on mounts whereas the point light source is located in between them and hidden by the frontal sphere. Figure 4.10b shows a difference sphere, and the red pixels denotes a surface that is illuminated by a positive light source. Figure 4.10c shows a contour representation of (b) and indicates a S_p -surface. Figure 4.10d shows an image that is generated by eliminating the effect of the light source that is estimated from (c). Figure 4.10e shows a contour representation of (d), and denotes three S_d -surface candidates. Figure 4.10f shows an image that is generated by eliminating the effect of the light sources that are estimated from (e). Furthermore, Figure 4.10g shows virtual spheres in the estimated lighting environment, and (h) shows the difference between (a) and (g). It illustrates that in the estimated lighting environment the virtual spheres are illuminated almost equivalently as in the input lighting environment.

Figure 4.11 shows virtual objects that are illuminated by estimated light sources. Images in Figure 4.11 (a), (b) and (c) are generated by illuminating a CG teapot by each of the estimated light source, respectively, whereas (d) is with all the three light sources and ambient light. Figure 4.11e shows a synthesized image by adding the virtual object and the shadows in (d) into the real scene. The point light source is now visible since the spheres

have been removed while two little real dolls are placed for comparison. As the point light source is not completely isotropic and illuminates only the upper hemisphere from a certain height, it may appear odd that the floor remains dark. Apart from that it illustrates that a CG object is added naturally with the real lighting environment.

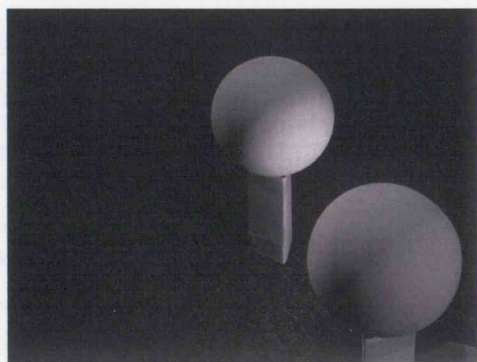
Finally, we show the estimated light source positions and the radiant intensities of the real scene (Table 4.2) under the following scene coordinate system:

- ◆ **Origin:** center of the bottom of the virtual teapot,
- ◆ **X-axis:** towards the right of the horizontal direction of the image,
- ◆ **Y-axis:** along the depth direction of the plane,
- ◆ **Z-axis:** orthogonal to the XY-plane.

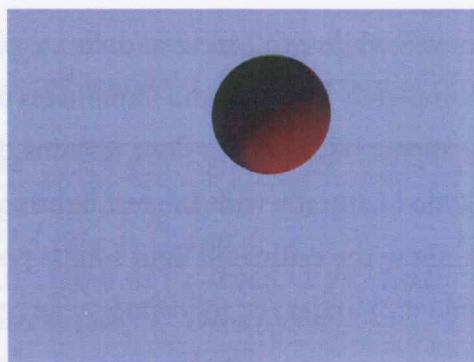
Table 4.2: *Estimated parameters of lighting environment in real scene*

The estimation is for light source 1–3 and ambient light source. The direction of directional light source is represented in (θ, ϕ) .

	Parameter	<i>estimated</i>
Light source 1	Type	<i>Point</i>
	Radiant intensity	<i>18394.0</i>
	Position	<i>(8.16, -13.9, 2.35)</i>
Light source 2	Type	<i>Directional</i>
	Radiant intensity	<i>143.3</i>
	Direction	<i>(61.31, 122.40)</i>
Light source 3	Type	<i>Directional</i>
	Radiant intensity	<i>103.0</i>
	Direction	<i>(65.80, -54.47)</i>
Light source 4	Type	<i>Ambient</i>
	Radiant intensity	<i>10.0</i>



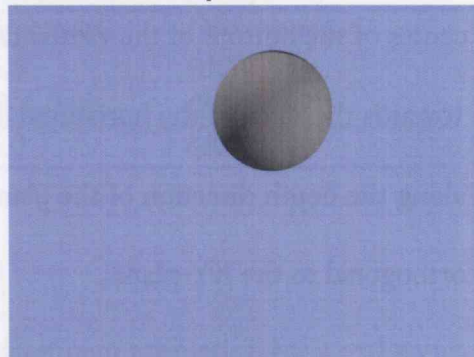
(a) Input image



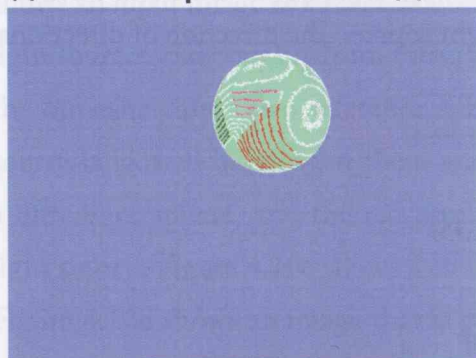
(b) Difference sphere



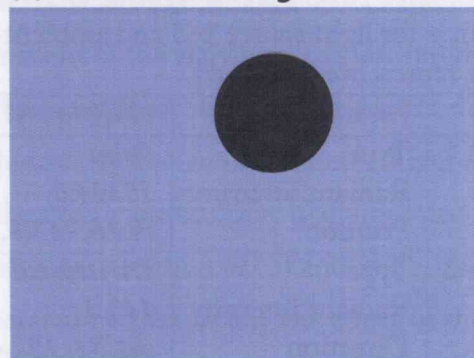
(c) Contour representation of (b)



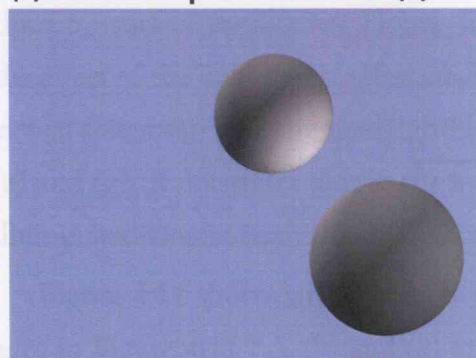
(d) Intermediate image



(e) Contour representation of (d)



(f) Result of eliminating lighting effects



(g) Virtual spheres in the estimated lighting environment

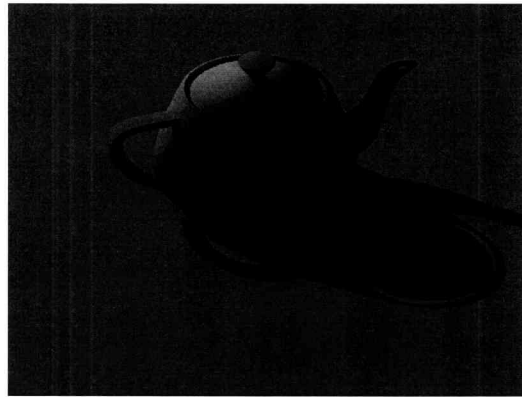


(h) Difference between (a) and (g)

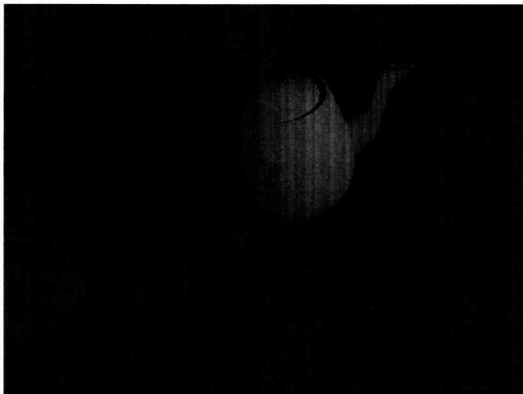
Figure 4.10: Procedure of light source estimation (Real scene)



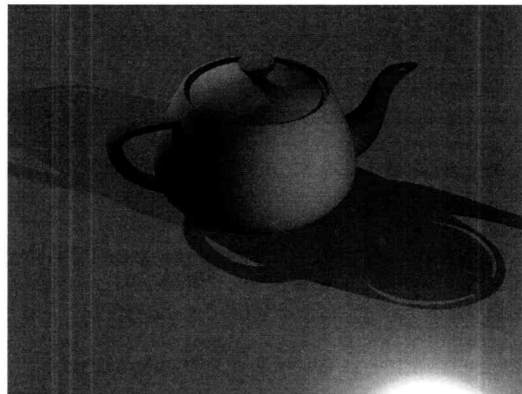
(a) Point light source



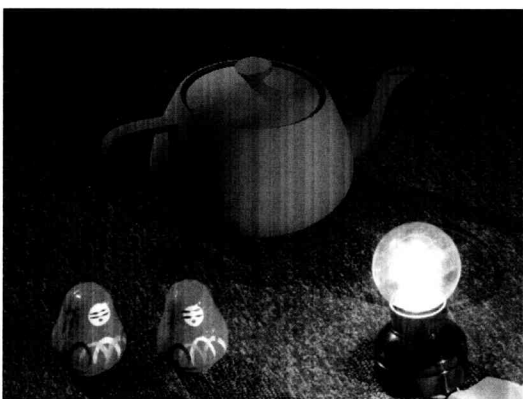
(b) Directional light source 1



(c) Directional light sources 2



(d) Estimated light sources



(e) Synthesized image

Figure 4.11: *The experiment of adding virtual object in a real scene*

Images in (a), (b) and (c) are generated by illuminating a virtual object (teapot) by each estimated light source, respectively. (d) is generated with the above three light sources and ambient light, (e) is a synthesized image in the real scene.

4.6 Discussions

4.6.1 Range of Identifying a Near Light Source

In the case of practical estimations, the range in which the point light sources can be identified depends on the size of the reference sphere and the distance between the light source and the sphere. That is, a light source that is positioned at a longer distance from the reference sphere can be regarded as a directional light source.

Figure 4.12 illustrates a reference sphere that is illuminated by a point light source, where O and P denote positions of the reference sphere and the point light source, respectively. The positions, x_p and x_d , denote the farthest points illuminated by the point light source and the directional light source that have the identical direction, l . Note that O , P , x_p and x_d are aligned on an identical plane. The arc length, l_{x_p, x_d} , between x_p and x_d is given by

$$l_{x_p, x_d} = r \cos^{-1} \frac{\sqrt{d_p^2 - r^2}}{d_p}, \quad (4.31)$$

where d_p denotes the distance between O and P , and r the radius of the reference sphere. By analyzing l_{x_p, x_d} according to d_p and r , we can determine the range in which we identify a point light source with a reference sphere.

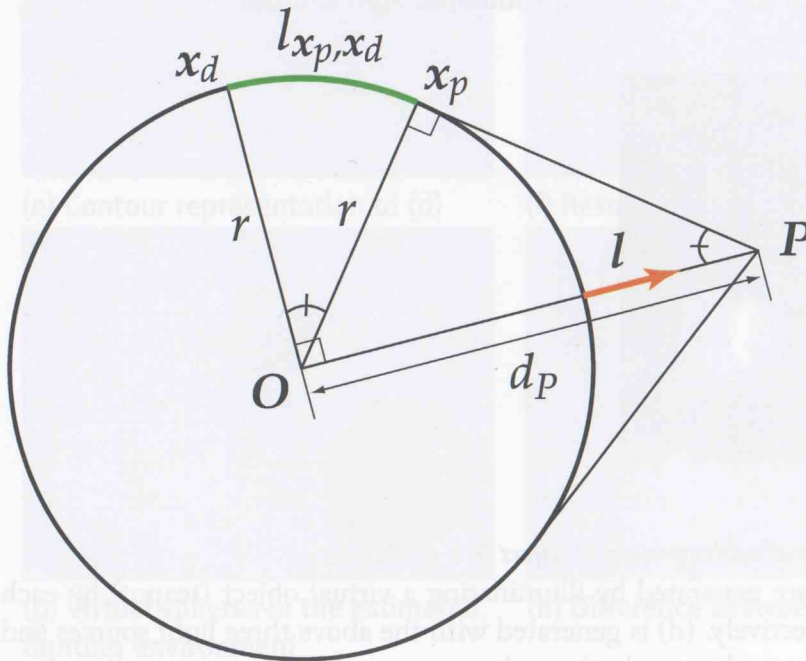


Figure 4.12: Reference sphere illuminated by a point light source

Figure 4.13 illustrates the relationship between the length, l_{x_p, x_d} , and the radius, r , of a reference sphere in the case that the reference sphere has the radius of the size of 1. The figure indicates that l_{x_p, x_d} becomes 0.1 at about ten of d_p . Considering the resolution of captured images, the accuracy of camera calibration, the shape of reference spheres, etc., we reach to a criterion for the range of identifying a near light source, d_p , as ten times of r . E.g., when utilizing a reference sphere whose radius is 5 cm, the range becomes 50 cm. In other words, we regard the light sources that are positioned at ten times farther than the radius of the reference sphere as directional light sources. Accordingly, we can eliminate the effects of the light sources using the difference sphere, and estimate the radiant intensity of near light sources.

By determining the range of identifying near light sources as in the above descriptions, we can explicitly distinguish the foreground lighting from the background lighting. We can therefore estimate precise parameters of near light sources with the difference sphere, and the complex lighting environment with the skeleton cube as we described in the introduction of the lighting environment estimation.

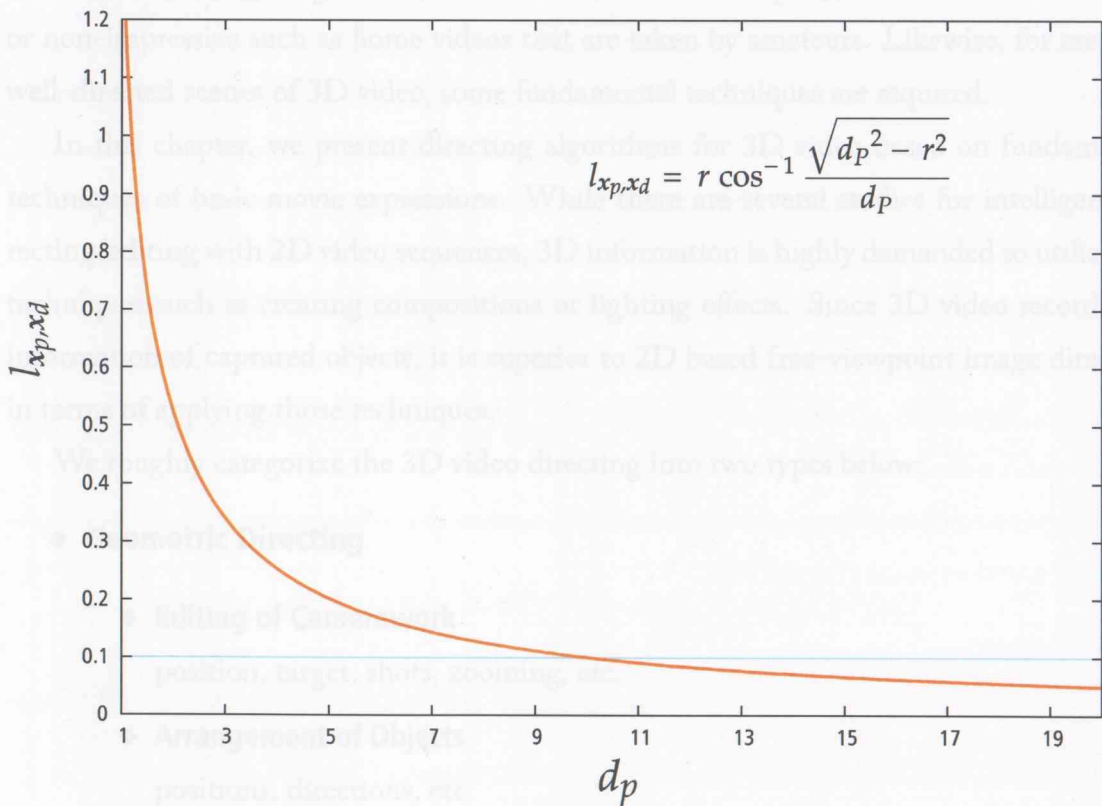


Figure 4.13: Relationship between l_{x_p, x_d} and d_p ($r = 1$)

4.6.2 Summary

We have presented a novel technique for lighting environment estimation which allows us to estimate parameters of point light sources as well as directional light sources and ambient light. We employ a contour representation of an image for analyzing shading of a pair of reference spheres and characterize 3D geometric and photometric properties of the light source. In particular we have proposed the difference sphere which enables us to estimate parameters of the point light sources by eliminating the lighting effects of ambient light and directional light sources. In order to show the theoretical availability, we have demonstrated the effectiveness by applying our method to CG images generated with a point light source, two directional light sources, and ambient light. We have also applied our method to lighting environment of a real scene, and demonstrated that our method enables us to render a virtual image that is geometrically and photometrically consistent to the real scene by synthesizing the image in the scene.

Chapter 5

Versatile Visualization of 3D Video

5.1 Introduction

Movie directors create impressive scenes by directing actions of actors, camerawork, lighting, etc. While many of these expressions are based on experience and sensibility of the directors, some expressions are established as fundamental techniques for creating compositions of scenes, lighting effects, etc. Without those techniques, created movies are boring or non-impressive such as home videos that are taken by amateurs. Likewise, for creating well-directed scenes of 3D video, some fundamental techniques are required.

In this chapter, we present directing algorithms for 3D video based on fundamental techniques of basic movie expressions. While there are several studies for intelligent directing/editing with 2D video sequences, 3D information is highly demanded to utilize the techniques such as creating compositions or lighting effects. Since 3D video records 3D information of captured objects, it is superior to 2D based free-viewpoint image directing in terms of applying those techniques.

We roughly categorize the 3D video directing into two types below:

◆ Geometric Directing

❖ Editing of Camerawork

position, target, shots, zooming, etc.

❖ Arrangement of Objects

positions, directions, etc.

❖ Arrangement of Background

positions, directions, etc.

◆ Photometric Directing

❖ Lighting

type of light sources, positions, directions, intensity, distribution, etc.

We describe the details of each directing in the following sections.

Furthermore, we have developed a 3D video viewer that plays 3D video in real time. This viewer enables us to observe a 3D video object from an arbitrary viewpoint, and moreover, it can render stereoscopic images for special displaying equipment [Sha, San] that can display eye-popping 3D images. With this viewer, users can enjoy the world of 3D video on real meaning.

5.2 Geometric Directing: Camerawork Creation using Object's Shape and Motion

5.2.1 Overview

Composition of the scene—which direction, how much size, and what position an object is captured—gives various impressions to audiences, even though it depicts the same scene. Hence movie directors arrange objects and cameras carefully to visualize what they desire to express.

In the case of ordinary film directing, directors can order actors to change their actions, positions, and directions, but in the case of 3D video directing, it is difficult to change shapes and motions of the captured object, or they must not be changed because they may have special meaning such as traditional dancing, ceremonies, action of athletes, etc. For this reason, an algorithm of camerawork creation that copes with the object's shape and motion is required for directing 3D video.

For geometric directing—i.e. camerawork creation, we arrange the following components in a virtual scene (Figure 5.1).

◆ Object(s)

It is captured by 3D video capturing system. We can also duplicate the object and arrange them in a same scene.

◆ Background

A hemispherical object that is texture-mapped with an image that is captured by an

omni-directional camera.

◆ Virtual camera

It captures the virtual scene by specifying some parameters.

These components have the following parameters that are defined in each coordinate system.

◆ Object parameters

Position: x_o, y_o, z_o , direction: ψ_o, ϕ_o, ρ_o , and scale: s_o

◆ Background parameters

Position: x_b, y_b, z_b , direction: ψ_b, ϕ_b, ρ_b , and scale: s_b

◆ Virtual camera parameters

Position: x_p, y_p, z_p , target point: x_t, y_t, z_t , up direction: x_u, y_u, z_u , field of view: θ , and aspect ratio: r

To create camerawork for actualizing required expressions, these parameters are required to be specified at each time. The number of parameters is, however, too numerous to specify and it is time consuming for 3D video directors. In the following section, we present a semi-automatic algorithm to create camerawork for fundamental expressions by specifying compositions of a scene (e.g. close up, middle shot, etc.) in some key frames.

5.2.2 Algorithm of Camerawork Creation

5.2.2.1 Outline

We show the outline of the procedure in the following:

1. Specify the composition of the scene in the first frame (*key frame 0*).
2. Calculate the camera parameters that actualize the composition.
3. Specify the composition of the scene in an arbitrary frame (*key frame i*).
4. Calculate the camera parameters that actualize the composition and interpolate the parameters between key frame i and key frame $i - 1$.
5. When reach the last frame, finish the procedure, otherwise go to 3.

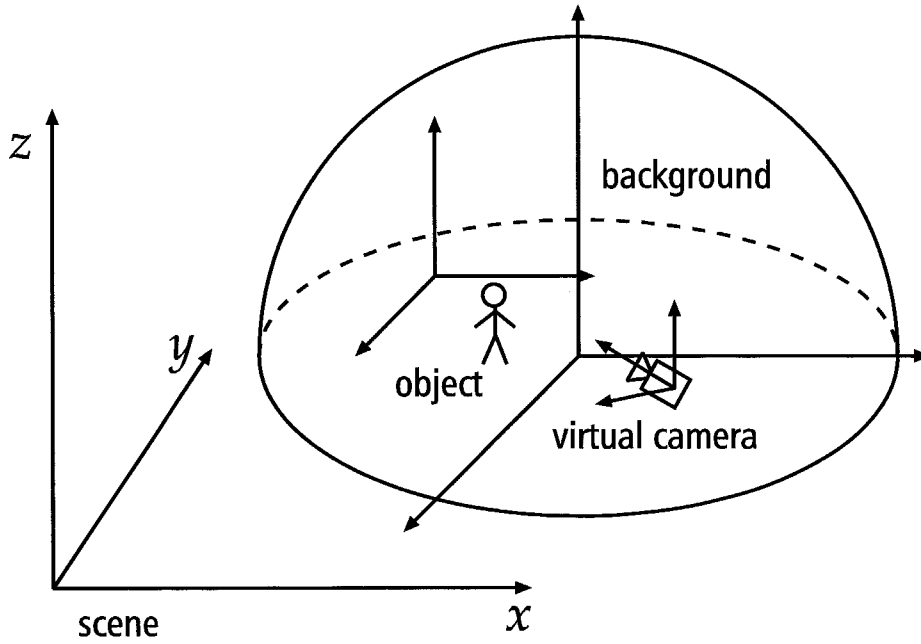


Figure 5.1: *Virtual Scene Setup*

We define the following parameters of an object before describing calculation of camera parameters.

- ◆ **Position:** the centroid of a group of voxels that form the object.
- ◆ **Height:** length of the object along the z axis.
- ◆ **Direction:** direction representing the front part of the object. (This can be estimated from the shape of the object. In this thesis, we specify this beforehand.)

Note that these parameters are defined assuming a standing human as the object.

5.2.2.2 Camera Parameter Calculation

Camera parameters to be calculated by our algorithm are position, target point and up direction. Other parameters—i.e. field of view and aspect ratio are specified with composition of the scene. Furthermore, by specifying the direction to capture an object (e.g. front, back, etc.), we can obtain a complete set of camera parameters that actualize the composition.

First, before calculating the position of the camera, we define the distance d between the camera and the object by the following equation:

$$d = \frac{r_h \cdot h}{2 \tan \frac{\theta}{2}}, \quad (5.1)$$

5.2. Geometric Directing: Camerawork Creation using Object's Shape and Motion 97

where h is the height of the object, and θ is the field of view. r_h is the ratio of the object's height in a picture to the height of the picture, which is defined as following:

- ◆ $r_h = 1.0$: full shot,
- ◆ $r_h = 0.7$: knee shot,
- ◆ $r_h = 0.5$: middle shot,
- ◆ $r_h = 0.3$: close shot, and
- ◆ $r_h = 0.2$: close up.

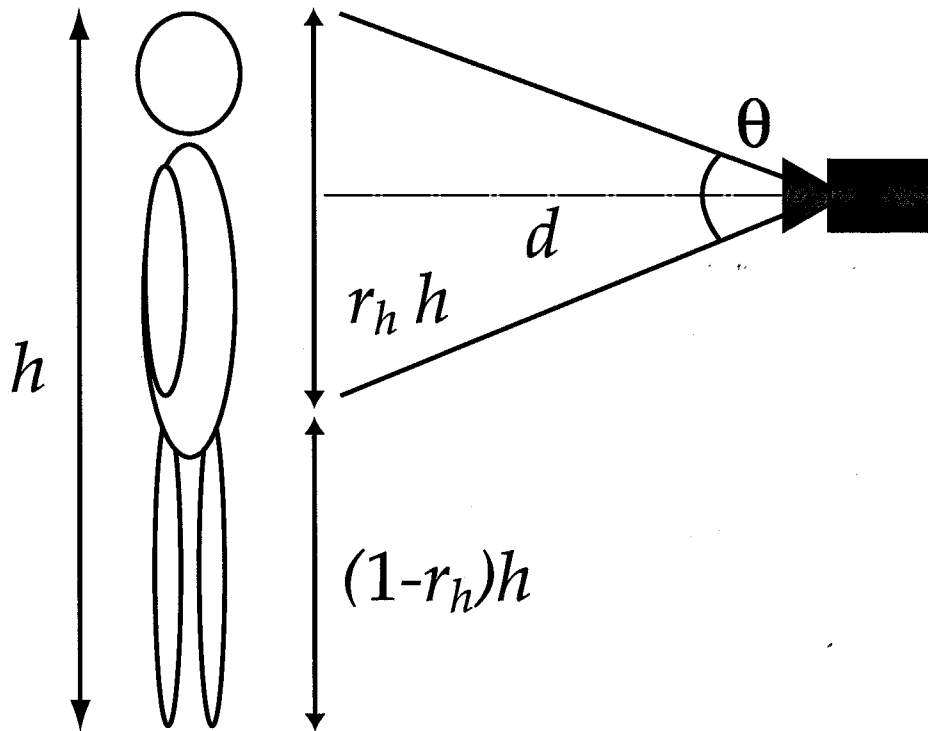


Figure 5.2: Relationship between a virtual camera and an object (I)

After obtaining the distance d , we determine the position of the camera (x_p, y_p, z_p) by specifying the direction to capture the object. Additionally, we determine the target point by the following equations:

$$\begin{cases} x_p = d \cos \phi \cos(\psi + \delta) + x_t \\ y_p = d \cos \phi \sin(\psi + \delta) + y_t \\ z_p = d \sin \phi + z_t \end{cases}, \quad (5.2)$$

where d is given by Equation(5.1), and δ is the angle between the direction of the object and the x axis in the scene (Figure 5.3). ϕ and ψ are arbitrarily specified, such as upper-front, lower-back, etc. (Figure 5.4). (x_t, y_t, z_t) is the target point, where x_t and y_t denote the same value as the position of the object, and z_t is given by

$$z_t = h - \frac{h \cdot r_h}{2} \quad (0 \leq r_h \leq 1). \quad (5.3)$$

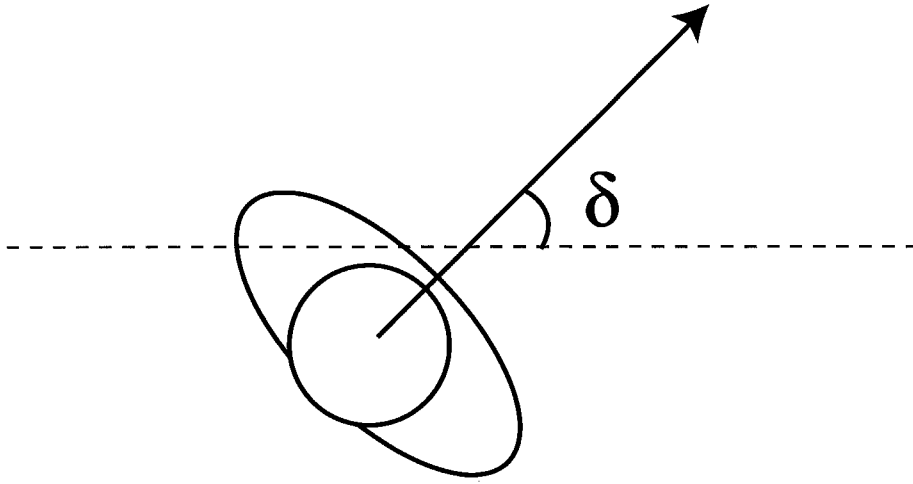


Figure 5.3: Relationship between object and scene coordinate systems

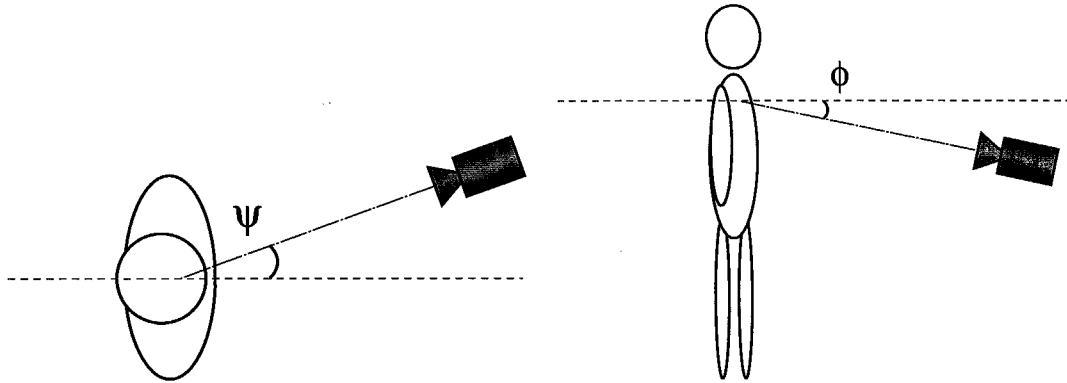


Figure 5.4: Relationship between a virtual camera and an object (II)

Finally, determining the up direction from the cross product of viewing direction (the direction from the camera position to the target point) and the horizontal direction of the scene, we can obtain a complete set of camera parameters.

After calculating the camera parameters in arbitrary key frames, we can obtain camerawork of a 3D video sequence by interpolating them between the neighboring key frames.

5.2.3 GUI Application for Geometric Directing

We have developed a GUI application for the geometric directing (Figure 5.5). A 3D video director can specify the positions, rotations, and scales of an object and the background with this application. Camera parameters are calculated from the composition of the scene that is specified in a key frame.

We have also implemented a method to create camerawork by specifying all of the camera parameters in arbitrary key frames, which is often employed by ordinary computer graphics softwares. With this method, a 3D video director can create various scenes that he/she desires to express.

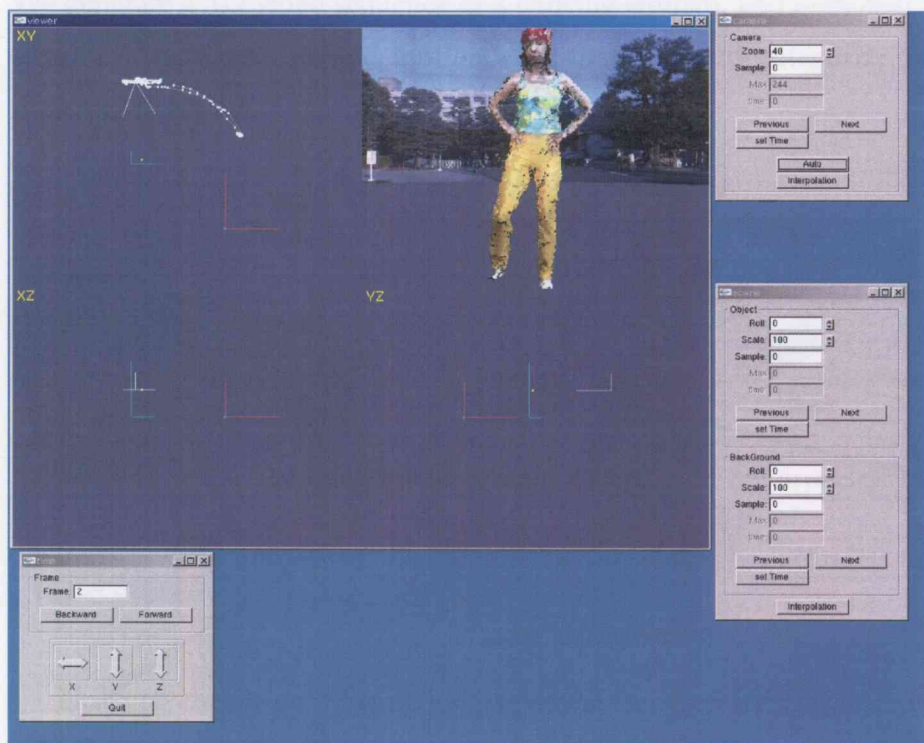


Figure 5.5: GUI application for 3D video editing and visualization

5.2.4 Performance Evaluation

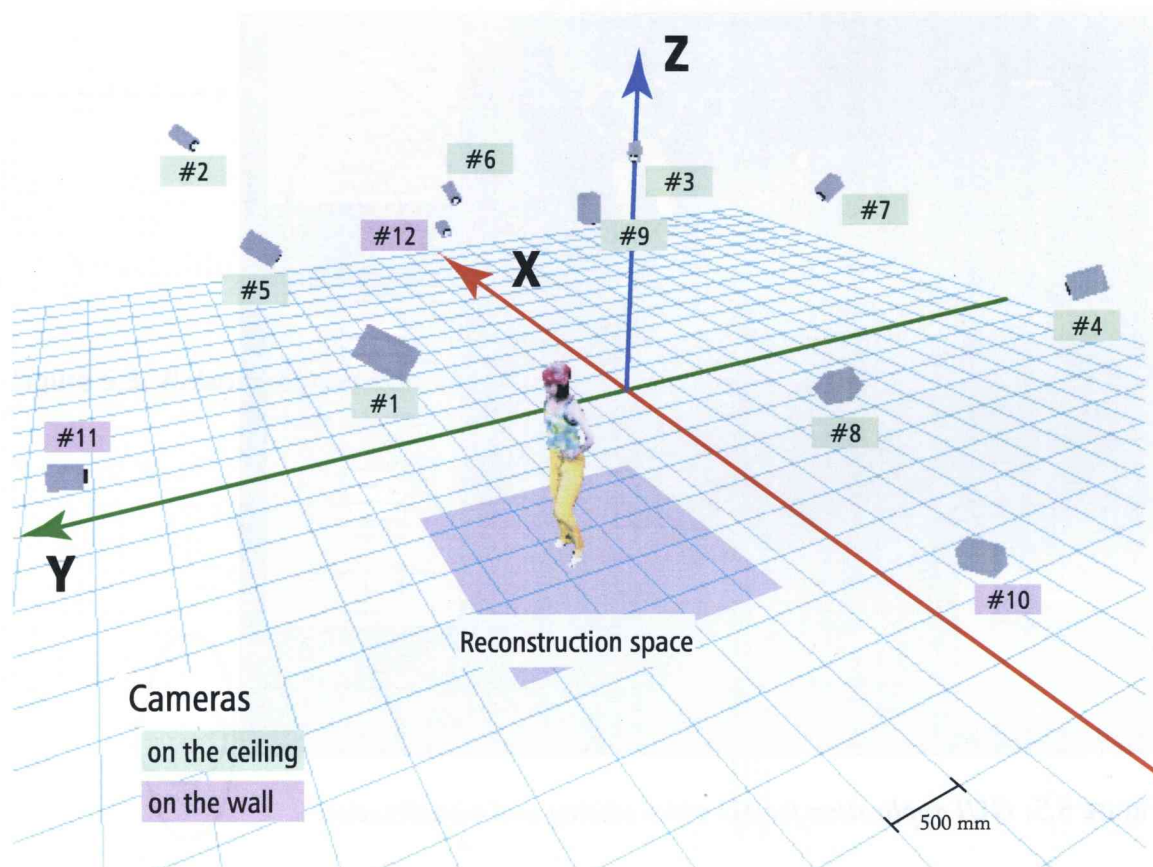
We show experimental results with an object that is captured by the following conditions (Figure 5.6):

- ◆ Number of onjects: 1
- ◆ Camera: Sony EVI G20 (Table 5.1)

- ◆ **Number of cameras:** #1 to #9 for shape reconstruction, and #1 to #12 for texture generation
- ◆ **Length of a sequence:** 334 frames (15 frames per second)

Table 5.1: *Specifications of Sony EVI G20*

Type of Camera	Analog video camera
Image Sensor	1/4-type color CCD
Image Size	640 × 480 pixel
Data Format	NTSC color
Frame rate	15 frames per second, software synchronization

Figure 5.6: *3D video capturing studio 1*

5.2.4.1 Composition

We first show a result of rendered scenes by specifying compositions of a scene at a single frame (Figure 5.7). The images indicate that the scenes are rendered with appropriate

compositions using geometric information of the object.



Figure 5.7: Images rendered by specifying a composition of a scene

5.2.4.2 Camerawork Creation using Object’s Shape and Motion

We generate a 3D video sequence of a dancing woman, and demonstrate the camerawork creation using its shape and motion. We first manually categorize its motion, which is shown in Table 5.2.

Table 5.2: Object’s motion

Frame Number	Motion
0–33	moving to right
34–76	swing her arms left to right
77–119	moving to left
120–159	swinging her arms left to right
160–194	moving to right
195–237	changing the direction and swinging her arms left to right
238–268	moving to left
269–314	changing the direction and swinging her arms left to right
315–333	raise her right hand

Utilizing the above information, we create camerawork that realizes capturing the object’s front part while the object is dancing. The composition is specified as shown in table 5.3, and the field of view and the aspect ratio are specified as 60 degree and 1.33(=640/480), respectively.

Table 5.3: Specified compositions

Frame Number	Composition
0	full shot
119	full shot
237	knee shot
299	middle shot
333	full shot

Figure 5.8 shows a final result of the geometric directing—i.e., camerawork creation using object's shape and motion. We also utilize an omni-directional background that is captured by a fixed-viewpoint pan-tilt camera [WM96]. The figure shows that the sequence is correctly rendered by the specified camerawork, and indicates that we can direct a 3D video sequence effectively by utilizing the geometric information of the object.



Figure 5.8: 3D video sequence directed using object's shape and motion

5.3 Photometric Directing: Lighting Effect Control

5.3.1 Overview

Lighting is also an important expression for direction of movies. Since 3D video has both geometric and photometric information, we can actualize the lighting effect on an object by analyzing its shape and reflectance parameters. It is one of the advantages for 3D video over ordinary image-based rendering methods.

To control lighting effects, we require the lighting environment, the shape of an object, and reflectance parameters of the object. In the following sections, we describe the details of these parameters, and a method to actualize the lighting effect control with them.

5.3.2 Estimation of Lighting Environment

While we can estimate the lighting environment with the skeleton cube or the difference sphere independently as described in the previous chapters, we can also estimate the lighting environment effectively by combining the two methods. That is, by compensating for the weakness of each method—the skeleton cube: approximate estimation of the distribution of light sources and the difference sphere: difficulties for estimating complex lighting environment—we can estimate a real, complex lighting environment effectively.

The procedure of the combination of the skeleton cube and the difference sphere is the following (Figure 5.9):

1. Model an approximate lighting environment (Figure 5.9b)
2. With the difference sphere, estimate precise parameters of the foreground lighting (Figure 5.9c)
3. With the skeleton cube, estimate radiant intensity of the lighting environment that is remodeled with the estimated foreground lighting (Figure 5.9d)

The key point of the procedure is that by eliminating the effects of the background lighting, we can estimate precise parameters of the foreground lighting with the difference sphere, and a complex lighting environment is then modeled and estimated with the skeleton cube by utilizing the estimation result of the difference sphere. Accordingly, we can estimate a lighting environment of real scenes effectively.

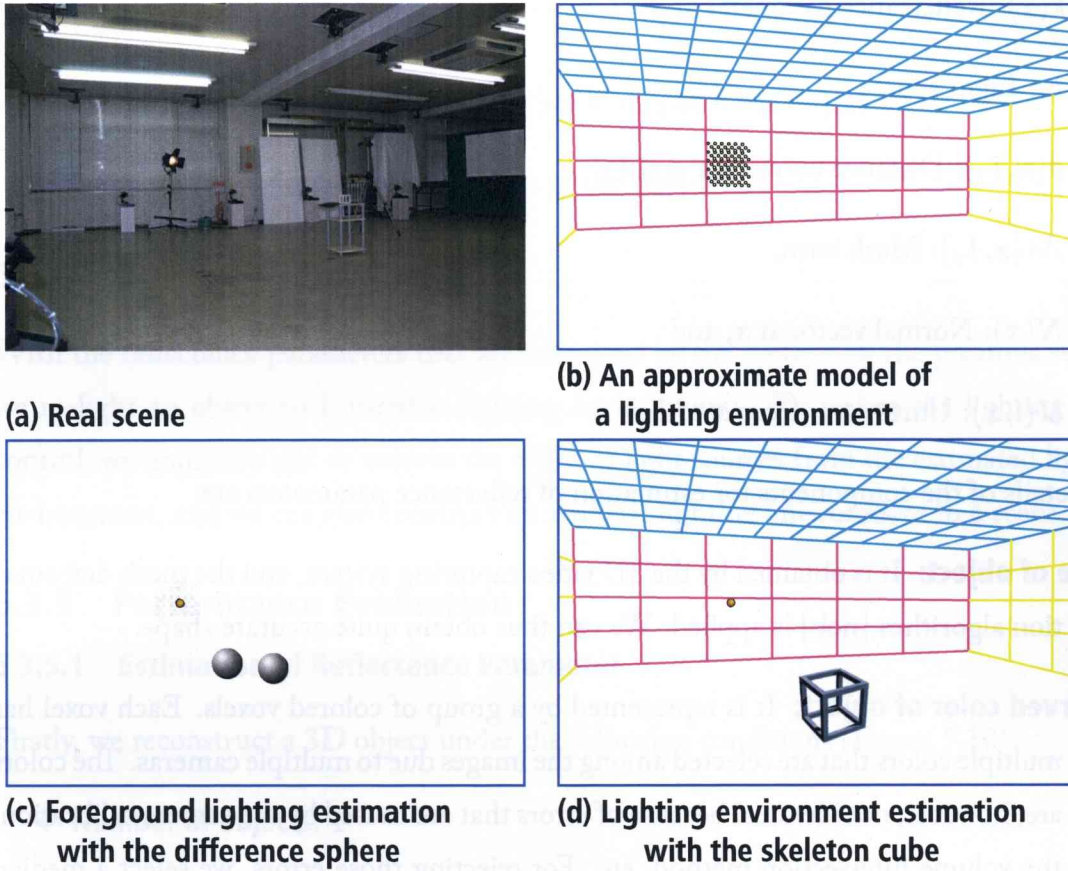


Figure 5.9: *Combination of the skeleton cube and the difference sphere*

5.3.3 Estimation of Reflectance Parameters

Observed color of a 3D video object is determined by the viewpoint, parameters of the object, and lighting environment. That is, it is given by

$$I(\mathbf{x}) = f(V, O(\mathbf{x}), L), \quad (5.4)$$

where $\mathbf{x}$ denotes the surface point of the object, $I(\mathbf{x})$, V , $O(\mathbf{x})$, and L denote the observed color at $\mathbf{x}$, the viewpoint, the parameters of the object—i.e. the normal vector and the reflectance parameter at $\mathbf{x}$, and the parameters of lighting environment, respectively¹.

Assuming the surface reflection model as Lambertian, we define the observed color of the object as

$$I(\mathbf{x}) = k(\mathbf{x}) \sum_{i=1}^n \frac{\mathcal{M}(\mathbf{x}, L_i)}{d(\mathbf{x}, L_i)^2} N(\mathbf{x}) \cdot \mathbf{U}(\mathbf{x}, L_i), \quad (5.5)$$

where each parameter is defined as following:

¹ These parameters are defined in each color band, that is, R, G, and B. For the sake of a simple description, we do not denote it explicitly in the following descriptions.

- ◆ $k(\mathbf{x})$: Reflectance parameter at $\mathbf{x}$,
- ◆ L_i : Light source i ,
- ◆ $d(\mathbf{x}, L_i)$: Distance between $\mathbf{x}$ and L_i ,
- ◆ $\mathcal{M}(\mathbf{x}, L_i)$: Mask term,
- ◆ $N(\mathbf{x})$: Normal vector at $\mathbf{x}$, and
- ◆ $U(i, \mathbf{x})$: Unit vector from $\mathbf{x}$ to L_i .

The details of the components for estimation of reflectance parameters are:

Shape of object: It is obtained by the 3D video capturing system, and the mesh deformation algorithm [nob] is applied. We can thus obtain quite accurate shape.

Observed color of object: It is represented by a group of colored voxels. Each voxel has multiple colors that are selected among the images due to multiple cameras. The colors are, however, not identical because of errors that are caused by the camera calibration, the volume intersection method, etc. For rejecting those errors, we select a median color of them, and define it as the color of the voxel.

Lighting environment: It is estimated by the methods that are described in the previous sections. The lighting environment is represented by a set of light sources that have the parameters for the type, the position, the direction and the intensity.

Mask term: It is calculated by the method that is described in Section 3.3.2.2, which indicates 0 or 1 for point light sources, and 0 to 1 for area light sources.

According to the above description, if the shape and the observed color of the object and the lighting environment are known, we can estimate the reflectance parameter by

$$k(\mathbf{x}) = \frac{I(\mathbf{x})}{\sum_i \frac{\mathcal{M}(\mathbf{x}, L_i)}{d(\mathbf{x}, L_i)^2} N(\mathbf{x}) \cdot U(\mathbf{x}, L_i)}. \quad (5.6)$$

Consequently, the procedure for the estimation of the reflectance parameters is the following:

1. Reconstruct the object

2. Estimate the color of the object
3. Estimate the lighting environment that illuminates the object
4. Estimate the reflectance parameters of the object

5.3.4 Lighting Effect Control

With the reflectance parameters that are estimated by the method in the previous section, we re-light an object with variable lighting environments. To realize the lighting effect control, we manually add or remove the effect of light sources from the estimated lighting environment, and we can also construct an artificial lighting environment in a scene.

5.3.5 Performance Evaluation

5.3.5.1 Estimation of Reflectance Parameter

Firstly, we reconstruct a 3D object under the following conditions (Figure 5.10):

- ◆ **Number of objects:** 1
- ◆ **Camera:** Sony DFW-VL500 (Table 5.4)
- ◆ **Number of cameras:** 25
- ◆ **Reconstruction Space:** $3000 \times 3000 \times 2000$ mm
- ◆ **Lighting:** 14 fluorescent light tubes and one spot light (The details are described in Section 3.6.2.2.) In this section, we only employ the skeleton cube for lighting environment estimation.

Table 5.4: Specifications of Sony DFW-VL500 [Son01]

Type of Camera	Digital video camera
Image Sensor	1/3-type progressive scan IT CCD
Image Size	640×480 pixel
Data Format	YUV (4:2:2) 8 bits each
Frame rate	12.5 fps at external synchronous mode

The captured images of an object (a human in a studio) and sample images of a reconstructed object are shown in Figure 5.11 and Figure 5.12, respectively. The object has 19220 vertices

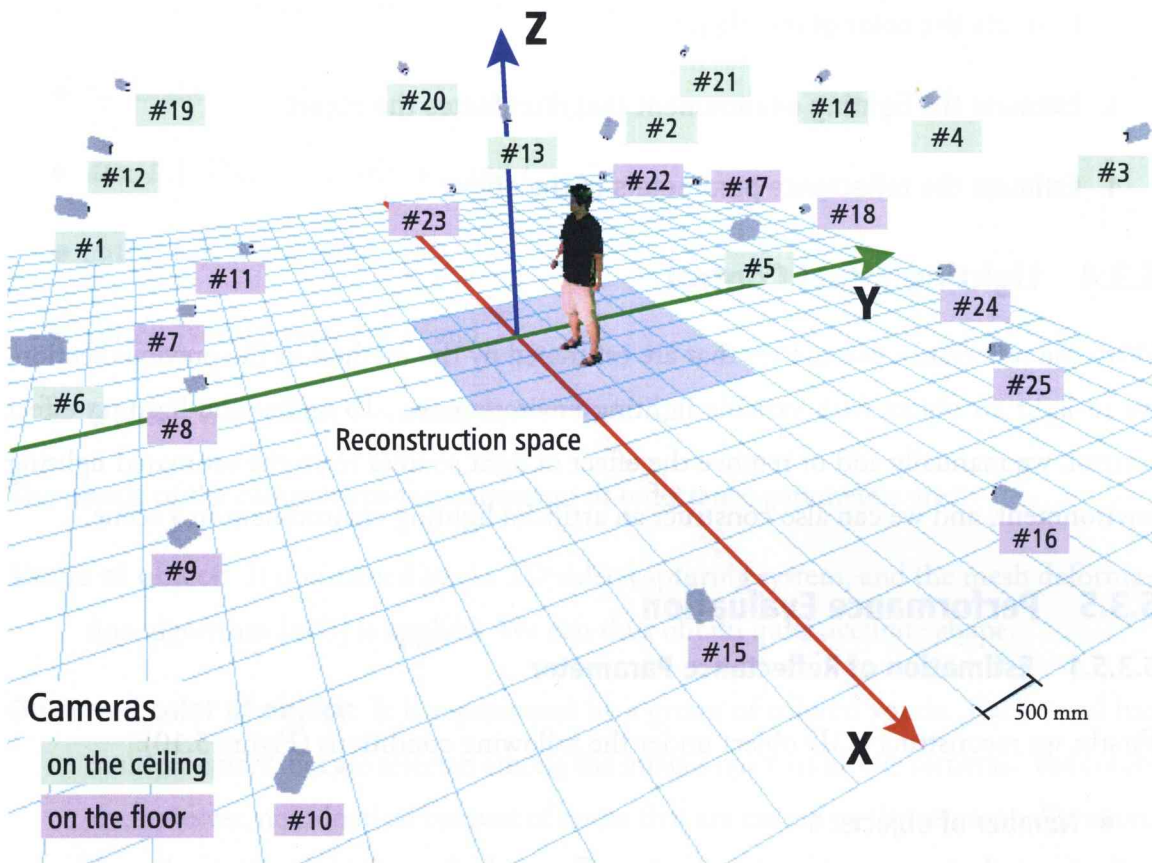


Figure 5.10: 3D video capturing studio 2

and 38436 faces in the voxel size of $10 \times 10 \times 10$ mm. Figure 5.12 shows that the resolution is sufficient for expressing the face of the object, but some errors occur especially on the left side of the object because of the calibration error.

Secondly, we calculate the median color of the object (Figure 5.13). The figure shows that the errors are reduced but some of the details are lost by the median filtering. Observing the entire object, we can obtain acceptable data for estimating the reflectance parameters.

Thirdly, we estimate the lighting environment with the skeleton cube (The details are described in Section 3.6.2.2).

Finally, we estimate the reflectance parameter with the components that are described above. Figure 5.14 shows sample images of the object that is colored with the estimated parameters, which are normalized to 0–255 in each color band for a display reason.

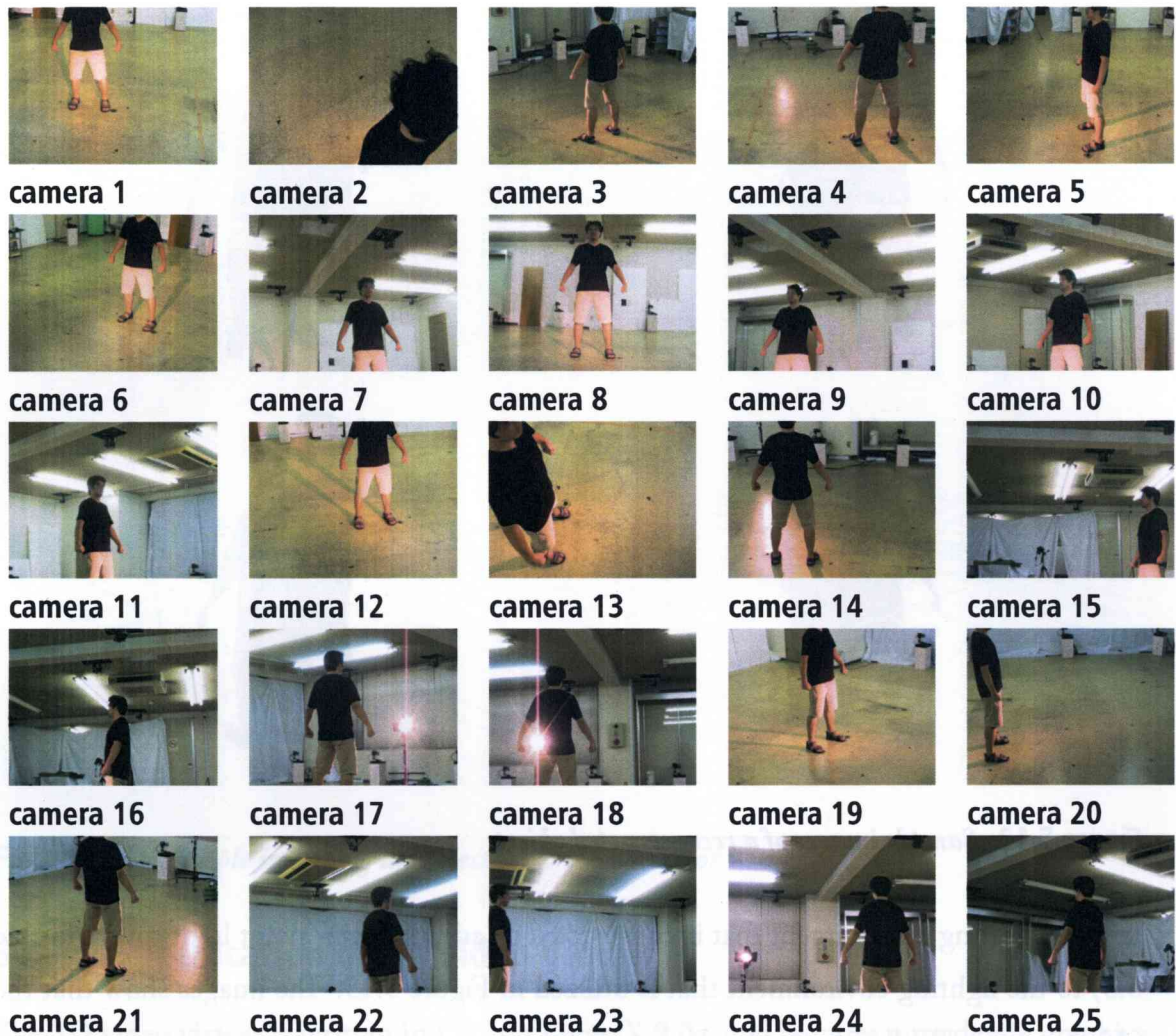


Figure 5.11: *Captured images for reconstruction of an object*

5.3.5.2 Elimination and Addition of Lighting Effects

We show two experimental results of lighting effect control; (i) elimination of the effect of light sources from the estimated lighting environment, and (ii) addition of the effect of light sources to the lighting environment.

For eliminating lighting effect, we remove light sources that are supposed to represent the spot light from the lighting environment that is estimated in the previous section. Figure 5.15 shows images that are rendered with the new lighting environment. These images demonstrate that the effect of the spot light is perfectly eliminated, which enables us to observe the object with an artificial lighting environment.

Figure 5.16 shows another result of the lighting effect control. The images are rendered

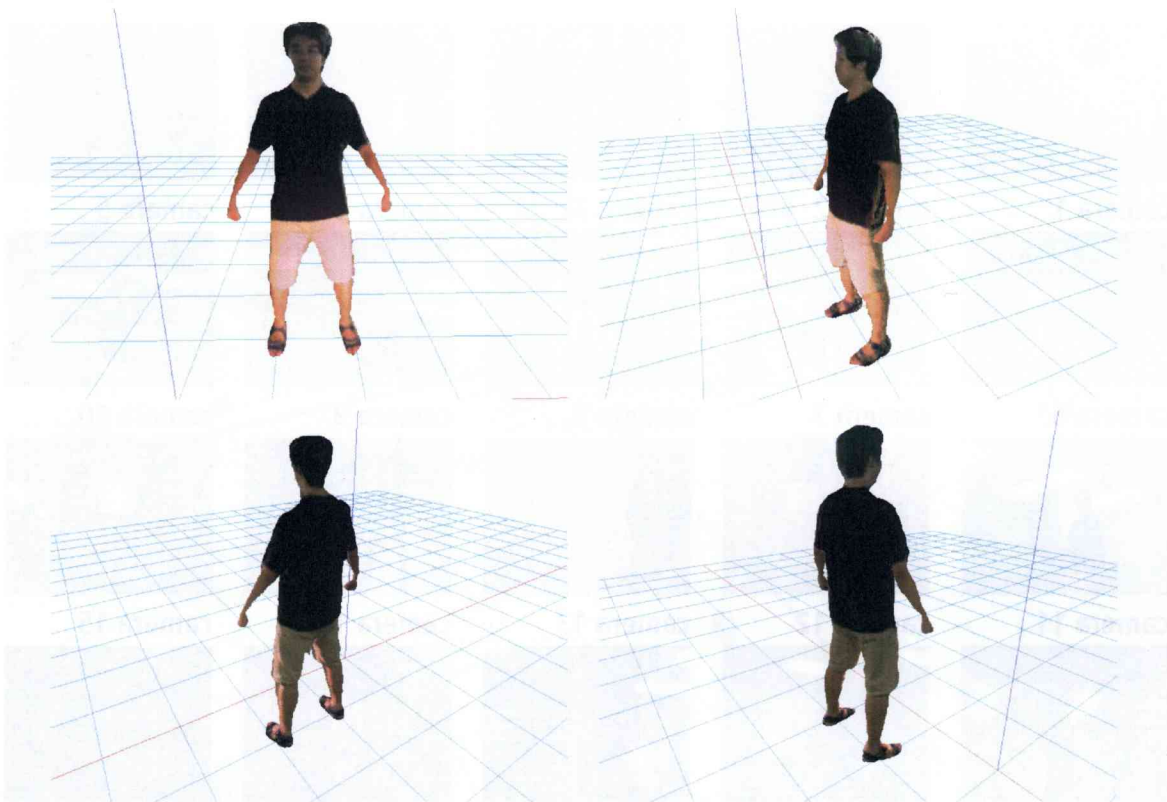


Figure 5.12: *Sample images of a reconstructed object*

under a lighting environment that is constructed by adding a new point light source (Table 5.5) to the lighting environment that is utilized in Figure 5.15. The images show that the object is naturally illuminated by the new light source.

Table 5.5: *Parameters of a new point light source*

Position: (x, y, z)	(800, -1500, 1000)
Radiant Intensity: (R, G, B)	(0.2, 1.0, 0.5)

Throughout the experiments we have acquired rich expressions of the scene by effectively integrating the geometric and photometric information, and shown the advantage of our propositions over ordinary image based rendering methods.

5.4 3D Video Viewer

We have developed an application for viewing 3D video—i.e., the 3D video viewer, which can render a sequence of 3D video data in real time. In this section, we describe the specification of the 3D video viewer and file formats for the 3D video object.

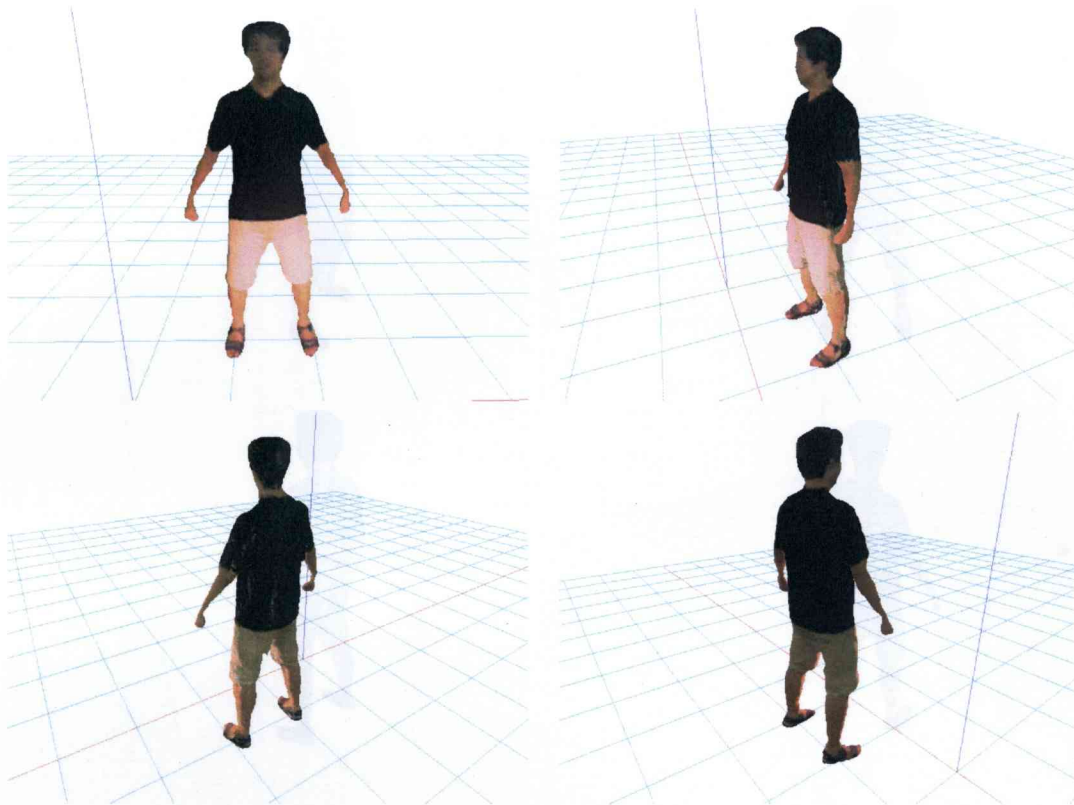


Figure 5.13: *Sample images of an object with median color*

5.4.1 Specification of 3D Video Viewer

The 3D video viewer is written in C# and DirectX 9.0c, and requires a moderate graphics processor unit. The 3D video viewer renders an object by changing viewpoint, gazing point and zoom arbitrarily with mouse operations, and it can also realize the camerawork that is generated by the GUI described in Section 5.2.3. The viewer features:

- ◆ Real-time rendering with the viewpoint dependent vertex-based method
- ◆ Real-time rendering with the viewpoint independent patch-based method
- ◆ Superimposing an omni-directional background

Furthermore, the 3D video viewer can render stereo images as input data for the following two types of displays:

- ◆ **Sanyo method:** renders the image for left eye in the left half of the display, and for right eye in the right (Figure 5.17).

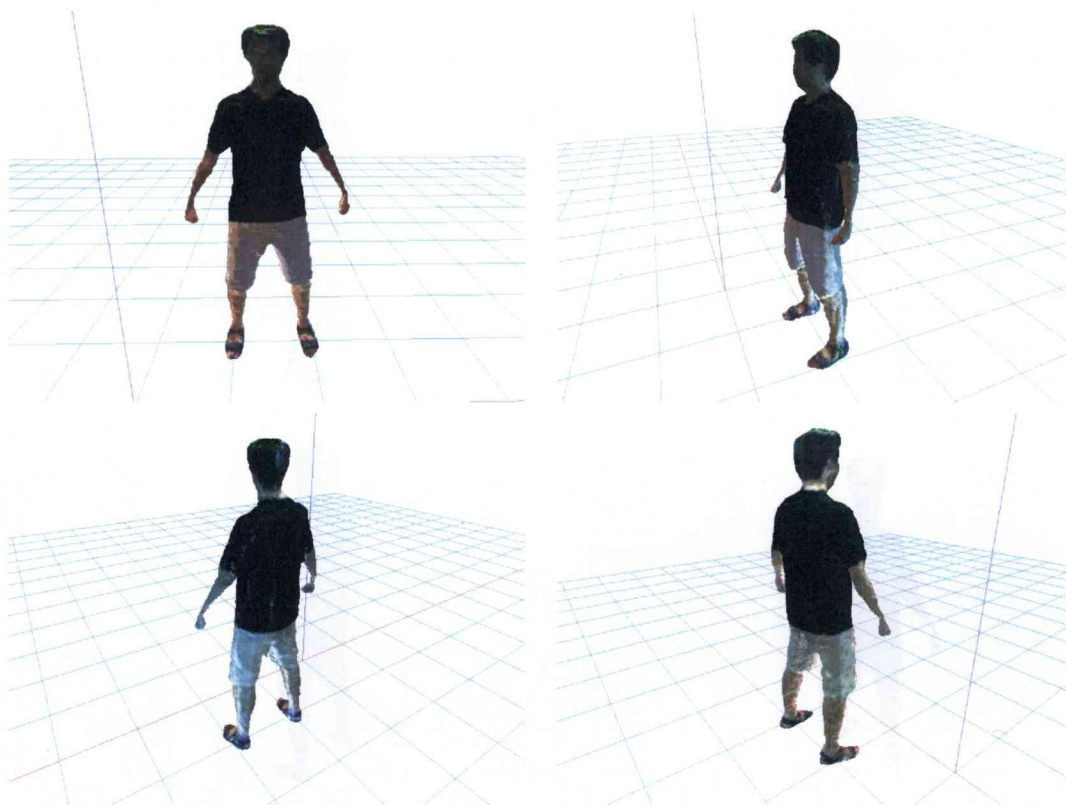


Figure 5.14: *Sample images of object colored by estimated reflectance parameters*

- ◆ **Sharp method:** renders a vertically split image arranging images for left and right eyes alternatively. Furthermore, green in each pixel is shifted 1 pixel to the right, as shown in Figure 5.18.

The physical method of displaying stereo images is similar to each other, that is, the parallax barrier method.

5.4.2 Format of 3D Video Object

5.4.2.1 Mesh

The shape of an object is represented as a mesh whose format is **OFF**, which is widely used for representing 3D object data in UNIX. The example of the mesh file is the following:

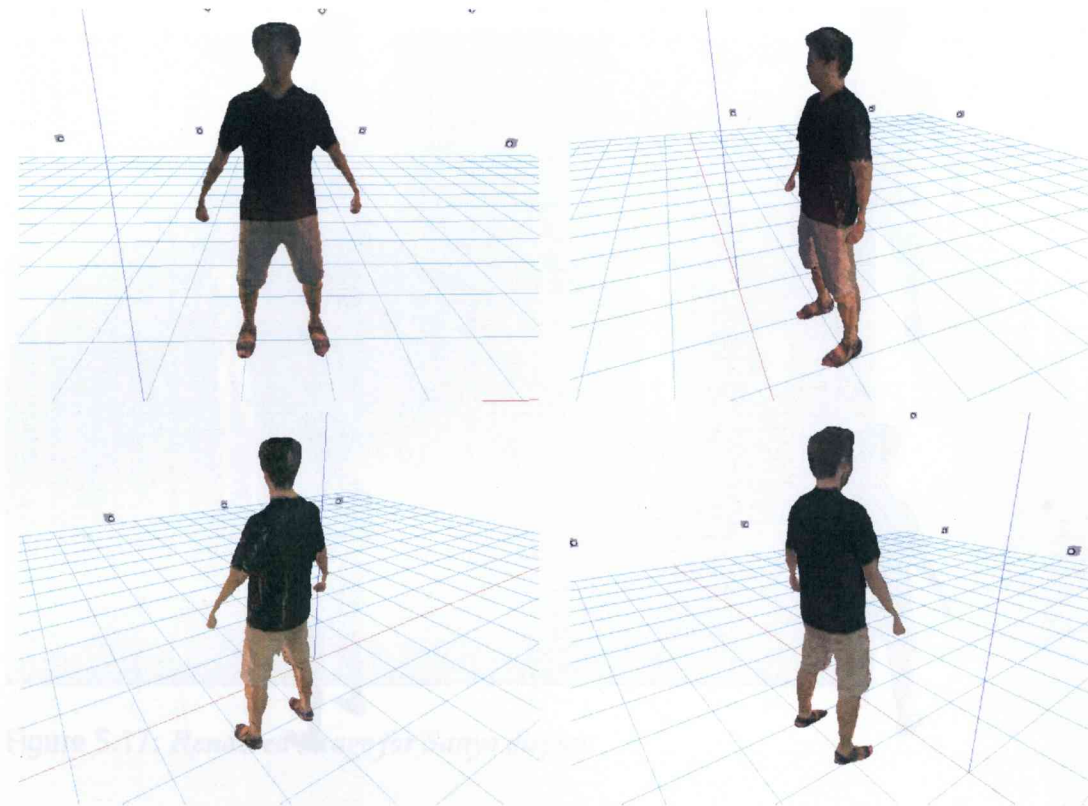


Figure 5.15: Sample images of object with new lighting environment 1

```

OFF # header
1 2 3 # number of vertices, number of faces, number of edges
x[0] y[0] z[0] # position of vertex[0]
...
3 v[0] v[1] v[2] # vertex id on this face
...
  
```

5.4.2.2 Texture

We prepare pre-compiled texture files for the 3D video viewer. It is generated by multiple images by calculating each vertex color in each image. The texture file is represented as Figure 5.19, and Figure 5.20 shows an example of the texture file.

5.5 Discussions

In this chapter, we have presented a geometric and photometric directing for 3D video, utilizing the geometric and photometric information of 3D video. It is a great advantage

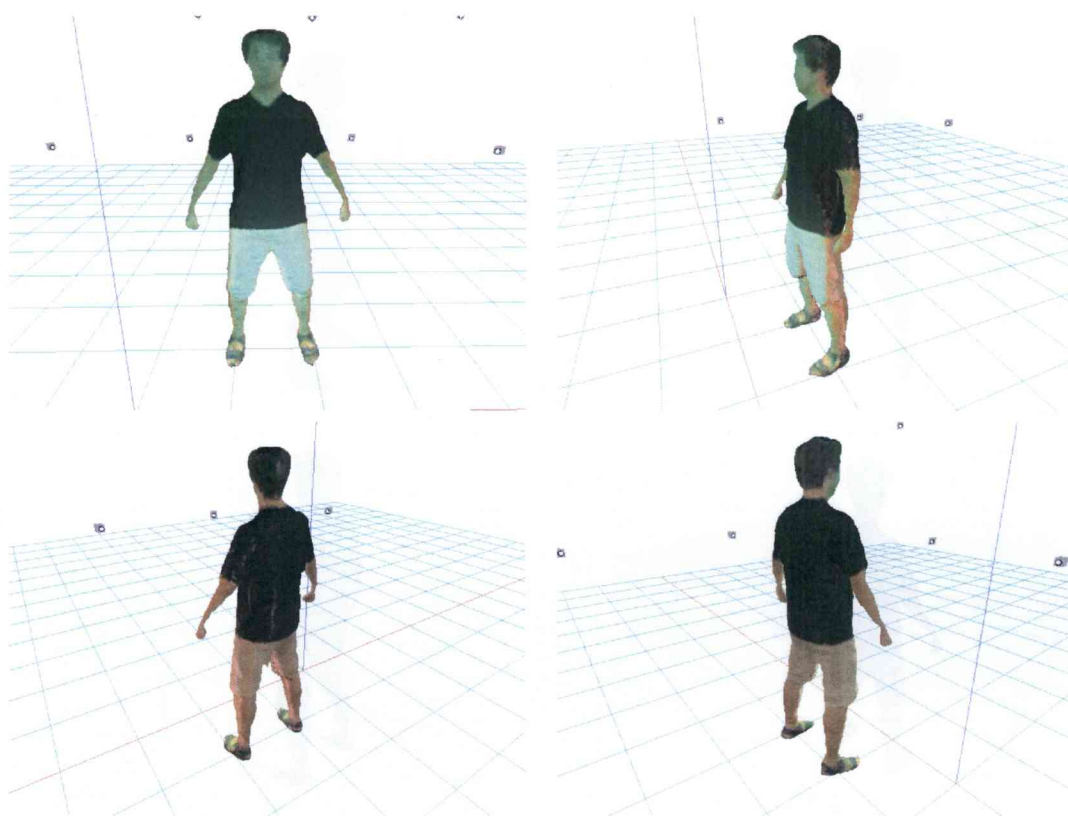


Figure 5.16: *Sample images of object with new lighting environment 2*

over ordinary image-based rendering methods, which enables us to create various expressions for free-viewpoint movies. We have developed a GUI application for the geometric directing that allows us to create camerawork easily by specifying a composition of a scene at arbitrary key frames. For the photometric directing, we have presented lighting effect control by estimating object's reflectance parameters and the lighting environment. This enables us to observe the object under not only an environment in which the object was captured, but also arbitrary (artificial) environments.

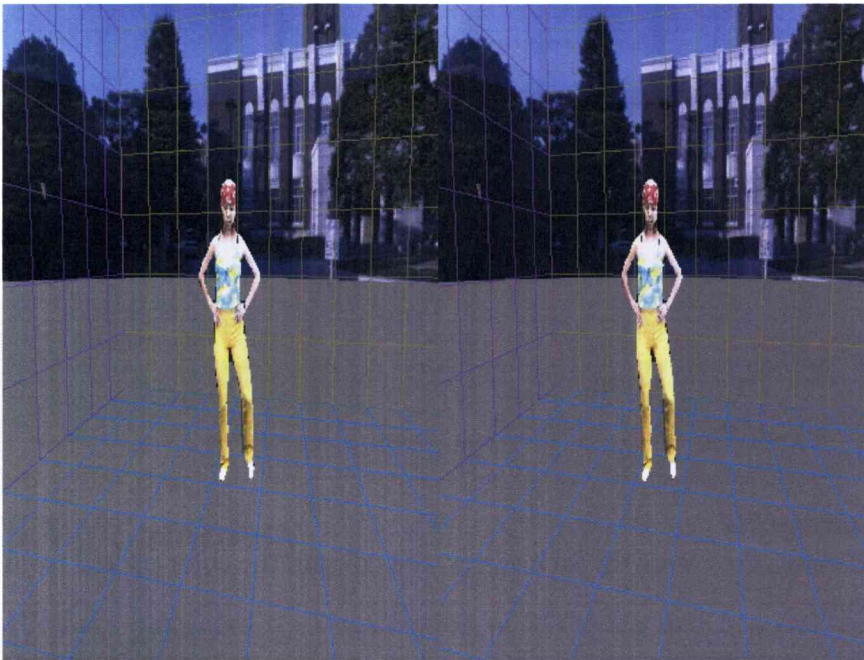


Figure 5.17: *Rendered image for Sanyo display*

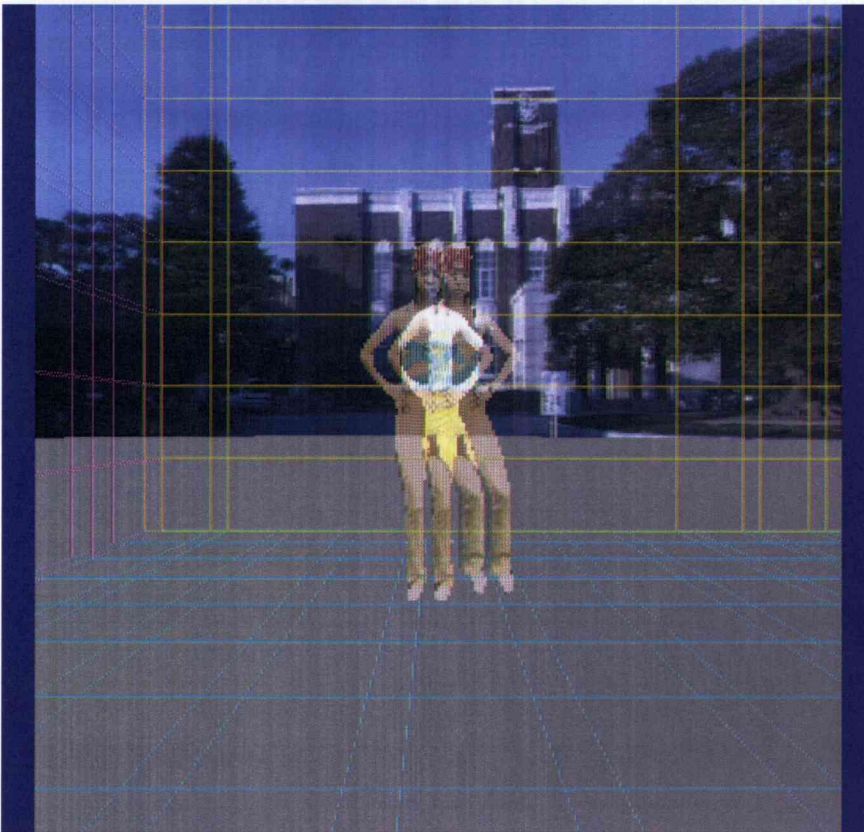


Figure 5.18: *Rendered image for Sharp display*

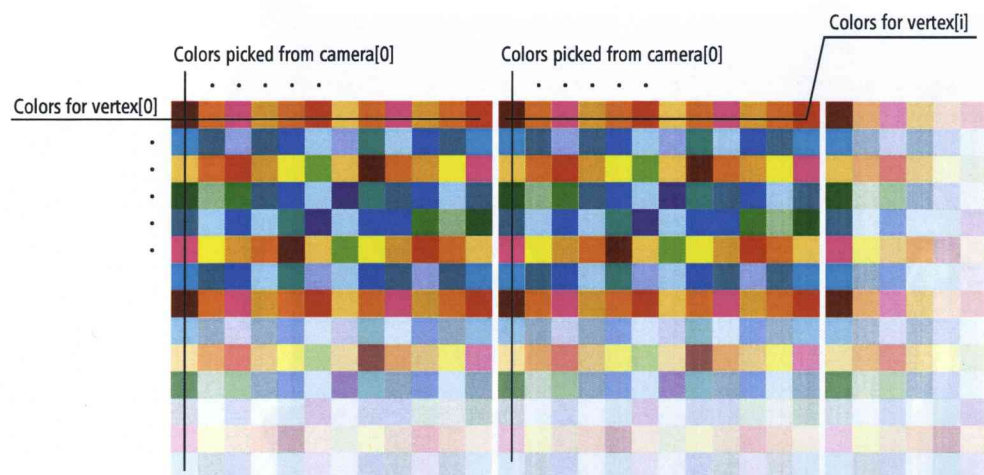


Figure 5.19: *Format of texture file*

The row of the texture denotes the colors for each vertex, the column the color picked from the image of each camera.

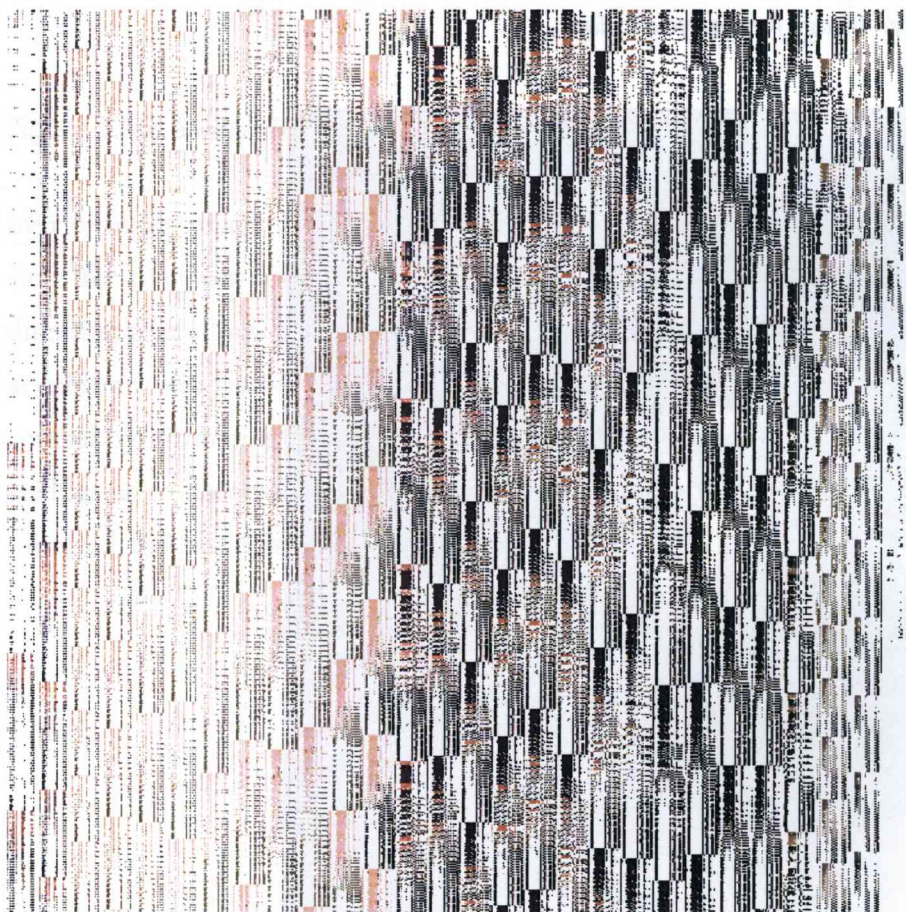


Figure 5.20: *An example of texture file*

Chapter 6

Concluding Remarks

6.1 Summary

We have presented the visualization of 3D video that is a real movie recording dynamic visual events in the real world as they are: time-varying 3D object shapes with a high-fidelity surface texture. We have proposed two different types of 3D video visualization, we call one of them “high fidelity visualization” and “versatile visualization,” which are defined them as following:

- ◆ **High fidelity visualization:** The goal of the high fidelity visualization is to generate free-viewpoint object images that reflect input multi-view images with high fidelity.
- ◆ **Versatile visualization:** The goal of the versatile visualization is to generate free-viewpoint object images with versatile effects, i.e., geometric directing by utilizing a shape and/or motion, and photometric directing by estimating reflectance parameters and lighting environment.

For the high fidelity visualization, we have proposed the viewpoint dependent vertex-based texture mapping method that enables us to visualize a 3D video object even though it has the shape of limited accuracy.

For the geometric directing, we have proposed to generate camerawork —composition of the scene and parameters of a virtual camera from which we observe an object— by utilizing the shape of an object. For the photometric directing, we have proposed two types of methods for lighting environment estimation of real scenes utilizing reference objects, i.e., the skeleton cube and the difference sphere. The skeleton cube has a capability to estimate complex lighting environment, and the difference sphere can estimate the precise parameters

of lighting environment. We have described estimation methods using the skeleton cube and the difference sphere independently, and moreover discussed the combination of the two methods. With the estimated lighting environments, we have estimated the reflectance parameters of an object.

We have finally demonstrated the versatile visualization of 3D video with the shape, the reflectance parameters, and the lighting environment, and showed its effectiveness with the various visualized images.

6.2 Future Work

In this section, we describe the future work that improves 3D video visualization from points of view of the high fidelity visualization and the versatile visualization.

6.2.1 High Fidelity Visualization

For the high fidelity visualization, we have proposed the viewpoint dependent vertex-based method. With the method, we can visualize a 3D video object with high fidelity even though the shape is limited accuracy. The texture, however, looks blurred because it is generated by interpolating the colors of vertices. We therefore need to employ texture-based method for high resolution textures, and overcome the difficulties of discontinuity and jittering.

6.2.2 Versatile Visualization

As we have described in Chapter 5, we have proceeded the first step of the geometric and the photometric directing. That is, we have prepared technical components of the directing, i.e., the 3D shape, the reflectance parameter, and the lighting environment.

6.2.2.1 Geometric Directing

While we assume that the posture of the 3D video object is known, the estimation of the posture is one of the significant tasks.

Although we have proposed the camerawork generation for one virtual camera, we need to consider camerawork for multiple cameras, e.g., the camerawork including smooth switching of multiple cameras' compositions of the scene, timing for switching, and so on.

The camerawork generation that is based on impressions that are caused by expressions of camerawork [Ari76] is also a good application for the geometric directing of 3D video.

6.2.2.2 Photometric Directing

Reflectance Property Estimation: We need to consider the specular reflection in order to visualize an object in the real scene, e.g., *kimono* and equipment that show specularities. Because it is difficult to estimate the precise parameters including the specular reflection directly from the multi-view images, we can consider the method as following:

1. Measure the precise parameters of an object a priori with a multi-spectrum camera.
2. Estimate the position of the object in multi-view images and encode the position and the reflectance parameters to 3D video.

Lighting Environment Estimation:

Skeleton cube: We need to consider the following points of view:

◆ Shape optimization:

We could optimize the design of the skeleton cube for further effective lighting estimation, i.e., its size and the width of pillar. Analysis of Matrix K (see Section 3.5.1) suggests a guideline of optimal shape of the skeleton cube.

◆ Arrangement of light sources:

An approach for enhancing the lighting environment model will be to combine different lighting distributions such as those employed in the simulation. Another practical approach will be to first obtain a rough model of lighting environment, and then refine it by iterating the estimation, i.e. by a coarse-to-fine technique.

Difference sphere: We first would like to examine our method in a further complex lighting environment since the current algorithm has naturally limitations even though we have described our algorithm as a general framework. For instance, notwithstanding the verification of S -surfaces, it will be difficult to deal with the cases with a very large number of light sources which make the identification of S -surface(s) complicated. In order to properly handle the situation, however, it should be useful to consider critical points for region segmentation, as discussed in [WS02].

Another aspect that we wish to investigate is the availability of Lambertian reflectance model for estimation of complex lighting environment. Since the Lambertian reflectance model performs low-pass filtering of the lighting environment [RH01, BJ03, NE03], the problem may be ill-posed or numerically ill-conditioned. For the problem to be alleviated, we need to consider the configuration of reference spheres for effectively generating the difference sphere although our algorithm should work once the S -surfaces are identified, even if the lighting environment is rather complex. Also, the other direction of extensions will be to model other light sources that have size and range, for estimating natural illumination.

Bibliography

- [AK97] T. Asano and S. Kimura. Contour representation of an image with applications. In *IPSJ Signotes Algorithms*, number 058-009, pages 65–70, 1997.
- [Ari76] Daniel Arijon. *The Grammar of the Film Language*. Focal Press, 1976.
- [BJ03] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):281–233, 2003.
- [Deb98] P. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH*, pages 189–198, 1998.
- [Deb99] P. Debevec. The bullet time, 1999. <http://www.debevec.org/Campanile/>.
- [DOSS04] F. Du, T. Okabe, Y. Sato, and A. Sugimoto. Reflectance estimation from motion under complex illumination. In *IEEE International Conference on Pattern Recognition*, volume I, pages 50–57, 2004.
- [FT04] T. Fujii and M. Tanimoto. Free-viewpoint tv (ftv) system. *PCM 2004*, pages 497–504, 2004.
- [HCS96] L. He, M. F. Cohen, and D. H. Salesin. The virtual cinematographer: A paradigm for automatic real-time camera control and directing. *Computer Graphics*, 30(Annual Conference Series):217–224, 1996.
- [HNI03] K. Hara, K. Nishino, and K. Ikeuchi. Determining reflectance and light position from a single image without distant illumination assumption. In *IEEE International Conference on Computer Vision*, pages 560–567, 2003.

- [INHO03] K. Ikeuchi, A. Nakazawa, K. Hasegawa, and T. Oishi. The great buddha project: Modeling cultural heritage for vr systems through observation. In *The second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2003)*, 2003.
- [IS91] K. Ikeuchi and K. Sato. Determining reflectance properties of an object using range and brightness images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1139–1153, 1991.
- [Kan01] T. Kanade. The eyevision, 2001. <http://www-2.cs.cmu.edu/afs/cs/project/VirtualizedR/www/explain.html>.
- [KKI99] Y. Kenmochi, K. Kotani, and A. Imiya. Marching cubes method with connectivity consideration. In *PRMU-98-218*, pages 297–204, 1999.
- [KRN97] T. Kanade, P. Rander, and P.J. Narayanan. Virtualized reality: Constructing virtual worlds from real scene. In *IEEE Multimedia*, pages 34–47. IEEE CG&A, 1997.
- [KSA⁺01] I. Kitahara, H. Saito, S. Akimichi, T. Ono, Y. Ohta, and T. Kanade. Large-scale virtualized reality. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.
- [Mat98] T. Matsuyama. Cooperative distributed vision –dynamic integration of visual perception, action, and communication–. In *Image Understanding Workshop*, pages 365–384, 1998.
- [MBR⁺00] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. MacMillan. Image-based visual hulls. In *ACM SIGGRAPH*, pages 369–374, 2000.
- [MG97] S. R. Marschner and D. P. Greenberg. Inverse lighting for photography. In *Fifth Color Imaging Conference*, pages 262–265, 1997.
- [MNO98] Y. Mukaigawa, Y. Nakamura, and Y. Ohta. Face synthesis with arbitrary pose and expression from several images—an integration of image-based and model-based approach—. In *Asian Conference on Computer Vision*, volume 1, pages 680–687, 1998.

- [MTG97] S. Moezzi, L. Tai, , and P. Gerard. Virtual view generation for 3d digital video. In *IEEE Multimedia*, pages 18–26. IEEE CG&A, 1997.
- [MYT03] T. Machida, N. Yokota, and H. Takemura. Surface reflectance modeling of real objects with interreflections. In *IEEE International Conference on Computer Vision*, volume I, pages 110–117, 2003.
- [NE01] P. Nillius and J.-O. Eklundh. Automatic estimation of the projected light source direction. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 1076–1083, 2001.
- [NE03] P. Nillius and J.-O. Eklundh. Low-dimensional representation of shaded surfaces under varying illumination. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 185–192, 2003.
- [Nik04] Nikon. *Nikon D70*, 2004.
- [NM04] S. Nobuhara and T. Matsuyama. Heterogeneous deformation model for 3d shape and motion recovery from multi-viewpoint images. In *The 2nd International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 566–573, 2004.
- [NN04a] K. Nishino and S. K. Nayar. Eyes for relighting. In *ACM SIGGRAPH*, volume 23, pages 704–711, 2004.
- [NN04b] K. Nishino and S. K. Nayar. The world in an eye. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 444–451, 2004.
- [NSI01] K. Nishino, Y. Sato, and K. Ikeuchi. Eigen-texture method: Appearance compression and synthesis based on a 3d model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1257–1265, 2001.
- [Oku04] Okutomi, editor. *Digital Image Processing*. CG-ARTS Society, 2004.
- [OMS99] A. Ota, K. Mori, and Y. Suenaga. A method for automated generation of a camerawork in computer graphics animation. In *Technical report of IEICE, PRMU*, pages 31–37, 1999.

- [Pen82] A. P. Pentland. Finding the illumination direction. *Journal of Optical Society of America*, 72(4):448–455, 1982.
- [Pho75] B. Phong. Illumination for computer generated pictures. In *Communications of the ACM*, pages 311–317, 1975.
- [PSG01] M. W. Powell, S. Sarkar, and D. Goldgof. A simple strategy for calibrating the geometry of light sources. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1022–1027, 2001.
- [RH01] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. In *ACM SIGGRAPH*, pages 117–128, 2001.
- [San] Sanyo. http://www.sanyo.co.jp/r_and_d/index_e.html.
- [Sha] Sharp. <http://www.sharp3d.com/>.
- [Son01] Sony. *Sony DFW-VL500 Technical Manual*, 2001.
- [SSI99a] I. Sato, Y. Sato, , and K. Ikeuchi. Illumination distribution from brightness in shadows: Adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. In *IEEE International Conference on Computer Vision*, pages 875–882, 1999.
- [SSI99b] I. Sato, Y. Sato, and K. Ikeuchi. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):1–12, 1999.
- [SWI97] Y. Sato, M. D. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. In *ACM SIGGRAPH 97*, pages 379–387, 1997.
- [THIS97] H. Tezuka, A. Hori, Y. Ishikawa, and M. Sato. Pm: An operating system coordinated high performance communication library *high-performance computing and networking* (p. sloot and b. hertzberger, eds.). In *Lecture Notes in Computer Science*, volume 1225, pages 708–717, 1997.
- [TI03] R. T. Tan and K. Ikeuchi. Separating reflection components of textured surfaces using a single image. In *IEEE International Conference on Computer Vision*, pages 870–877, 2003.

- [TM02] S. Tokai and E. Miyagawa. Generating virtual camera-work image sequence from a tracking sequence by an active camera system. In *Meeting on Image Recognition and Understanding (MIRU) (in Japanese)*, 2002.
- [TS67] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughness surface. *Journal of the Optical Society of America*, 57:1105–1114, 1967.
- [TT02] T.Matsuyama and T.Takai. Generation, visualization, and editing of 3d video. In *The 1st Symposium on 3D Data Processing Visualization and Transmission*, pages 234–245, 2002.
- [WM96] T. Wada and T. Matsuyama. Appearance sphere:background model for pan-tilt- zoom camera. In *13th International Conference on Pattern Recognition*, pages A-718–A-722, 1996.
- [WM03] X. Wu and T. Matsuyama. Real-time active 3d shape reconstruction for 3d video. In *3rd International Symposium on Image and Signal Processing and Analysis*, pages 186–191, 2003.
- [WS02] Y. Wang and D. Samaras. Estimation of multiple illuminants from a single image of arbitrary known geometry. In *IEEE European Conference on Computer Vision*, pages 272–288, 2002.
- [YXSM99] R. Yumiba, W. Xiaojun, S.Tokai, and T. Matsuyama. Natural scene visualization using active camera —seamless composition of central camera image and peripheral panorama—. In *CVIM*, pages 15–19, 1999.
- [YY91] Y. Yang and A. Yuille. Sources from shading. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 534–539, 1991.
- [ZC91] Q. Zheng and E. Chellappa. Estimation of illuminant direction, albedo, and shape from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):680–702, 1991.
- [ZK02] W. Zhou and C. Kambhamettu. Estimation of illuminant direction and intensity of multiple light sources. In *IEEE European Conference on Computer Vision*, pages 206–220, 2002.

- [ZY01] Y. Zhang and Y. Yang. Multiple illuminant direction detection with application to image system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:915–920, 2001.

List of Publications

◆ Journal Paper

1. T. Matsuyama, X. Wu, T. Takai, and S. Nobuhara. Real-time 3D shape reconstruction, dynamic 3D mesh deformation, and high fidelity visualization for 3D video. *Computer Vision and Image Understanding*, 96 (2004), pages 393–434, 2004.
2. T. Matsuyama, X. Wu, T. Takai, and T. Wada. Real-Time Dynamic 3-D Object Shape Reconstruction and High-Fidelity Texture Mapping for 3D Video. *IEEE Transactions on Circuits and systems for Video Technology*, pages 357–369, 2004.
3. T. Matsuyama, T. Takai, X. Wu, and S. Nobuhara. Generation, Editing, and Visualization of 3D Video. *The Journal of The Virtual Reality Society of Japan*, Vol.7, No. 4, pages 521–532, 2002.
4. T. Takai and T. Matsuyama. High Fidelity Visualization Algorithm and 3D Editing System for 3D Video. *The Journal of the Institute of Image Information and Television Engineers*, Vol.56, No. 4, pages 593–602, 2002.

◆ International Conference

1. T. Takai, K. Niinuma, A. Maki, and T. Matsuyama. Difference Sphere: An Approach to Near Light Source Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. I, pages 98–105, 2004.
2. T. Matsuyama and T. Takai. Generation, Visualization, and Editing of 3D Video, In *1st International Symposium on 3D Data Processing Visualization and Transmission*, pages 234–245, 2002.

3. T. Takai and T. Matsuyama. Interactive Viewer for 3D Video, In *The 4th International Workshop on Cooperative Distributed Vision*, pages 475–494, 2001.

◆ Presentation

1. T. Takai, A. Maki, and T. Matsuyama. Skeleton Cube for Lighting Environment Estimation. In *Meeting on Image Recognition and Understanding 2004*, Vol. II, pages 241–246, 2004.
2. T. Takai, K. Niinuma, and T. Matsuyama. 3D Lighting Environment Sensing with Reference Spheres, In *Information Processing Society of Japan SIG Notes*, 2003-CVIM-137, pages 117–124, 2003.
3. T. Takai and T. Matsuyama. Generation, Visualization, and Editing of 3D Video, In *The Institute of Image Information and Television Engineers*, Vol.25, No.53, pages 17–22, 2001.